南京大学 ACM-ICPC 集训队
calabash_boy
代码模版库

目录

1 String

1.1 Hash

```
427e  // Created by calabash_boy on 18-6-1.
427e  // CF 1003F
302f  #include<bits/stdc++.h>
421c  using namespace std;
b773  typedef unsigned long long ULL;
93c3  const int maxn = 305*305;
75c0  /* 字符集大小 */
0852  const int sigma = maxn;
0338  /* hash次数 */
cab3  const int HASH_CNT = 2;
5c83  int n;
4c95  int s[maxn];
bef3   /* char* 1-bas
5cb4   * sum[i] = s[i]+s[i-1]*Seed+s[i-2]*Seed^2+...+s[1]*Seed^(i-1)*/
cf6f  ULL Prime_Pool[] = {1998585857ul,23333333333ul};
d095  ULL Seed_Pool[]={911,146527,19260817,91815541};
c437  ULL Mod_Pool[]={29123,998244353,1000000009,4294967291ull};
b060  struct Hash_1D{
3e0c      ULL Seed,Mod;
3bc4      ULL bas[maxn];ULL sum[maxn];
ad94      int perm[sigma];
be03      void init(int seedIndex,int modIndex){
e7a7          Seed = Seed_Pool[seedIndex];
53c7          Mod = Mod_Pool[modIndex];
bf6d          bas[0]=1;
6dbf          for (int i=1;i<=n;i++){
d57c              bas[i] = bas[i-1]*Seed%Mod;
95cf          }
6dbf          for (int i=1;i<=n;i++){
1e15              sum[i] = (sum[i-1]*Seed%Mod+s[i])%Mod;
95cf          }
95cf      }
c2c1      /*random_shuffle 离散化id，防止kill_hash*/
b864      void indexInit(int seedIndex,int modIndex){
324a          for (int i=1;i<n;i++){
871a              perm[i]=i;
95cf          }
cee0          random_shuffle(perm+1,perm+1+sigma);
e7a7          Seed = Seed_Pool[seedIndex];
```

```
        Mod = Mod_Pool[modIndex];                                  53c7
        bas[0]=1;                                                  bf6d
        for (int i=1;i<=n;i++){                                    6dbf
            bas[i] = bas[i-1]*Seed%Mod;                            d57c
        }                                                          95cf
        for (int i=1;i<=n;i++){                                    6dbf
            sum[i] = (sum[i-1]*Seed%Mod+perm[s[i]])%Mod;           cd52
        }                                                          95cf
    }                                                              95cf
    ULL getHash(int l,int r){                                      b2c3
        return (sum[r]-sum[l-1]*bas[r-l+1]%Mod+Mod)%Mod;           46bc
    }                                                              95cf
}hasher[HASH_CNT];                                                 bb59
map<pair<pair<ULL,ULL>,int>,int>veid;int vecnt;                    f09b
map<string,int>id;int idcnt;                                       5d53
vector<int> pos[maxn];                                             7fbd
string a[maxn];                                                    fae2
int sumL[maxn];                                                    f06b
int main(){                                                        3117
    cin>>n;                                                        e1b6
    for (int i=1;i<=n;i++){                                        6dbf
        cin>>a[i];                                                 879c
        if (!id[a[i]])id[a[i]] = ++idcnt;                          d0a8
        s[i] = id[a[i]];                                           7798
        sumL[i] = sumL[i-1]+a[i].size();                           9892
    }                                                              95cf
    for (int i=0;i<HASH_CNT;i++){                                  da02
        hasher[i].indexInit(i,i);                                  42fc
    }                                                              95cf
    int ans = sumL[n]+n-1;                                         b20c
    for (int i=1;i<=n;i++){                                        6dbf
        for (int j=1;j<=n;j++){                                    ede7
            ULL hash1 = hasher[0].getHash(i,j);                    e9bb
            ULL hash2 = hasher[1].getHash(i,j);                    2a70
            int len = j-i+1;                                       de4a
            pair<pair<ULL,ULL>,int> x = {{hash1,hash2},len};       46fa
            if (veid[x]==0)veid[x] = ++vecnt;                      67ca
            pos[veid[x]].push_back(i);                             2251
        }                                                          95cf
    }                                                              95cf
    int maxDelta =0;                                               04c1
    for (auto x:veid){                                             0086
        int len = x.first.second;                                  5c1e
        int i = x.second;                                          76c1
        sort(pos[i].begin(),pos[i].end());
        int num =0;
        for (int j=0,last = -maxn;j<pos[i].size();j++){
            if (pos[i][j]>=last+len){
                last = pos[i][j];
                num++;
            }
        }
        if (num==1)continue;
        int cost1 = sumL[pos[i][0]+len-1]-sumL[pos[i][0]-1]+len-1;
        int cost2 = len;
        int tempDelta = (cost1-cost2)*num;
        maxDelta = max(maxDelta,tempDelta);
    }
    cout<<ans-maxDelta<<endl;
    return 0;
}
```

1.2 KMP

```
// Created by calabash_boy on 18-7-23.
//最小权值和 二维循环节
//找到最小 每行公共循环节+每列公共循环节。
//单调队列找固定大小矩形最小权值和。
#include<bits/stdc++.h>
using namespace std;
const int maxn = 1e6+100;
struct KMP{
    int nxt[maxn];int len;
    char t[maxn];
    void clear(){
        len =nxt[0] = nxt[1] =0;
    }
    /* 1-bas */
    /* 注意在ss结尾添加‘\0’ */
    void init(char* ss){
        len = strlen(ss+1);
        memcpy(t,ss,(len+2)*sizeof(char));
        for (int i=2;i<=len;i++){
            nxt[i] = nxt[i-1];
            while (nxt[i]&&ss[i]!=ss[nxt[i]+1]) nxt[i] = nxt[nxt[i]];
            nxt[i]+=(ss[i]==ss[nxt[i]+1]);
        }
    }
    /* 求所有在ss串中的start_pos. 如果first_only设置为true，则只返回第一个位置*/
    vector<int> match(char *ss,bool first_only = false){
        int len_s = strlen(ss+1);
        vector<int> start_pos(0);
        for (int i=1,j=1;i<=len_s;){
            while (j!=1 && ss[i] != t[j])j = nxt[j-1]+1;
            if (ss[i] == t[j]) j++,i++;
            else i++;
            if (j == len+1){
                start_pos.push_back(i-j+1);
                if (first_only)return start_pos;
                j = nxt[len]+1;
            }
        }
        return start_pos;
    }
    void debug(){
        for (int i=0;i<=len;i++){
            printf("[debug]␣nxt[%d]=%d\n",i,nxt[i]);
        }
    }
    /* 循环周期 形如 acaca 中 ac 是一个合法周期 */
    vector<int> periodic(){
        vector<int> ret;
        int now = len;
        while (now){
            now = nxt[now];
            ret.push_back(len-now);
        }
        return ret;
    }
    /* 循环节 形如 acac 中ac、acac是循环节，aca不是*/
    vector<int> periodic_loop(){
        vector<int>ret ;
        for (int x :periodic()){
            if (len%x==0)ret.push_back(x);
        }
        return ret;
    }
    int min_periodic_loop(){
        return periodic_loop()[0];
    }
997f }kmper;
0324 vector<string> s;
04c5 vector<vector<int> > a,maxVal;
0fcd int cnt1[maxn],cnt2[maxn],n,m;
5f67 char S[maxn];
e6f2 pair<int,int> pq[maxn];int l,r;
3117 int main(){
9af0     cin>>n>>m;
9d25     s.resize(n+1);
035f     maxVal.resize(n+1);
6dbf     for (int i=1; i<=n;i++){
f9af         cin>>s[i];
95cf     }
246a     a.resize(n+1);
6dbf     for (int i=1;i<=n;i++){
4356         a[i].resize(m+1);
0901         maxVal[i].resize(m+1);
8e5f         for (int j=1;j<=m;j++){
0fb4             cin>>a[i][j];
95cf         }
95cf     }
d580     int p,q;kmper.clear();
6dbf     for (int i=1;i<=n;i++){
8e5f         for (int j=1;j<=m;j++){
69f1             S[j] = s[i][j-1];
95cf         }
5239         S[m+1]='\0';
8dce         kmper.init(S);
1d4f         for (int x:kmper.periodic()){
3b83             cnt1[x]++;
95cf         }
95cf     }
8e5f     for (int j=1;j<=m;j++){
6dbf         for (int i=1;i<=n;i++){
3e08             S[i] = s[i][j-1];
95cf         }
80ba         S[n+1]='\0';
8dce         kmper.init(S);
1d4f         for (int x:kmper.periodic()){
e14e             cnt2[x]++;
95cf         }
95cf     }
b042     for (int i=maxn;i>=1;i--){
415e         if (cnt1[i]==n){ q = i; }
a87c         if (cnt2[i]==m){ p=i; }
95cf     }
6dbf     for (int i=1;i<=n;i++){
25ea         l = 0,r=0;
8e5f         for (int j=1;j<=m;j++){
872e             while (r>l&&pq[l].second<=j-q)l++;
26e9             while (r>l&&pq[r-1].first<=a[i][j])r--;
3497             pq[r++] = {a[i][j],j};
862b             if (j>=q){
1dcc                 maxVal[i][j-q+1] = pq[l].first;
95cf             }
95cf         }
95cf     }
54ad     int ans = 0x3f3f3f3f;
2f5d     for (int j=1;j<=m-q+1;j++){
edd7         l=r=0;
6dbf         for (int i=1;i<=n;i++){
be46             while (r>l&&pq[l].second<=i-p)l++;
bb56             while (r>l&&pq[r-1].first<=maxVal[i][j])r--;
c5e8             pq[r++] = {maxVal[i][j],i};
b6cf             if (i>=p){
3003                 ans = min(ans,pq[l].first);
95cf             }
427e 
95cf         }
95cf     }
fc9a     cout<<1LL*(p+1)*(q+1)*ans<<endl;
7021     return 0;
95cf }
```

1.3 EX KMP

```
427e // Created by calabash_boy on 2019/12/11.
427e // CF 1200E
302f #include <bits/stdc++.h>
421c using namespace std;
94a1 const int maxn = 1e6 + 100;
427e //result: extend[i] = LCP(S[i,lens],T)
427e //require: nxt[i] = LCP(T[i,lent],T)
4543 void exkmp(char *s,int lens,char *t,int lent,int *extend,int *nxt){
deaa     extend[0] = 0;
05de     for (int i = 1,p0 = 0,p = 0;i <= lens;i ++){
```

```
c132        extend[i] = i <= p ? min(nxt[i - p0 + 1], p - i + 1) : 0;
d4c2        while (i + extend[i] <= lens and extend[i] < lent and s[i + extend[i]]
              == t[extend[i]+1])extend[i] ++;
cc3f        if (i + extend[i] - 1 >= p and i != 1)p0 = i,p = i + extend[i] - 1;
95cf    }
95cf }
8304 char s[maxn],t[maxn];
bbbc int extend[maxn];
51d9 int nxt[maxn];
3117 int main(){
5c83     int n;
cd91     scanf("%d",&n);
9f73     int LEN = 0;
6dbf     for (int i=1;i<=n;i++){
79ca         scanf("%s",t+1);
be5a         int lent = strlen(t+1);
f312         int lens = min(LEN,lent);
0640         exkmp(t,lent,t,lent,nxt,nxt);
b881         exkmp(s + LEN - lens,lens,t,lent,extend,nxt);
eb16         int sp = 0;
0d89         for (int j = 1;j <= lens;j ++){
80e7             if (lens - j + 1 == extend[j]){
fa6b                 sp = extend[j];
6173                 break;
95cf             }
95cf         }
1e32         strcpy(s + LEN+1,t + sp + 1);
df1b         LEN += lent - sp;
95cf     }
ae85     printf("%s\n",s+1);
7021     return 0;
95cf }
```

1.4 Manacher

```
427e // Created by calabash_boy on 18-9-14.
302f #include<bits/stdc++.h>
421c using namespace std;
571f const int MAX = 2e5+10000;
99d0 char s[MAX];
81d4 struct Manacher{
9ccd     int lc[MAX];
    char ch[MAX];                                        04f3
    int N;                                               d7af
    Manacher(char *s){init(s);manacher();}               053c
    /* s 1 bas */                                        44ca
    void init(char *s){                                  e798
        int n = strlen(s+1);                             0de8
        ch[n*2 +1] = '#';                                ad19
        ch[0] = '@';                                     ce0d
        ch[n*2 +2] = '\0';                               46cd
        for (int i=n;i>=1;i--){                          0c3f
            ch[i*2] = s[i];ch[i*2 -1] = '#';             6beb
        }                                                95cf
        N = 2* n +1;                                     5991
    }                                                    95cf
    void manacher(){                                     6c5f
        lc[1]=1;  int k=1;                               a461
        for (int i=2;i<=N;i++){                          256b
            int p = k+lc[k]-1;                           7957
            if (i<=p){                                   5e04
                lc[i]=min(lc[2*k-i],p-i+1);              24a1
            }else{  lc[i]=1;  }                          87d6
            while (ch[i+lc[i]]==ch[i-lc[i]])lc[i]++;     aa80
            if (i+lc[i]>k+lc[k])k=i;                     2b9a
        }                                                95cf
    }                                                    95cf
    void debug(){                                        56dd
        puts(ch);                                        b492
        for (int i=1;i<=N;i++){                          cd0f
            printf("lc[%d]=%d\n",i,lc[i]);               0d62
        }                                                95cf
    }                                                    95cf
};                                                       329b
int main(){                                              3117
    scanf("%s",s+1);                                     a275
    Manacher manacher(s);                                382e
    manacher.debug();                                    9c07
    return 0;                                            7021
}                                                        95cf
```

1.5 Palindrome Series

```
// Created by calabash_boy on 19-11-20.                 427e
```

```
427e // CF 932G 偶回文分割方案数
427e // 优化DP转移：枚举所有回文后缀转移
427e // 使用时，只需要修改3行TBD
302f #include<bits/stdc++.h>
421c using namespace std;
5d33 const int mod = 1e9 + 7;
94a1 const int maxn = 1e6+100;
466b struct Palindromic_AutoMaton{
9f36     int s[maxn],now;
f801     int nxt[maxn][26],fail[maxn],l[maxn],last,tot;
7220     int diff[maxn],anc[maxn],g[maxn],f[maxn];
1126     void clear(){
427e         //1节点：奇数长度root 0节点：偶数长度root
78a6         s[0] = l[1] = -1;
b6d0         fail[0] = tot = now =1;
f40b         last = l[0] = 0;
21a1         memset(nxt[0],0,sizeof nxt[0]);
9b85         memset(nxt[1],0,sizeof nxt[1]);
95cf     }
61ff     Palindromic_AutoMaton(){clear();}
7c3e     int newnode(int len){
71cf         tot++;
87f4         memset(nxt[tot],0,sizeof nxt[tot]);
cdd3         fail[tot]=0;l[tot]=len;
91fb         return tot;
95cf     }
4284     int get_fail(int x){
8ef1         while (s[now-l[x]-2]!=s[now-1])x = fail[x];
d074         return x;
95cf     }
a791     void add(int ch){
3622         s[now++] = ch;
051b         int cur = get_fail(last);
a980         if(!nxt[cur][ch]){
80d2             int tt = newnode(l[cur]+2);
2f33             fail[tt] = nxt[get_fail(fail[cur])][ch];
01cb             nxt[cur][ch] = tt;
ba51             diff[tt] = l[tt] - l[fail[tt]];
9bbc             anc[tt] = diff[tt] == diff[fail[tt]]? anc[fail[tt]] : fail[tt];
95cf         }
4e23         last = nxt[cur][ch];
95cf     }
ea60     void trans(int i){
8380         for (int p = last;p>1;p = anc[p]){
2237             g[p] = f[i - l[anc[p]] - diff[p]];/*TBD*/
752d             if (diff[p] == diff[fail[p]]){
a45f                 (g[p] += g[fail[p]]) %= mod;/*TBD*/
95cf             }
4ae5             (f[i] += (i % 2 == 0) *g[p]) %= mod;/*TBD*/
95cf         }
95cf     }
01f2     int init(char* s){
0cac         f[0] = 1;
0de8         int n = strlen(s + 1);
6dbf         for (int i=1;i<=n;i++){
341e             add(s[i] - 'a');
271c             trans(i);
95cf         }
1df1         return f[n];
95cf     }
de71 }pam;
394b char t[maxn], s[maxn];
3117 int main(){
a275     scanf("%s",s + 1);
0de8     int n = strlen(s+1);
3966     for (int i=1;i<=n/2;i++){
45f1         t[2 * i - 1] = s[i];
d9af         t[2 * i] = s[n + 1- i ];
95cf     }
d348     cout<<pam.init(t)<<endl;
7021     return 0;
95cf }
```

1.6 Suffix Array

```
87e7 /*
1e1d  * for each 2-power string.
f606  * let its length is 2L. add edge of length w[L] between every i and i + L.
f3db  * calculate the spanning forests.
f2b5  */
302f #include <bits/stdc++.h>
18f5 #define rank rkrkrk
427e //#define _DEBUG
f11b #define RMQ
421c using namespace std;
6428 const int maxn = 3e5+100;
```

```
82ea int w[maxn];
1283 int lg[maxn];
2f33 struct Run{
8f36     int l,r,k;
329b };
bd89 struct UFS {
33ef     int fa[maxn];
7dd9     void init(int n) { iota(fa, fa + n + 1, 0); }
38dd     int find(int x) { return fa[x] == x ? x : fa[x] = find(fa[x]); }
9662     bool unite(int u, int v) {
576f         u = find(u); v = find(v);
2448         fa[u] = v;
4042         return u != v;
95cf     }
d71b } ufs[20];
427e
4d49 int unite(int u, int v, int k) {
10fe     if (ufs[k].unite(u, v)) {
d11e         if (k == 0) return 1;
81a9         return unite(u, v, k - 1) + unite(u + (1<<(k-1)), v + (1<<(k-1)), k - 1)
               ;
aad3     } else return 0;
95cf }
427e
6b2b long long merge(int u, int v, int l) {
0fa9     int k = log2(l);
2c46     int ret =  unite(u, v, k) +
270b                unite(u + l - (1<<k), v + l - (1<<k), k);
ee0f     return ret;
95cf }
3b88 struct SA{
4eb6 #ifndef RMQ
9c29     struct Segment_Tree{
77b7         int min_val[maxn*4];
d08d         void up(int x){
10d7             min_val[x] = min(min_val[x<<1],min_val[x<<1|1]);
95cf         }
3e01         void build(int x,int l,int r,int*h){
3a0d             if (l == r){
e948                 min_val[x] = h[l];
4f2d                 return;
95cf             }
b8b7             int mid = l + r >>1;
fdb0             build(x<<1,l,mid,h);
```

```
            build(x<<1|1,mid+1,r,h);  06e9
            up(x);  cf00
        }  95cf
        int query(int x,int l,int r,int L,int R){  30b1
            if (l > R || L > r)return 0x3f3f3f3f;  133b
            if (L<= l && r <= R)return min_val[x];  0739
            int mid = l + r >> 1;  b8b7
            return min(query(x<<1,l,mid,L,R),query(x<<1|1,mid+1,r,L,R));  edf8
        }  95cf
    }segtree;  f7fb
#else  a8cb
    int st[maxn][20];  fb7f
    void st_init(int n,int*h){  a66e
        for (int i=1;i<=n;i++){  6dbf
            st[i][0] = h[i];  fc74
        }  95cf
        for (int j=1;(1<<j)<=n;j++){  c8a2
            for (int i=1;i<=n-(1<<j)+1;i++){  672f
                st[i][j] = min(st[i][j-1],st[i+(1<<(j-1))][j-1]);  3c6e
            }  95cf
        }  95cf
    }  95cf
#endif  1937
    int cntA[maxn],cntB[maxn],tsa[maxn],A[maxn],B[maxn];  6e4f
    int sa[maxn],rank[maxn],height[maxn];  f3d8
    void get_sa(int * ch,int n){  81e4
        ch[0] = ch[n+1] = -1;  b5cc
        for (int i=0;i<=n;i++)cntA[i] = 0;  c7f9
        for (int i=1;i<=n;i++)cntA[ch[i]]++;  e86b
        for (int i=1;i<=n;i++)cntA[i] += cntA[i-1];  c35a
        for (int i=n;i>=1;i--)sa[cntA[ch[i]]--] = i;  625e
        rank[sa[1]] = 1;  c9f2
        for (int i=2;i<=n;i++){  a5c5
            rank[sa[i]] = rank[sa[i-1]];  dc5c
            if (ch[sa[i]] != ch[sa[i-1]])rank[sa[i]] ++;  459c
        }  95cf
        for (int l=1;rank[sa[n]]<n;l<<=1){  f62b
            for (int i=0;i<=n;i++)cntA[i] = cntB[i] = 0;  c794
            for (int i=1;i<=n;i++){  6dbf
                cntA[A[i] = rank[i]] ++;  d9ab
                cntB[B[i]=(i+l<=n)?rank[i+l]:0]++;  c846
            }  95cf
            for (int i=1;i<=n;i++)cntB[i] += cntB[i-1];  72d7
            for (int i=n;i>=1;i--)tsa[cntB[B[i]]--] = i;  4c62
```

```
c35a            for (int i=1;i<=n;i++)cntA[i] += cntA[i-1];
1626            for (int i=n;i>=1;i--)sa[cntA[A[tsa[i]]]--] = tsa[i];
c9f2            rank[sa[1]] = 1;
a5c5            for (int i=2;i<=n;i++){
dc5c                rank[sa[i]] = rank[sa[i-1]];
021c                if (A[sa[i]] != A[sa[i-1]] || B[sa[i]] != B[sa[i-1]])rank[sa[i]]
                        ++;
95cf            }
95cf        }
95cf    }
bbe8    void get_height(int *ch,int n){
0820        get_sa(ch,n);
5c18        sa[0] = rank[0] = 0;
0956        for (int i=1,j=0;i<=n;i++){
1a82            if (j) j--;
757e            while (ch[i+j] == ch[sa[rank[i]-1]+j])j++;
24a7            height[rank[i]] = j;
95cf        }
ed5c #ifdef _DEBUG
6dbf        for (int i=1;i<=n;i++){
dfcf            printf("height[%d]=%d\n",i,height[i]);
95cf        }
1937 #endif
4eb6 #ifndef RMQ
3b40        segtree.build(1,1,n,height);
a8cb #else
a852        st_init(n,height);
1937 #endif
95cf    }
ead2    int get_lcp(int x,int y,int n){
6606        int rkx = rank[x];
a728        int rky = rank[y];
4e5e        if (rkx>rky)swap(rkx,rky);
216a        rkx++;
4eb6 #ifndef RMQ
dee6        int lcp = segtree.query(1,1,n,rkx,rky);
a8cb #else
780d        int k = lg[(rky - rkx+1)];
f5b5        int lcp = min(st[rkx][k],st[rky - (1<<k)+1][k]);
1937 #endif
427e
ed5c #ifdef _DEBUG
33df        printf("[get_lcp]␣x=%d␣y=%d␣rkx=%d,rky=%d,lcp␣=%d\n",x,y,rkx,rky,lcp);
1937 #endif
```

```
        return lcp;  9a6a
    }  95cf
}sa1,sa2;  5a1e
int ch2[maxn];  96d9
vector<Run> get_run(int*ch,int n){  4d50
    sa1.get_height(ch,n);  7c77
    for (int i=0;i<=n+1;i++){  842e
        ch2[i] = ch[i];  13b4
    }  95cf
    reverse(ch2+1,ch2+1+n);  7db6
    sa2.get_height(ch2,n);  945d
    vector<Run> result(0);  c4b1
    int len_max = n/2;  a2dc
    for (int len = 1;len <=len_max;len ++){  dbca
        //get_len_run  427e
        for (int i=1;i<=n;i+=len){  870e
            int j = i+len;  d3da
            if (j >n)break;  dd33
            int lcp = sa1.get_lcp(i,j,n);  f2a5
            int lcs = sa2.get_lcp(n+1-i,n+1-j,n);  8ef0
            lcp = min(lcp,len);  f20d
            lcs = min(lcs,len);  97fa
            assert(j+lcp-1<=n);  2cd9
            assert(i-lcs+1>=1);  6a34
#ifdef _DEBUG  ed5c
            printf("i=%d,j=%d,len=%d,lcp=%d,lcs=%d\n",i,j,len,lcp,lcs);  8dbc
#endif  1937
            if (lcp + lcs - 1 < len)continue;  37d6
            int L = j-lcs+1;  09d8
            int R = j + lcp -1;  856e
            result.push_back((Run){L,R,len});  ab80
        }  95cf
    }  95cf
#ifdef _DEBUG  ed5c
    for (Run run : result){  7d48
        printf("[run]:␣l=%d,␣r=%d,k=%d\n",run.l,run.r,run.k);  7252
    }  95cf
#endif  1937
    return result;  56b0
}  95cf
int n;  5c83
typedef long long ll;  4085
ll spanning_forest(vector<Run> &runs){  aec3
    sort(runs.begin(),runs.end(),[](Run x,Run y){  4f70
b6e2          return w[x.k] < w[y.k];
b251      });
19f3      ll ans = 0;
ec84      for (auto& R : runs) {
de4b          int l = R.l, r = R.r;
bbac          ans += 1ll * merge(l - R.k, l, r - l + 1) * w[R.k];
95cf      }
4206      return ans;
95cf  }
7767  int ch[maxn];
3117  int main(){
c592      for (int i=2;i<maxn;i++)lg[i] = lg[i/2] + 1;
9523      int T;
1fd9      scanf("%d",&T);
60ca      while (T--){
cd91          scanf("%d",&n);
4721          for (int i = 0; i < 20 ; i++) ufs[i].init(n);
d15f          ch[n+1] = -1;
d442          ch[0] = -1;
6dbf          for (int i=1;i<=n;i++){
b3d6              scanf("%d",ch+i);
95cf          }
9f8e          int m = n/2;
e052          for (int i=1;i<=m;i++){
ef59              scanf("%d",w+i);
95cf          }
3690          vector<Run> all_run = get_run(ch,n);
1ccd          printf("%lld\n",spanning_forest(all_run));
95cf      }
7021      return 0;
95cf  }
```

1.7 Trie Graph

```
427e  // Created by calabash_boy on 2019/10/25.
302f  #include <bits/stdc++.h>
421c  using namespace std;
eb45  const int maxn = 2e5 + 100;
a281  struct Trie {
4562      int nxt[maxn][26], fail[maxn];
427e      // the node I should go when append a character
95ce      int Go[maxn][26];
    int root = 0;                                                     e7b0
    int cnt = 0;                                                      8abb
    void clear(){                                                     1126
        root = cnt = 0;                                               082c
        memset(nxt[0],0,sizeof nxt[0]);                               21a1
    }                                                                 95cf
    int newnode() {                                                   ee91
        cnt++;                                                        6fb3
        //clear memory                                                427e
        return cnt;                                                   6808
    }                                                                 95cf
    void insert(char *s) {                                            9bb4
        int now = root;                                               8f56
        while (*s){                                                   f205
            now = insert(now,*s - 'a');                               3c7d
            s ++;                                                     85be
        }                                                             95cf
    }                                                                 95cf
    int insert(int pre, int ch) {                                     7e27
        return nxt[pre][ch]?nxt[pre][ch] : nxt[pre][ch] = newnode();  721c
    }                                                                 95cf
    void build() {                                                    2114
        queue<int> Q;                                                 aafa
        Q.push(0);                                                    98ae
        memcpy(Go[0],nxt[0],sizeof nxt[0]);                           8b79
        while (!Q.empty()) {                                          11e5
            int head = Q.front();                                     fda7
            Q.pop();                                                  f2f8
            for (int ch = 0; ch < 26; ch++) {                         b3c9
                int v = nxt[head][ch];                                ec9d
                if (!v)continue;                                      e151
                if (head == 0) fail[v] = 0;                           eb93
                else fail[v] = Go[fail[head]][ch];                    f8d4
                memcpy(Go[v],Go[fail[v]],sizeof Go[v]);               2616
                for (int cc = 0; cc < 26; cc++) {                     fbe7
                    if (nxt[v][cc])Go[v][cc] = nxt[v][cc];            1696
                }                                                     95cf
                Q.push(v);                                            78e5
            }                                                         95cf
        }                                                             95cf
    }                                                                 95cf
}trie;                                                                1cc7
```

1.8 Trie Graph (Segment Tree)

```cpp
// Created by calabash_boy on 2019/10/27.
#include <bits/stdc++.h>
using namespace std;
const int maxn = 2e5 + 100;
int n;
int p[maxn], c[maxn];
struct Node{
    int lson,rson;
    int val;
    void init(){
        val = 0;
        lson = rson = -1;
    }
}nodes[maxn * 25];
int node_cnt = 1;
int build(int x,int l,int r){
    int now = node_cnt ++;
    nodes[now].init();
    if (l == r) return now;
    int mid = l + r >> 1;
    nodes[now].lson = build(x<<1,l,mid);
    nodes[now].rson = build(x<<1|1,mid+1,r);
    return now;
}
int update(int id,int l,int r,int pos,int val){
    int now = node_cnt ++;
    nodes[now] = nodes[id];
    if (l == r){
        nodes[now].val = val;
        return now;
    }
    int mid = l + r >> 1;
    if (pos <= mid)nodes[now].lson = update(nodes[id].lson,l,mid,pos,val);
    else nodes[now].rson = update(nodes[id].rson,mid+1,r,pos,val);
    return now;
}
int query(int id,int l,int r,int pos){
    if (l == r)return nodes[id].val;
    int mid = l + r >> 1;
    if (pos <= mid)return query(nodes[id].lson,l,mid,pos);
    else return query(nodes[id].rson,mid+1,r,pos);
}
struct Trie{
    map<int,int> nxt[maxn];
    int root[maxn], fail[maxn];
    int cnt = 1;
    int insert(int pre,int ch) {
        return nxt[pre].find(ch) == nxt[pre].end() ? nxt[pre][ch] = cnt++ : nxt[pre][ch];
    }
    void build(){
        queue<int> Q;Q.push(0);
        root[0] = ::build(1,1,n);
        for (auto edge : nxt[0]){
            int v,ch;
            tie(ch,v) = edge;
            root[0] = update(root[0],1,n,ch,v);
        }
        while (!Q.empty()){
            int head = Q.front();Q.pop();
            for (auto edge: nxt[head]){
                int v,ch;
                tie(ch,v) = edge;
                if (head == 0)fail[v] = 0;
                else fail[v] = query(root[fail[head]],1,n,ch);
                root[v] = root[fail[v]];
                for (auto edge2 : nxt[v]){
                    int v2,ch2;
                    tie(ch2,v2) = edge2;
                    root[v] = update(root[v],1,n,ch2,v2);
                }
                Q.push(v);
            }
        }
    }
}trie;
int main(){
    cin>>n;
    for (int i=1;i<=n;i++){
        scanf("%d",p+i);
    }
    for (int i=1;i<=n;i++){
        scanf("%d",c+i);
        trie.insert(p[i],c[i]);
    }
    trie.build();
6dbf     for (int i=1;i<=n;i++){
7f97         cout<<trie.fail[i] <<"␣";
95cf     }
3251     cout<<endl;
7021     return 0;
95cf }
```

1.9 Dictionary of Basic Factors

```
427e // Created by calabash_boy on 2019/10/28.
427e // CF 100962D 求区间border series，最大border。
b54d #pragma GCC optimize(3)
302f #include <bits/stdc++.h>
18f5 #define rank rkrkrk
421c using namespace std;
96ad const int maxn = 4e5 + 100;
8b50 const int maxlog = 19;
5690 struct Sequence{
bd2b     /** l + k*d <=r **/
6bee     int l,r,d;
f41b     Sequence(int ll = 0,int rr = 0,int dd = 0){
9e51         l = ll;r = rr;d = dd;
95cf     }
3e46     Sequence(const vector<int> & pos){
83e6         if (pos.empty()){
7629             l = r = d = 0;
5620         }else if (pos.size() == 1){
651d             l = pos.front();
8790             r = pos.front();
4753             d = 1;
8e2e         }else{
651d             l = pos.front();
603d             r = pos.back();
7cad             d = pos[1] - pos[0];
95cf         }
95cf     }
b065     bool has(int x){
5ca2         return d and x >= l and x <= r and x % d == l % d;
95cf     }
5bcc     int count(){
c7de         if (d == 0)return 0;
9916         return (r - l) / d + 1;
95cf     }
4083     vector<int> to_list(){
dc6c         vector<int> list(0);
02b2         if (d == 0)return list;
7a3e         for (int i=l;i<=r;i+=d)list.push_back(i);
7b4d         return list;
95cf     }
329b };
a950 Sequence operator -(int X, Sequence S){return Sequence(X - S.r,X - S.l, S.d);}
3997 Sequence operator -(Sequence S, int X){return Sequence(S.l - X, S.r - X, S.d);}
4ad6 Sequence operator &(Sequence S1, Sequence S2){
55d2     int cnt1 = S1.count(), cnt2 = S2.count();
8ea9     if (cnt1 == 0 || cnt2 == 0) return Sequence(0,0,0);
433a     if (cnt1 > cnt2){
8b8f         swap(S1,S2);swap(cnt1,cnt2);
95cf     }
06db     if (cnt1 < 3){
8e82         vector<int> pos(0);
fd94         for (int x : S1.to_list()){
9391             if (S2.has(x)) pos.push_back(x);
95cf         }
94cb         return Sequence(pos);
8e2e     }else{
912e         if (S1.d == S2.d){
2db6             int l = max(S1.l,S2.l), r = min(S1.r,S2.r);
2346             if (r >= l && S1.l % S1.d == S2.l % S1.d)return Sequence(l,r,S1.d);
bf46             else return Sequence(0,0,0);
ddc2         }else assert(0);
95cf     }
95cf }
2f33 struct Run{
427e     //S[l,r] is a run of period of length d.
6bee     int l,r,d;
fc18     Run(int ll = 0,int rr = 0,int dd = 0){
9e51         l = ll;r = rr;d = dd;
95cf     }
04f0     bool operator < (const Run &other)const{
fc87         if (l != other.l)return l < other.l;
7836         if (r != other.r)return r < other.r;
a241         if (d != other.d)return d < other.d;
438e         return false;
95cf     }
b8bc     bool operator == (const Run &other)const{
98a4         return l == other.l and r == other.r and d == other.d;
```

```
95cf         }
329b     };
dcc9     struct Dictionary_of_Basic_Factories{
6ebf         /** 1-base **/
4d80         int name[maxn][maxlog];int n;
64b3         vector<vector<int> > pos[maxlog];
6e4f         int cntA[maxn],cntB[maxn],tsa[maxn],A[maxn],B[maxn];
a540         int sa[maxn],rank[maxn];
c3e5         int height[maxn];
1126         void clear(){
26c3             for (int i=0;i<=max(n,'z' + 10);i++){
0808                 cntA[i] = cntB[i] = tsa[i] = A[i] = B[i] = sa[i] = rank[i] = height[
                       i] = 0;
d71d                 for (int k=0;k<maxlog;k++)name[i][k] = 0;
95cf             }
67d7             for (int k=0;k<maxlog;k++)pos[k].clear();
95cf         }
8ea9         void init(char * ch,int n){
316e             ch[0] = ch[n+1] = '#'; this->n = n;
e86b             for (int i=1;i<=n;i++)cntA[ch[i]]++;
a49f             for (int i=1;i<maxn;i++)cntA[i] += cntA[i-1];
625e             for (int i=n;i>=1;i--)sa[cntA[ch[i]]--] = i;
c9f2             rank[sa[1]] = 1;
a5c5             for (int i=2;i<=n;i++){
dc5c                 rank[sa[i]] = rank[sa[i-1]];
459c                 if (ch[sa[i]] != ch[sa[i-1]])rank[sa[i]] ++;
95cf             }
b7da             pos[0].resize(rank[sa[n]] + 1,vector<int>(0));
6dbf             for (int i=1;i<=n;i++){
fa1a                 name[i][0] = rank[i];
b708                 pos[0][rank[i]].push_back(i);
95cf             }
a867             for (int step = 1,l=1;l <= n;l<<=1,step ++){
c794                 for (int i=0;i<=n;i++)cntA[i] = cntB[i] = 0;
6dbf                 for (int i=1;i<=n;i++){
d9ab                     cntA[A[i] = rank[i]] ++;
c846                     cntB[B[i]=(i+l<=n)?rank[i+l]:0]++;
95cf                 }
72d7                 for (int i=1;i<=n;i++)cntB[i] += cntB[i-1];
4c62                 for (int i=n;i>=1;i--)tsa[cntB[B[i]]--] = i;
c35a                 for (int i=1;i<=n;i++)cntA[i] += cntA[i-1];
1626                 for (int i=n;i>=1;i--)sa[cntA[A[tsa[i]]]--] = tsa[i];
c9f2                 rank[sa[1]] = 1;
a5c5                 for (int i=2;i<=n;i++){
```

```
                    rank[sa[i]] = rank[sa[i-1]];                                      dc5c
                    if (A[sa[i]] != A[sa[i-1]] || B[sa[i]] != B[sa[i-1]])rank[sa[i]]  021c
                       ++;
                }                                                                     95cf
                pos[step].resize(rank[sa[n]] + 1,vector<int> (0));                    4d1e
                for (int i = 1;i <=n;i++){                                            6dbf
                    name[i][step] = rank[i];                                          c0e3
                    pos[step][rank[i]].push_back(i);                                  5a42
                }                                                                     95cf
            }                                                                         95cf
        }                                                                             95cf
        void get_height(char *ch, int n){                                             a8cb
                                                                                      427e
            sa[0] = rank[0] = 0;                                                      5c18
            for (int i=1,j=0;i<=n;i++){                                               0956
                if (j)j--;                                                            1a82
                while (ch[i+j] == ch[sa[rank[i]-1] +j])j++;                           757e
                height[rank[i]]= j;                                                   24a7
            }                                                                         95cf
        }                                                                             95cf
        // get sequence [2^step,2^(step+1))                                           427e
        Sequence get_seq(vector<int> & list, int l, int r){                           7757
            vector<int> pos(0);                                                       8e82
            int idx = lower_bound(list.begin(),list.end(),l) - list.begin();          5dee
            while (idx < list.size() && pos.size() < 3 && list[idx] <= r){            6a21
                pos.push_back(list[idx]);idx ++;                                      7d69
            }                                                                         95cf
            if (pos.size() < 3)return Sequence(pos);                                  8d9e
            else{                                                                     037f
                int last = upper_bound(list.begin(),list.end(),r) - list.begin() -    b0ae
                  1;
                int L = pos.front(), d = pos[1] - pos[0], R = list[last];             88c2
                return Sequence(L,R,d);                                               7985
            }                                                                         95cf
        }                                                                             95cf
        Sequence get_border(int l,int r,int step){                                    25d0
            int len = r - l + 1;                                                      937e
            int baby = 1 << step, giant = min(len-1,(baby * 2-1));                    701e
            int namel = name[l][step], namer = name[r - baby + 1][step];              ed64
            Sequence seql = get_seq(pos[step][namel],r - giant + 1,r - baby + 1),     681e
                    seqr = get_seq(pos[step][namer],l,l + giant - baby);              987b
            seql = (r + 1) - seql; seqr = seqr - (l -baby);                           4d27
            return  seql & seqr;                                                      8804
        }                                                                             95cf
```

```
    /** return O(logn) border series of S[l,r].
     * Attention: can contain empty sequence (0,0,0)
     * if [2^i,2^(i+1)) border does not exist.*/
    vector<Sequence> get_border_series(int l,int r){
        vector<Sequence> ret(0);
        for (int step = 0;(1<<step) < r - l + 1;step++){
            ret.push_back(get_border(l,r,step));
        }
        return ret;
    }
    int get_biggest_border(int l,int r){
        int len = r - l + 1;
        for (int k = maxlog - 1; k >=0;k--){
            if ((1<<k) >= len)continue;
            Sequence seq = get_border(l,r,k);
            if (seq.r)return seq.r;
        }
        return 0;
    }
    int lcp(int x,int y){
        int len = 0;
        for (int k = maxlog-1;k>=0;k--){
            int LEN = 1<< k;
            if (x + LEN - 1 <= n and y + LEN - 1 <= n and name[x][k] == name[y][k]){
                len += 1<<k;
                x += 1<<k;
                y += 1<<k;
            }
        }
        return len;
    }
    int lcs(int x,int y){
        int len = 0;
        for (int k = maxlog-1;k>=0;k--){
            int LEN = 1 << k;
            if (x >= LEN and y >= LEN and name[x - LEN + 1][k] == name[y - LEN + 1][k]){
                len += LEN;
                x -= LEN;
                y -= LEN;
            }
        }
        return len;
    }
    vector<Run> get_all_runs(){
        // cerr<<n<<endl;
        vector<Run> run_list(0);
        for (int per = 1; per * 2 <= n; per ++){
            for (int pos = per;pos <= n;pos += per){
                int left = lcs(pos,pos + per);
                int right = lcp(pos,pos + per);
                if (left + right > per){
                    run_list.push_back(Run(pos - left + 1,pos + per + right - 1,per));
                }
            }
        }
        vector<Run> result(0);
        pair<int,int> pre = {-1,-1};
        for (auto run : run_list){
            pair<int,int> now = {run.l,run.r};
            if (pre != now){
                pre = now;
                result.push_back(run);
            }
        }
        return result;
    }
}dbf;
char s[maxn];
int n,q;
int main(){
    scanf("%d%d",&n,&q);
    scanf("%s",s + 1);
    dbf.init(s,n);
    while (q--){
        int l,r;
        scanf("%d%d",&l,&r);
        printf("%d\n",dbf.get_biggest_border(l,r));
    }
    return 0;
}
```

2 String Automaton

2.1 ACAM

```
427e // Created by calabash_boy on 18-6-5.
427e // HDU 6138
427e //给定若干字典串。
427e // query:strx stry 求最长的p,p为strx、stry子串，且p为某字典串的前缀
302f #include<bits/stdc++.h>
421c using namespace std;
52c1 const int maxn = 1e5+100;
6b3e struct Aho_Corasick_Automaton{
427e     //basic
141b     int nxt[maxn*10][26],fail[maxn*10];
7a04     int root,tot;
427e     //special
8f42     int flag[maxn*10];
d3a5     int len[maxn*10];
1126     void clear(){
21a1         memset(nxt[0],0,sizeof nxt[0]);
0ae1         root = tot=0;
95cf     }
ee91     int newnode(){
71cf         tot++;
87f4         memset(nxt[tot],0,sizeof nxt[tot]);
a231         flag[tot] = len[tot]=0;
91fb         return tot;
95cf     }
9bb4     void insert(char *s ){
8f56         int now = root;
f205         while (*s){
e37a             int id = *s-'a';
ce8f             if(!nxt[now][id])nxt[now][id] = newnode();
7134             len[nxt[now][id]] = len[now]+1;
6f00             now = nxt[now][id];
95cf         }
95cf     }
bcf9     void insert(string str){
8f56         int now = root;
10ad         for (int i=0;i<str.size();i++){
25da             int id = str[i]-'a';
ce8f             if(!nxt[now][id])nxt[now][id] = newnode();
7134             len[nxt[now][id]] = len[now]+1;
```

```
            now = nxt[now][id];                                          6f00
        }                                                                95cf
    }                                                                    95cf
    void build(){                                                        2114
        fail[root] = root;                                               30ee
        queue<int>Q;Q.push(root);                                        c19d
        while (!Q.empty()){                                              11e5
            int head = Q.front();Q.pop();                                ff8a
            for (int i=0;i<26;i++){                                      414f
                if(!nxt[head][i])continue;                               c591
                int temp = nxt[head][i];                                 762f
                fail[temp] = fail[head];                                 c509
                while (fail[temp]&&!nxt[fail[temp]][i]){                 a7fb
                    fail[temp] = fail[fail[temp]];                       5e80
                }                                                        95cf
                if(head&&nxt[fail[temp]][i])fail[temp] = nxt[fail[temp]][i];   3198
                Q.push(temp);                                            6b09
            }                                                            95cf
        }                                                                95cf
    }                                                                    95cf
    void search(string str,int QID);                                     fddd
    int query(string str,int QID);                                       cf07
}acam;                                                                   5ede
void Aho_Corasick_Automaton::search(string str,int QID) {                1874
    int now = root;                                                      8f56
    for (int i=0;i<str.size();i++){                                      10ad
        int id = str[i]-'a';                                             25da
        now = nxt[now][id];int temp = now;                               b2b6
        while (temp!=root&&flag[temp]!=QID){                             694e
            flag[temp] = QID;                                            22a4
            temp = fail[temp];                                           f597
        }                                                                95cf
    }                                                                    95cf
}                                                                        95cf
int Aho_Corasick_Automaton::query(string str, int QID) {                 126b
    int ans =0;int now = root;                                           81f4
    for (int i=0;i<str.size();i++){                                      10ad
        int id = str[i]-'a';                                             25da
        now = nxt[now][id];                                              6f00
        int temp = now;                                                  c20a
        while (temp!=root){                                              dead
            if(flag[temp]==QID){                                         497d
                ans = max(ans,len[temp]);                                79cd
                break;                                                   6173
95cf                }
f597                temp = fail[temp];
95cf            }
95cf        }
4206        return ans;
95cf    }
fae2    string a[maxn];
24df    int m,n,qid;
3117    int main(){
42db        int T;cin>>T;
60ca        while (T—){
67f3            acam.clear();cin>>n;
6dbf            for (int i=1;i<=n;i++){
879c                cin>>a[i];
e321                acam.insert(a[i]);
95cf            }
1ccd            acam.build();cin>>m;
e052            for (int i=1;i<=m;i++){
74ca                int x,y;cin>>x>>y;
6a4f                qid++;
071c                acam.search(a[x],qid);
c2f3                int ans = acam.query(a[y],qid);
d592                cout<<ans<<endl;
95cf            }
95cf        }
7021        return 0;
95cf    }
```

2.2 SAM

```
427e    // Created by calabash_boy on 18–6–4.
427e    //SPOJ substring
427e    // calc ans_i=长度=i的所有子串，出现次数最多的一种出现了多少次。
302f    #include<bits/stdc++.h>
374e    #define RIGHT
427e    //RIGHT: parent树的dfs序上主席树，求每个点的Right集合
421c    using namespace std;
40fb    const int maxn = 25e4+100;
dd0f    #ifdef RIGHT
d273    struct Node{int L,R,val;}Tree[maxn*40];
6207    struct Chairman_Tree{
8abb        int cnt = 0;
```

```
    int root[maxn*2];  bd4f
    void init(){  5d53
        memset(root,0,sizeof root);  a4f5
        cnt =0;  8766
    }  95cf
    /* 建T0空树 */  94cf
    int buildT0(int l, int r){  cf84
        int k = cnt++;  64f2
        Tree[k].val =0;  ecaf
        if (l==r) return k;  eb40
        int mid = l+r >>1;  b8b7
        Tree[k].L = buildT0(l, mid);Tree[k].R = buildT0(mid + 1, r);  0bf4
        return k;  e27b
    }  95cf
    /* 上一个版本节点P, 【ppos】+=del 返回新版本节点*/  e965
    int update (int P,int l,int r,int ppos,int del){  3a6b
        assert(cnt < maxn*50);  d4b1
        int k = cnt++;  64f2
        Tree[k].val = Tree[P].val +del;  73d2
        if (l==r) return k;  eb40
        int mid = l+r >>1;  b8b7
        if (ppos<=mid){  4af7
            Tree[k].L = update(Tree[P].L,l,mid,ppos,del);  5b36
            Tree[k].R = Tree[P].R;  de01
        }else{  8e2e
            Tree[k].L = Tree[P].L;  0d44
            Tree[k].R = update(Tree[P].R,mid+1,r,ppos,del);  a179
        }  95cf
        return k;  e27b
    }  95cf
    int query(int PL,int PR,int l,int r,int L,int R){  b13a
        if (l>R || L>r)return 0;  b8e7
        if (L <= l && r <= R)return Tree[PR].val – Tree[PL].val;  03d9
        int mid = l + r >> 1;  b8b7
        return query(Tree[PL].L,Tree[PR].L,l,mid,L,R) + query(Tree[PL].R,Tree[PR  ff4f
          ].R,mid+1,r,L,R);
    }  95cf
}tree;  b0c1
#endif  1937
char s[maxn];int n,ans[maxn];  6f83
/*注意需要按l将节点基数排序来拓扑更新parent树*/  8a63
struct Suffix_Automaton{  3e3e
    //basic  427e
    int nxt[maxn*2][26],fa[maxn*2],l[maxn*2];  0037
```

```
0db0      int last,cnt;
427e      //extension
f6ac      int cntA[maxn*2],A[maxn*2];/*辅助拓扑更新*/
b0fc      int num[maxn*2];/*每个节点代表的所有串的出现次数*/
dd0f  #ifdef RIGHT
0641      vector<int> E[maxn*2];
6561      int dfsl[maxn*2],dfsr[maxn*2],dfn;
4296      int pos[maxn*2];
efe5      int end_pos[maxn*2];//1基
1937  #endif
c75a      Suffix_Automaton(){ clear(); }
1126      void clear(){
651a          last =cnt=1;
63e2          fa[1]=l[1]=0;
9b85          memset(nxt[1],0,sizeof nxt[1]);
95cf      }
e798      void init(char *s){
f205          while (*s){
d3f9              add(*s-'a');s++;
95cf          }
95cf      }
681b      void add(int c){
a4cf          int p = last;
4428          int np = ++cnt;
8b9f          memset(nxt[cnt],0,sizeof nxt[cnt]);
97c0          l[np] = l[p]+1;last = np;
b7f5          while (p&&!nxt[p][c])nxt[p][c] = np,p = fa[p];
fdc4          if (!p)fa[np]=1;
037f          else{
5740              int q = nxt[p][c];
d84d              if (l[q]==l[p]+1)fa[np] =q;
037f              else{
2401                  int nq = ++ cnt;
bc67                  l[nq] = l[p]+1;
da26                  memcpy(nxt[nq],nxt[q],sizeof (nxt[q]));
66a6                  fa[nq] =fa[q];fa[np] = fa[q] =nq;
5dc1                  while (nxt[p][c]==q)nxt[p][c] =nq,p = fa[p];
95cf              }
95cf          }
95cf      }
2114      void build(){
4006          memset(cntA,0,sizeof cntA);
7b40          memset(num,0,sizeof num);
1a84          for (int i=1;i<=cnt;i++)cntA[l[i]]++;
```

```
        for (int i=1;i<=cnt;i++)cntA[i]+=cntA[i-1];                                    856c
        for (int i=cnt;i>=1;i--)A[cntA[l[i]]--] =i;                                     ebb3
        /*更行主串节点*/                                                                  f42d
        int temp=1;                                                                     3c9b
        for (int i=0;i<n;i++){                                                          1294
            num[temp = nxt[temp][s[i]-'a'] ]=1;                                         3bd2
        }                                                                               95cf
        /*拓扑更新*/                                                                      e1a0
        for (int i=cnt;i>=1;i--){                                                       5258
            //basic                                                                     427e
            int x = A[i];                                                               b7fa
            num[fa[x]]+=num[x];                                                         32d6
            //special                                                                   427e
            ans[l[x]] = max(ans[l[x]],num[x]);                                          f982
        }                                                                               95cf
        //special                                                                       427e
        for (int i=l[last];i>1;i--){                                                    66f2
            ans[i-1] = max(ans[i-1],ans[i]);                                            88a3
        }                                                                               95cf
    }                                                                                   95cf
                                                                                        427e
#ifdef RIGHT                                                                            dd0f
    int get_right_between(int u,int l,int r){                                           a1e1
        return tree.query(tree.root[dfsl[u] - 1],tree.root[dfsr[u]],1,::n,l,r);         64ba
    }                                                                                   95cf
    void dfs(int u){                                                                    d714
        dfsl[u] = ++ dfn;                                                               2b56
        pos[dfn] = u;                                                                   98d9
        for (int v : E[u]){                                                             2c0f
            dfs(v);                                                                     5f3c
        }                                                                               95cf
        dfsr[u] = dfn;                                                                  64a8
    }                                                                                   95cf
    void extract_right(){                                                               0350
        int temp = 1;                                                                   3c9b
        for (int i=0;i<n;i++){                                                          1294
            temp = nxt[temp][s[i] - 'a'];                                               ac16
            end_pos[temp] = i+1;                                                        6940
        }                                                                               95cf
        for (int i=2;i<=cnt;i++){                                                       f6b7
            E[fa[i]].push_back(i);                                                      5e80
        }                                                                               95cf
        dfn = 0;                                                                        0426
        dfs(1);                                                                         dcdd
```

```
5087            tree.root[0] = tree.buildT0(1,n);
7b35            for (int i=1;i<=cnt;i++){
cda5                int u = pos[i];
1c34                if (end_pos[u]){
9965                    int idx = end_pos[u];
b360                    tree.root[i] = tree.update(tree.root[i-1],1,n,idx,1);
8e2e                }else{
d757                    tree.root[i] = tree.root[i-1];
95cf                }
95cf            }
95cf        }
1937 #endif
56dd        void debug(){
5258            for (int i=cnt;i>=1;i--){
01ab                printf("num[%d]=%d␣l[%d]=%d␣fa[%d]=%d\n",i,num[i],i,l[i],i,fa[i]);
95cf            }
95cf        }
5eed }sam;
3117 int main(){
587c     scanf("%s",s);
aaa0     /* calc n must before sam.init()*/
5264     n = strlen(s);
84b5     sam.init(s);
bb59     sam.build();
6dbf     for (int i=1;i<=n;i++){
6240         printf("%d\n",ans[i]);
95cf     }
7021     return 0;
95cf }
```

2.3 Generlized SAM

```
427e // Created by calabash_boy on 19-4-5.
427e //wf2019 first of her name
427e //build sam using trie
302f #include<bits/stdc++.h>
421c using namespace std;
94a1 const int maxn = 1e6+100;
4085 typedef long long ll;
3e3e struct Suffix_Automaton{
0037     int nxt[maxn*2][26],fa[maxn*2],l[maxn*2];
0db0     int last,cnt;
    vector<int> E[maxn*2];                                  0641
    int Num[maxn*2];                                        61cb
    Suffix_Automaton(){ clear(); }                          c75a
    void clear(){                                           1126
        last =cnt=1;                                        651a
        fa[1]=l[1]=0;                                       63e2
        memset(nxt[1],0,sizeof nxt[1]);                     9b85
    }                                                       95cf
    int add(int pre,int c,int num){                         6cab
        last = pre;                                         2d24
        int p = last;                                       a4cf
        int np = ++cnt;                                     4428
        Num[np] = num;                                      b844
        memset(nxt[cnt],0,sizeof nxt[cnt]);                 8b9f
        l[np] = l[p]+1;last = np;                           97c0
        while (p&&!nxt[p][c])nxt[p][c] = np,p = fa[p];      b7f5
        if (!p)fa[np]=1;                                    fdc4
        else{                                               037f
            int q = nxt[p][c];                              5740
            if (l[q]==l[p]+1)fa[np] =q;                     d84d
            else{                                           037f
                int nq = ++ cnt;                            2401
                l[nq] = l[p]+1;                             bc67
                memcpy(nxt[nq],nxt[q],sizeof (nxt[q]));     da26
                fa[nq] =fa[q];fa[np] = fa[q] =nq;           66a6
                while (nxt[p][c]==q)nxt[p][c] =nq,p = fa[p]; 5dc1
            }                                               95cf
        }                                                   95cf
        return np;                                          597e
    }                                                       95cf
    int dfsl[maxn*2],dfsr[maxn*2];                          b432
    int dfn = 0;                                            b4c2
    ll sum[maxn*2];                                         45bd
    void dfs(int u){                                        d714
        dfsl[u] = ++dfn;                                    2b56
        sum[dfn] = Num[u];                                  445a
        for (int v : E[u]){                                 2c0f
            dfs(v);                                         5f3c
        }                                                   95cf
        dfsr[u] = dfn;                                      64a8
    }                                                       95cf
    void build(){                                           2114
        for (int i=2;i<=cnt;i++){                           f6b7
            E[fa[i]].push_back(i);                          5e80
```

```
95cf         }
dcdd         dfs(1);
7b35         for (int i=1;i<=cnt;i++){
036a             sum[i] += sum[i-1];
95cf         }
95cf     }
c250     void query(char * s){
3c9b         int temp = 1;
f205         while (*s){
6147             int ch = *s - 'A';
323f             if (!nxt[temp][ch]){
3257                 printf("0\n");
4f2d                 return;
95cf             }
9439             temp = nxt[temp][ch];
85be             s++;
95cf         }
a64e         ll ans = sum[dfsr[temp]] - sum[dfsl[temp] - 1];
8542         printf("%lld\n",ans);
95cf     }
5eed }sam;
a281 struct Trie{
f142     int Root = 1;
e317     int cnt = 2;
e2e6     int nxt[maxn][26];
dd2d     int num[maxn];
75bc     int sam_pos[maxn];
1f95     int add(int p,int ch){
2e0c         if (!nxt[p][ch]){
621d             nxt[p][ch] = cnt++;
95cf         }
86e9         int now = nxt[p][ch];
e204         num[now] ++;
7d47         return now;
95cf     }
06b4     void bfs(){
aafa         queue<int> Q;
4ad5         Q.push(1);
4f25         sam_pos[1] = 1;
11e5         while (!Q.empty()){
fda7             int head = Q.front();
f2f8             Q.pop();
414f             for (int i=0;i<26;i++){
c591                 if (!nxt[head][i])continue;
```

```
                int now = nxt[head][i];                                2f97
                sam_pos[now] = sam.add(sam_pos[head],i,num[now]);      7ee9
                Q.push(now);                                           e77a
            }                                                          95cf
        }                                                              95cf
    }                                                                  95cf
}trie;                                                                 1cc7
int trie_pos[maxn];                                                    2616
int main(){                                                            3117
    int n,k;                                                           232a
    scanf("%d%d",&n,&k);                                               9927
    trie_pos[0] = 1;                                                   7b34
    for (int i=1;i<=n;i++){                                            6dbf
        static char s[5];                                              66c9
        int p;                                                         4ec4
        scanf("%s%d",s,&p);                                            66ef
        int ch = s[0] - 'A';                                           d259
        trie_pos[i] = trie.add(trie_pos[p],ch);                        faf2
    }                                                                  95cf
    trie.bfs();                                                        49c4
    sam.build();                                                       bb59
    for (int i=0;i<k;i++){                                             f3ea
        static char t[maxn];                                           8fa9
        scanf("%s",t);                                                 f184
        int N = strlen(t);                                             56bc
        reverse(t,t+N);                                                7bd6
        sam.query(t);                                                  3c43
    }                                                                  95cf
    return 0;                                                          7021
}                                                                      95cf
```

2.4 C-SAM(CDAWG)

```
// Created by calabash_boy on 2019/11/5.                               427e
// 求后缀树的每条边代表字符串的本质不同子串个数的和                           427e
// 后缀树边上字符串unique之后总长度为3*n.                                  427e
// unique的结果与压缩sam上的边相同。                                      427e
#include <bits/stdc++.h>                                               302f
using namespace std;                                                   421c
const int maxn = 5e5 + 100;                                            6f64
typedef long long ll;                                                  4085
struct Suffix_Automaton{                                               3e3e
```

```
0037 int nxt[maxn*2][26],fa[maxn*2],l[maxn*2];
e21c bool vis[maxn*2];
1ca4 int dirNxt[maxn*2][26];
ceb8 int dirLen[maxn*2][26];
6070 int ed[maxn*2];
65a0 vector<pair<int,int> > lens[maxn*2];
0db0 int last,cnt;
1126 void clear(){
651a     last =cnt=1;
63e2     fa[1]=l[1]=0;
9b85     memset(nxt[1],0,sizeof nxt[1]);
95cf }
33b9 void init(string s){
90a8     for (int c : s)add(c - 'a');
0fc1     for (int i=0;i<=cnt;i++){
0cd8         vis[i] = false;
9673         ed[i] = -1;
d6a5         lens[i].clear();
8f42         memset(dirLen[i],0,sizeof dirLen[i]);
da3a         memset(dirNxt[i],0,sizeof dirNxt[i]);
95cf     }
95cf }
681b void add(int c){
a4cf     int p = last;
4428     int np = ++cnt;
8b9f     memset(nxt[cnt],0,sizeof nxt[cnt]);
97c0     l[np] = l[p]+1;last = np;
b7f5     while (p&&!nxt[p][c])nxt[p][c] = np,p = fa[p];
fdc4     if (!p)fa[np]=1;
037f     else{
5740         int q = nxt[p][c];
d84d         if (l[q]==l[p]+1)fa[np] =q;
037f         else{
2401             int nq = ++ cnt;
bc67             l[nq] = l[p]+1;
da26             memcpy(nxt[nq],nxt[q],sizeof (nxt[q]));
66a6             fa[nq] =fa[q];fa[np] = fa[q] =nq;
5dc1             while (nxt[p][c]==q)nxt[p][c] =nq,p = fa[p];
95cf         }
95cf     }
95cf }
5a6f int find_nxt(int u){
868a     int res = -1;
b3c9     for (int ch = 0;ch < 26;ch ++){
```

```
ea10         int v = nxt[u][ch];
e151         if (!v)continue;
ee4b         if (res == -1)res = ch;
4796         else return -1;
95cf     }
244d     return res;
95cf }
d714 void dfs(int u){
6cfd     vis[u] = true;
b3c9     for (int ch = 0;ch < 26;ch ++){
ea10         int v = nxt[u][ch];
e151         if (!v)continue;
384d         if (!vis[v])dfs(v);
c21d         int dirch = find_nxt(v);
4ff2         if(dirch == -1){
31a6             dirNxt[u][ch] = v;
b309             dirLen[u][ch] = 1;
8e2e         }else{
1896             dirNxt[u][ch] = dirNxt[v][dirch];
bc25             dirLen[u][ch] = dirLen[v][dirch] + 1;
95cf         }
a3aa         if (find_nxt(u) == -1 or u == 1)lens[dirNxt[u][ch]].push_back(
                  make_pair(dirLen[u][ch],u == 1?1:l[u] - l[fa[u]]));
329b     };
95cf }
5c37 void build(string s,Suffix_Automaton & sam_t){
3c9b     int temp = 1;
6545     for(int i=0;i<s.length();i++){
ac16         temp = nxt[temp][s[i] - 'a'];
33af         int t = temp;
294a         while (ed[t] == -1 && t != 1){
b78e             ed[t] = i;
44bb             t = fa[t];
95cf         }
95cf     }
19f3     ll ans = 0;
f6b7     for (int i=2;i<=cnt;i++){
e538         if (lens[i].empty())continue;
a8af         int Mx = 0;
df85         for (auto x : lens[i])Mx = max(Mx,x.first);
0bb1         string t = s.substr(ed[i] - Mx + 1,Mx);
8f31         sam_t.clear();
fa94         reverse(t.begin(),t.end());
9fa9         vector<ll> cnt(1,0);
```

```
9a29             for (int c : t){
ab84                 sam_t.add(c - 'a');
596e                 cnt.push_back(cnt.back() + sam_t.l[sam_t.last] - sam_t.l[sam_t.
                       fa[sam_t.last]]);
95cf             }
e33a             for (auto x : lens[i])ans += cnt[x.first] * x.second;
95cf         }
d592         cout<<ans<<endl;
95cf     }
fbe7 }sam,temp_sam;
3117 int main(){
9523     int T;
3f76     cin>>T;
60ca     while (T--){
b301         string s;
d694         cin>>s;
3f76         sam.clear();
84b5         sam.init(s);
7826         sam.dfs(1);
0a7f         sam.build(s,temp_sam);
95cf     }
7021     return 0;
95cf }
```

2.5 PAM

```
427e // Created by calabash_boy on 18-6-4.
427e // BZOJ 3676
427e // calc max(len(t)*cnt(t)) t为s回文子串，cnt(t)=t出现次数
302f #include<bits/stdc++.h>
421c using namespace std;
6428 const int maxn = 3e5+100;
466b struct Palindromic_AutoMaton{
427e     //basic
9f36     int s[maxn],now;
f801     int nxt[maxn][26],fail[maxn],l[maxn],last,tot;
427e     // extension
e216     int num[maxn];/*节点代表的所有回文串出现次数*/
1126     void clear(){
427e         //1节点：奇数长度root 0节点：偶数长度root
78a6         s[0]=l[1]=-1;
b6d0         fail[0] = tot = now =1;
f40b         last = l[0]=0;
21a1         memset(nxt[0],0,sizeof nxt[0]);
9b85         memset(nxt[1],0,sizeof nxt[1]);
95cf     }
61ff     Palindromic_AutoMaton(){clear();}
ca1c     int newnode(int ll){
71cf         tot++;
87f4         memset(nxt[tot],0,sizeof nxt[tot]);
dd2b         fail[tot]=num[tot]=0;
1621         l[tot]=ll;
91fb         return tot;
95cf     }
4284     int get_fail(int x){
8ef1         while (s[now-l[x]-2]!=s[now-1])x = fail[x];
d074         return x;
95cf     }
a791     void add(int ch){
3622         s[now++] = ch;
051b         int cur = get_fail(last);
a980         if(!nxt[cur][ch]){
80d2             int tt = newnode(l[cur]+2);
2f33             fail[tt] = nxt[get_fail(fail[cur])][ch];
01cb             nxt[cur][ch] = tt;
95cf         }
c2d8         last = nxt[cur][ch];num[last]++;
95cf     }
2114     void build(){
427e         //fail[i]<i，拓扑更新可以单调扫描。
0f06         for (int i=tot;i>=2;i--){
925b             num[fail[i]]+=num[i];
95cf         }
6b35         num[0]=num[1]=0;
95cf     }
2e3f     void init(char* ss){
36c9         while (*ss){
884f             add(*ss-'a');ss++;
95cf         }
95cf     }
d155     void init(string str){
10ad         for (int i=0;i<str.size();i++){
e6ef             add(str[i]-'a');
95cf         }
95cf     }
7b0e     long long query();
de71 }pam;
26a1 long long Palindromic_AutoMaton::query(){
8955     long long ret =1;
84e9     for (int i=2;i<=tot;i++){
e902         ret = max(ret,1LL*l[i]*num[i]);
95cf     }
ee0f     return ret;
95cf }
15df char s[maxn];
3117 int main(){
587c     scanf("%s",s);
6780     pam.init(s);
bcac     pam.build();
baad     printf("%lld\n",pam.query());
7021     return 0;
95cf }
```

2.6 区间本质不同子串

```
3829 /* Created by calabash_boy on 19-12-4.
c04c  * tutorial:
9b9b  *https://codeforces.com/blog/entry/62331?tdsourcetag=s_pctim_aiomsg
f2b5  */
302f #include<bits/stdc++.h>
421c using namespace std;
4085 typedef long long ll;
52c1 const int maxn = 1e5 + 100;
ed50 /* 维护最后出现位置在i（左端点）的本质不同串数量 */
c09e struct SegmentTree_Sum{
f820     ll Sum[maxn * 8],Lazy[maxn*8];
1465     void down(int x,int l,int mid,int r){
7c75         Sum[x<<1] += Lazy[x] * (mid - l + 1);
0344         Sum[x<<1|1] += Lazy[x] * (r - mid);
58d1         Lazy[x<<1] += Lazy[x];
e178         Lazy[x<<1|1] += Lazy[x];
ce3b         Lazy[x] = 0;
95cf     }
5326     void up(int x){Sum[x] = Sum[x<<1] + Sum[x<<1|1];}
688c     void update(int x,int l,int r,int L,int R,int val){
f9e7         if (l > R or L > r)return;
0746         if (L <= l and r <= R){
2f19             Sum[x] += 1ll * val * (r - l + 1);
```

```
            Lazy[x] += val;                                         7721
            return;                                                 4f2d
        }                                                           95cf
        int mid = l + r >> 1;down(x,l,mid,r);                       19e9
        update(x<<1,l,mid,L,R,val);update(x<<1|1,mid+1,r,L,R,val);  6b0a
        up(x);                                                      cf00
    }                                                               95cf
    ll query(int x,int l,int r,int L,int R){                        5a84
        if (l > R or L > r)return 0;                                51da
        if (L <= l and r <= R)return Sum[x];                        ce7a
        int mid = l + r >> 1;down(x,l,mid,r);                       19e9
        return query(x<<1,l,mid,L,R) + query(x<<1|1,mid+1,r,L,R);   01f2
    }                                                               95cf
}segtree;                                                           f7fb
struct SegmentTree_Max{                                             65fe
    int Max[maxn*8];                                                1e7b
    void update(int x,int l,int r,int pos,int val){                 67cd
        Max[x] = max(Max[x],val);                                   738a
        if (l == r)return;                                          0eec
        int mid = l + r >> 1;                                       b8b7
        if (pos <= mid)update(x<<1,l,mid,pos,val);                  28a4
        else update(x<<1|1,mid+1,r,pos,val);                        f050
    }                                                               95cf
    int query(int x,int l,int r,int L,int R){                       30b1
        if (l > R or L > r)return -1;                               460f
        if (L <= l and r <= R)return Max[x];                        1339
        int mid = l + r >> 1;                                       b8b7
        return max(query(x<<1,l,mid,L,R),query(x<<1|1,mid+1,r,L,R)); 2a00
    }                                                               95cf
}dfstree;                                                           3689
int n,q;                                                            1ed7
char s[maxn];                                                       15df
ll ans[maxn];                                                       e652
typedef pair<pair<int,int>,int> Query;                              89d9
vector<Query> query;                                                2cc5
struct Suffix_Automaton{                                            3e3e
    int nxt[maxn*2][26],fa[maxn*2],l[maxn*2];                       0037
    int last,cnt;                                                   0db0
    /* 每个color最上边一个点 */                                      73e4
    int up_to[maxn];                                                add3
    /* 是否被染过色 */                                               e197
    bool used[maxn*2];                                              2d93
    Suffix_Automaton(){ clear(); }                                  c75a
    void clear(){                                                   1126
```

```
8bdb         last =cnt=1;fa[1]=l[1]=0;
9b85         memset(nxt[1],0,sizeof nxt[1]);
95cf     }
e798     void init(char *s){
0bef         while (*s){add(*s-'a');s++;}
95cf     }
681b     void add(int c){
a4cf         int p = last;
4428         int np = ++cnt;
8b9f         memset(nxt[cnt],0,sizeof nxt[cnt]);
97c0         l[np] = l[p]+1;last = np;
b7f5         while (p&&!nxt[p][c])nxt[p][c] = np,p = fa[p];
fdc4         if (!p)fa[np]=1;
037f         else{
5740             int q = nxt[p][c];
d84d             if (l[q]==l[p]+1)fa[np] =q;
037f             else{
2401                 int nq = ++ cnt;
bc67                 l[nq] = l[p]+1;
da26                 memcpy(nxt[nq],nxt[q],sizeof (nxt[q]));
66a6                 fa[nq] =fa[q];fa[np] = fa[q] =nq;
5dc1                 while (nxt[p][c]==q)nxt[p][c] =nq,p = fa[p];
95cf             }
95cf         }
95cf     }
0641     vector<int> E[maxn * 2];
14d9     int in[maxn*2],out[maxn*2],dfn;
d714     void dfs(int u){
c964         in[u] = ++dfn;
905c         for (int v:E[u])dfs(v);
5383         out[u] = dfn;
95cf     }
eb55     void gao(){
e3bd         for (int i=2;i<=cnt;i++)E[fa[i]].push_back(i);
dcdd         dfs(1);
922c         for (int i=1,now = 1;i<=n;i++){
fc61             now = nxt[now][s[i] - 'a'];
de8e             assert(l[now] == i);
c927             segtree.update(1,1,n,1,i,1);
228a             int u = now;
9ad1             while (u != 1 and !used[u]){
191b                 used[u] = true;
dd8a                 u = fa[u];
95cf             }
```

```
            while (u != 1){                                                    9b7c
                int cur = dfstree.query(1,1,cnt,in[u],out[u]);                 9ccc
                segtree.update(1,1,n,cur - l[u]+1,cur - l[up_to[cur]],-1);     7534
                swap(up_to[cur],u);                                            8094
            }                                                                  95cf
            dfstree.update(1,1,cnt,in[now],i);                                 1262
            up_to[i] = 1;                                                      82e3
            while (!query.empty() and query.back().first.second == i){         d537
                int l = query.back().first.first;                              ba5e
                int id = query.back().second;                                  778a
                ans[id] = segtree.query(1,1,n,l,i);                            931e
                query.pop_back();                                              cdbf
            }                                                                  95cf
        }                                                                      95cf
    }                                                                          95cf
}sam;                                                                          5eed
int main(){                                                                    3117
    cin>>n>>q;                                                                 9c97
    cin>>s+1;                                                                  499d
    sam.init(s+1);                                                             3eb4
    for (int i=1;i<=q;i++){                                                    949d
        int l,r;                                                               9f6b
        cin>>l>>r;                                                             ad6f
        query.push_back({{l+1,r+1},i});                                        735a
    }                                                                          95cf
    sort(query.begin(),query.end(),[](Query x,Query y){                        a4ee
        return x.first.second > y.first.second;                                3fd9
    });                                                                        b251
    sam.gao();                                                                 1923
    for (int i=1;i<=q;i++){                                                    949d
        cout<<ans[i]<<endl;                                                    ccb3
    }                                                                          95cf
    return 0;                                                                  7021
}                                                                              95cf
```

3 Algorithm

3.1 Geometry

```
#include <bits/stdc++.h>                                                       302f
using namespace std;                                                           421c
const int maxn = 10000 + 50;                                                   ce18
```

```
320e template<class type>
9704 struct point{
ce03     type x,y;
5cb2     point(){};
f40a     point(type x_,type y_):x(x_),y(y_){}
f510     point operator +(const point &p)const {return point(x + p.x,y + p.y);}
3ecb     point operator -(const point &p)const {return point(x - p.x,y - p.y);}
427e     //a related to b
427e     //clockwise : positive
427e     //anti-clockwise : negative
427e     //share a line : zero
dce3     type cross(const point &p)const {return x * p.y - y * p.x;}
a809     type dot(const point &p)const {return x * p.x + y * p.y;}
2f3a     type cross(const point &a,const point &b)const {return (a - *this).cross(b -
             *this);}
7f6b     type dot(const point &a,const point &b)const {return (a - *this).dot(b - *
            this);}
d92f     type sqrLen()const{return this->dot(*this);}
5bed     type sqrDis(const point &p)const {return (p - *this).sqrLen();}
329b };
d7b8 typedef point<long long> pt;
9d10 namespace Geometry{
fd78     const double PI = acos(-1.0);
427e     //res[0]: left most and bottom most
427e     //anti-clockwise
427e     //no three points share one line
427e     //WARN: this function modifies points
2325     vector<pt> Convex_Hull(vector<pt> &points){
8fa3         vector<pt> res(0);
0ca4         assert(points.size() >= 3);
bf80         int idx = 0;
6281         for (int i=1;i<points.size();i++){
28dc             pt temp = points[i];
a34c             pt now = points[idx];
4897             if (temp.x < now.x || temp.x == now.x && temp.y < now.y)idx = i;
95cf         }
8d08         swap(points[idx],points[0]);
9837         sort(points.begin()+1,points.end(),[&](pt x,pt y){
89c2             double cro = points[0].cross(x,y);
69ef             if (cro != 0)return cro > 0;
180e             return points[0].sqrDis(x) < points[0].sqrDis(y);
b251         });
7271         res.push_back(points[0]);
c57e         res.push_back(points[1]);
```

```
        for (int i=2;i<points.size();i++){                                      8316
            pt now = points[i];                                                 b7b9
            while (res.size() >= 2){                                            b94e
                double cro = res[res.size()-2].cross(now,res.back());           df0d
                auto p = res[res.size()-2];                                     f72d
                auto pp = res.back();                                           e810
                if (cro >= 0)res.pop_back();                                    63f2
                else break;                                                     caf8
            }                                                                   95cf
            res.push_back(now);                                                 49f1
        }                                                                       95cf
        return res;                                                             244d
    }                                                                           95cf
    //calc the Minkowski Sum of two Convex Hull                                 427e
    vector<pt> Minkowski(const vector<pt> &ch1,const vector<pt> &ch2){          d0a9
        assert(ch1.size() >= 3);                                                ef50
        assert(ch2.size() >= 3);                                                ff7e
        stack<pt> vec1;                                                         7c15
        stack<pt> vec2;                                                         ee2b
        for (int i = ch1.size() - 1;i >=0;i--){                                 7245
            vec1.push(ch1[(i+1)%ch1.size()] - ch1[i]);                          a9f5
        }                                                                       95cf
        for (int i = ch2.size() - 1;i >= 0;i--){                                3cde
            vec2.push(ch2[(i+1)%ch2.size()] - ch2[i]);                          6f4f
        }                                                                       95cf
        vector<pt> res(0);                                                      8fa3
        res.push_back(ch1.front() + ch2.front());                               2219
        while (!vec1.empty() && !vec2.empty()){                                 186a
            auto v1 = vec1.top();                                               b518
            auto v2 = vec2.top();                                               f296
            long long cro = v1.cross(v2);                                       dca9
            if (cro > 0){                                                       6b8d
                res.push_back(res.back() + v1);                                 0c49
                vec1.pop();                                                     cb19
            }else{                                                              8e2e
                res.push_back(res.back() + v2);                                 0ea2
                vec2.pop();                                                     fe8d
            }                                                                   95cf
        }                                                                       95cf
        while (!vec1.empty())res.push_back(res.back() + vec1.top()),vec1.pop(); 6ca4
        while (!vec2.empty())res.push_back(res.back() + vec2.top()),vec2.pop(); b356
        return Convex_Hull(res);                                                1f73
    }                                                                           95cf
    //logn                                                                      427e
```

```
427e         //wether point in or on convex hull
a023         bool within(pt p,const vector<pt> &ch){
0c3b             assert(ch.size() >= 3);
5221             auto base = ch.front();
d6e7             if (base.cross(p,ch[1]) > 0 || base.cross(p,ch.back()) < 0)return false;
684c             if (base.cross(p,ch[1]) == 0 && (p - base).sqrLen() <= (ch[1] - base).
                   sqrLen())return true;
265b             auto cmp = [&](const pt x,const pt y){
d8cd                 long long cro = base.cross(x,y);
61b4                 return cro>0;
329b             };
d4ae             int i = lower_bound(ch.begin(),ch.end(),p,cmp) - ch.begin() - 1;
8132             int j = i+1;
635b             assert(j < ch.size());
c740             return ch[i].cross(ch[j],p) >= 0;
95cf         }
329b     };
```

3.2 Max Flow

```
427e     // Created by calabash_boy on 18-9-14.
302f     #include<bits/stdc++.h>
421c     using namespace std;
4085     typedef long long ll;
32d7     const int maxn = 11000;
3378     const int maxm = 110000;
08a4     const int INF = 0x3f3f3f3f;
5650     struct Max_Flow{
f1b1         int first[maxn],nxt[maxm*2],des[maxm*2],c[maxm*2],tot;
4e95         int dep[maxn];int ss,tt;
b376         Max_Flow(){ clear(); }
1126         void clear(){
4e61             memset(first,-1,sizeof first);tot =-1;
95cf         }
4a69         inline void addEdge(int u,int v,int w){
71cf             tot++;
73e4             des[tot] = v;c[tot] =w;
6570             nxt[tot] = first[u];first[u] = tot;
95cf         }
1836         bool bfs(){
d568             memset(dep,-1,sizeof dep);
0881             dep[ss] =0;
```

```
            queue<int> Q;Q.push(ss);                                            fc6b
            while (!Q.empty()){                                                 11e5
                int q = Q.front();Q.pop();                                      d7b1
                for (int t = first[q];t!=-1;t= nxt[t]){                         9c72
                    int v = des[t],cx = c[t];                                   b7bb
                    if (dep[v]==-1&&cx){                                        c804
                        dep[v] = dep[q]+1;                                      31e8
                        Q.push(v);                                              78e5
                    }                                                           95cf
                }                                                               95cf
            }                                                                   95cf
            return dep[tt]!=-1;                                                 45fe
        }                                                                       95cf
        int dfs(int node,int now){                                              c29e
            if (node==tt)return now;                                            0031
            int res =0;                                                         5839
            for (int t = first[node];t!=-1&&res<now;t=nxt[t]){                  1e7e
                int v = des[t],cx = c[t];                                       b7bb
                if (dep[v]==dep[node]+1&&cx){                                   da1a
                    int x = min(cx,now-res);                                    223c
                    x = dfs(v,x);                                               6c2e
                    res+=x;c[t]-=x;c[t^1]+=x;                                   29d4
                }                                                               95cf
            }                                                                   95cf
            if (!res) dep[node] = -2;                                           7399
            return res;                                                         244d
        }                                                                       95cf
        // tuple<from,to,flow>                                                  427e
        void init(vector<tuple<int,int,int> > Edge){                            4649
            for (auto tp : Edge){                                               1cbd
                int u,v,w;tie(u,v,w) = tp;                                      1de2
                addEdge(u,v,w);addEdge(v,u,0);                                  16fe
            }                                                                   95cf
        }                                                                       95cf
        // s->t max_flow                                                        427e
        ll max_flow(int s,int t){                                               9783
            ss = s;tt = t;                                                      8786
            ll res =0,del =0;                                                   692e
            while (bfs()){while (del = dfs(ss,INF)){res += del;}}               75d3
            return res;                                                         244d
        }                                                                       95cf
    }net;                                                                       8596
    int n,m,s,t;                                                                4dbf
    vector<tuple<int,int,int> > E;                                              8f52
```

```
3117 int main(){
5dae     scanf("%d%d%d%d",&n,&m,&s,&t);
356f     for (int i=0;i<m;i++){
3676         int u,v,w;
95a1         scanf("%d%d%d",&u,&v,&w);
be22         E.push_back(make_tuple(u,v,w));
95cf     }
08d9     net.init(E);
9560     printf("%lld\n",net.max_flow(s,t));
7021     return 0;
95cf }
```

3.3 Min Cost Max Flow(Min Cost Flow)

```
427e 
427e // Created by calabash_boy on 19-10-5.
302f #include <bits/stdc++.h>
421c using namespace std;
6cca const int maxn = 3 * 250 + 100;
1517 const int maxm = 2 * 250 * 250 + 100;
b9bf const int inf = 10000;
08a4 const int INF = 0x3f3f3f3f;
c6cb struct MCMF{
5217     int ss,tt,dis[maxn],pre[maxn];
4b98     int first[maxn],from[maxm*2],des[maxm*2],nxt[maxm*2],cost[maxm*2],flow[maxm
           *2],tot;
e50d     bool in[maxn];
2826     MCMF(){
1d56         clear();
95cf     }
1126     void clear(){
ee65         tot =-1;
8eac         memset(first,-1,sizeof first);
95cf     }
427e     // <u,v,flow,cost>
d399     void init(vector<tuple<int,int,int,int> > E){
757c         for (auto edge : E){
4240             int u,v,f,c;
231d             tie(u,v,f,c) = edge;
b841             addEdge(u, v, f, c);
95cf         }
95cf     }
dbb4     void __addE(int x,int y,int f,int c){
71cf         tot++;
575f         from[tot] =x;des[tot] =y;
4b45         flow[tot] =f;cost[tot] =c;
6d84         nxt[tot] = first[x];first[x] = tot;
95cf     }
f1f8     inline void addEdge(int x,int y,int f,int c){
f355         __addE(x,y,f,c);__addE(y,x,0,-c);
95cf     }
3c52     bool spfa(){
f25d         memset(in,0,sizeof in);
a9d8         for (int i=0;i<maxn;i++)dis[i] = INF;
56b2         memset(pre,-1,sizeof pre);
9669         dis[ss] =0;in[ss] =1;
fc6b         queue<int> Q;Q.push(ss);
11e5         while (!Q.empty()){
3b29             int q = Q.front();
f56a             Q.pop();in[q] = 0;
9c72             for (int t = first[q];t!=-1;t = nxt[t]){
4993                 int v=des[t],len=cost[t],cx=flow[t];
50ae                 if (cx&&dis[v]>dis[q]+len){
e29b                     dis[v] = dis[q]+len;
0986                     pre[v] = t;
7476                     if (!in[v]){
d143                         Q.push(v);in[v] = 1;
95cf                     }
95cf                 }
95cf             }
95cf         }
427e         // min cost max flow
427e         //return pre[tt] != -1;
427e 
427e         // min cost. flow needn't be max.
5287         return pre[tt]!=-1 && dis[tt] < 0;
95cf     }
427e     // <flow,cost>
ae82     pair<int,int> run(int s, int t){
8786         ss =s;tt=t;
eb96         int totflow =0,totcost =0,nowflow =0,nowcost =0;
22dc         while (spfa()){
2c90             nowcost =0;nowflow = INF;
d3ff             int now =pre[tt];
21b8             while (now!=-1){
f5f6                 nowflow = min(nowflow,flow[now]);
61af                now = pre[from[now]];
95cf            }
83dd            now = pre[tt];
21b8            while (now!=-1){
1839                flow[now] -= nowflow;
fee0                flow[now^1] += nowflow;
96be                nowcost +=cost[now];
61af                now = pre[from[now]];
95cf            }
db07            nowcost*=nowflow;
9bc4            totflow +=nowflow;
0178            totcost +=nowcost;
95cf        }
9589        return make_pair(totflow,totcost);
95cf    }
427e    // special
0abd    void output(int cost);
70ae }mcmf;
35b8 int n,m;
8960 int a[maxn];
e8ac int id[maxn];
5718 int argvalue[maxn];
a300 vector<string> ans;
c056 void copy(int argid,int val){
3970     stringstream stm;
2fb3     stm<<(char)('a' + argid - 1)<<"="<<val;
e0f6     ans.push_back(stm.str());
95cf }
2def void print(int argid){
3970     stringstream stm;
ab5f     stm<<"print("<<(char)('a' + argid - 1)<<")";
e0f6     ans.push_back(stm.str());
95cf }
5273 void MCMF::output(int cost){
610d     int argid = 0;
6dbf     for (int i=1;i<=n;i++){
3db1         int A = 2 * i-1;
fe76         int B = 2 * i;
3979         if (id[A] == 0){
dbc5             argid ++;
c40b             id[A] = argid;
a4ca             copy(argid, a[i]);
9257             print(argid);
79a3             argvalue[argid] = a[i];
```

```
        }else{                                              8e2e
            int temp_value = argvalue[id[A]];               2c77
            if (temp_value != a[i]){                        080d
                copy(id[A], a[i]);                          16e6
                argvalue[id[A]] = a[i];                     8c83
            }                                               95cf
            print(id[A]);                                   b391
        }                                                   95cf
        for (int t = first[B];t != -1;t = nxt[t]){          2516
            int v = des[t];                                 e8e0
            int f = flow[t];                                2bc8
            if (f|| v == A){                                c8f5
                continue;                                   b333
            }                                               95cf
            if (v == 2 * n + 3) break;                      f914
            else{                                           037f
                id[v] = id[A];                              8919
            }                                               95cf
        }                                                   95cf
    }                                                       95cf
    cout<<ans.size()<<"␣"<<cost<<endl;                      6f76
    for (auto str : ans){                                   03de
        cout<<str<<endl;                                    cc6d
    }                                                       95cf
}                                                           95cf
int main(){                                                 3117
    cin>>n>>m;                                              9af0
    for (int i=1;i<=n;i++){                                 6dbf
        cin>>a[i];                                          879c
    }                                                       95cf
    vector<tuple<int,int,int,int> > E(0);                   efbd
    int SS = 2 * n + 1;                                     f385
    int S = 2 * n + 2;                                      dc84
    int T = 2 * n + 3;                                      c8df
    E.push_back(make_tuple(SS,S,m,0));                      6962
    for (int i=1;i<=n;i++){                                 6dbf
        int A = 2 * i - 1;                                  3db1
        int B = 2 * i;                                      fe76
        E.push_back(make_tuple(A,B,1,-inf));                3531
        E.push_back(make_tuple(S,A,1,__builtin_popcount(a[i])));  1cb5
        E.push_back(make_tuple(B,T,1,0));                   0673
        for (int j=i+1;j<=n;j++){                           ed35
            int AA = 2 * j - 1;                             71ea
            int BB = 2 * j;                                 1e22
            if (a[i] == a[j]){
                E.push_back(make_tuple(B,AA,1,0));
            }else{
                E.push_back(make_tuple(B,AA,1,__builtin_popcount(a[j])));
            }
        }
    }
    mcmf.init(E);
    pair<int,int> ans = mcmf.run(SS, T);
    //cerr<<ans.first<<","<<ans.second<<endl;
    mcmf.output((ans.second% inf + inf) % inf);
    return 0;
}
```

3.4 LCA

```
// Created by calabash_boy on 18-7-7.
#include<bits/stdc++.h>
using namespace std;
const int maxn = 5e5+100;
int first[maxn],des[maxn*2],nxt[maxn*2],tot;
int n,m,s;
inline int addEdge(int x,int y){
    tot++;des[tot] = y;
    nxt[tot] = first[x];
    first[x] = tot;
}
namespace Multiply_LCA{
    int fa[maxn][20],dep[maxn];
    void dfs(int u,int father){
        fa[u][0] = father;
        dep[u] = dep[father]+1;
        for (int i=1;i<20&&fa[u][i-1];i++){
            fa[u][i] = fa[fa[u][i-1]][i-1];
        }
        for (int t=first[u];t;t=nxt[t]){
            int v = des[t];
            if (v==father)continue;
            dfs(v,u);
        }
    }
    int lca(int x,int y){
        if (dep[x]<dep[y])swap(x,y);
        for (int i=19;i>=0;i--){
            if (dep[fa[x][i]]>=dep[y]){
                x = fa[x][i];
            }
        }
        if (x==y)return x;
        for (int i=19;i>=0;i--){
            if (fa[x][i]!=fa[y][i]){
                x = fa[x][i];
                y = fa[y][i];
            }
        }
        return fa[y][0];
    }
};
int main(){
    scanf("%d%d%d",&n,&m,&s);
    for (int i=1;i<n;i++){
        int x,y;
        scanf("%d%d",&x,&y);
        addEdge(x,y);addEdge(y,x);
    }
    Multiply_LCA::dfs(s,0);
    while (m--){
        int x,y;scanf("%d%d",&x,&y);
        printf("%d\n",Multiply_LCA::lca(x,y));
    }
    return 0;
}
```

3.5 DSU On Tree

```
// Created by calabash_boy on 18-10-8.
// 1-rooted tree
// query vertex with height H in subtree of V
// whether the letter can form a palindrome
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef pair<int,int> pii;
#define rep(i,l,r) for (ll i = l,_ = r;i<_;i++)
```

```
5879 #define REP(i,l,r) for (ll i=l,_=r;i<=_;i++)
6f64 const int maxn = 5e5+100;
2ff9 int n,tot,first[maxn],des[maxn],nxt[maxn],m;
28d5 vector<pii> Q[maxn];
f96d int cnt[maxn][26],Cnt[maxn];
bbe3 int sz[maxn],dep[maxn],wson[maxn];
f0f2 bool ans[maxn],big[maxn];
15df char s[maxn];
453e inline void addEdge(int x,int y){
4704     tot++;des[tot] = y;
465b     nxt[tot] = first[x];
86fa     first[x] = tot;
95cf }
0d39 void get_sz(int node,int depth){
2b42     dep[node] = depth;sz[node] = 1;
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
a0d5         get_sz(v,depth+1);
47d5         sz[node] += sz[v];
03ee         if (sz[v] > sz[wson[node]])wson[node] = v;
95cf     }
95cf }
5efd void add(int node,int sign){
b01b     Cnt[dep[node]] -= cnt[dep[node]][s[node]-'a'];
d2e8     cnt[dep[node]][s[node]-'a'] ^=1;
937f     Cnt[dep[node]] += cnt[dep[node]][s[node]-'a'];
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
dcb7         if (big[v])continue;
ec6e         add(v,sign);
95cf     }
95cf }
5cc1 void dfs(int node,bool keep){
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
5279         if (v == wson[node])continue;
4bc1         dfs(v,0);
95cf     }
d010     if (wson[node]){
6048         big[wson[node]]=1;
11b7         dfs(wson[node],1);
95cf     }
7111     add(node,1);
3a0c     for (auto q:Q[node]){
            ans[q.second] = Cnt[q.first] <=1; 1c95
        } 95cf
        if (wson[node])big[wson[node]] = 0; 918e
        if (!keep)add(node,-1); dc2a
} 95cf
int main(){ 3117
    scanf("%d%d",&n,&m); ac98
    REP(i,2,n){ eeaf
        int p; 4ec4
        scanf("%d",&p); e75e
        addEdge(p,i); be80
    } 95cf
    scanf("%s",s+1); a275
    rep(i,0,m){ a826
        int v,h; 8213
        scanf("%d%d",&v,&h); fdd4
        Q[v].push_back({h,i}); 3e7f
    } 95cf
    get_sz(1,1);dfs(1,0); ff05
    rep(i,0,m)printf("%s\n",ans[i]?"Yes":"No"); 8823
    return 0; 7021
} 95cf
```

4 Data Structure

4.1 01 Trie

```
// Created by calabash_boy on 18-7-7. 427e
// max(XorSum(a_l^r)) 427e
#include<bits/stdc++.h> 302f
using namespace std; 421c
const int MAX = 1e6+100; ed66
int bas[35],n,Cas; 80de
const int INF = 2147483645; 92ad
struct Trie{ a281
    int nxt[MAX<<2][2],l[MAX<<2]; abd0
    int cnt,ansl,ansr,ansv; a945
    void init(){ 5d53
        cnt =ansv = 0; 68de
        memset(nxt[0],0,sizeof (nxt[0])); 16d8
        memset(l,0x3f3f3f3f,sizeof(l)); aa76
    } 95cf
```

```
b87c     int create(){
6fb3         cnt++;
3b79         memset(nxt[cnt],0,sizeof (nxt[cnt]));
6808         return cnt;
95cf     }
d5dd     void insert(int id,int x){
875c         int y = 0;
7ecf         for (int i=30;i>=0;i--){
0c9f             int t = x&bas[i];
2e46             t>>=i;
713f             if (!nxt[y][t])nxt[y][t] = create();
f056             y = nxt[y][t];
95cf         }
a4a7         l[y] = min(l[y],id);
95cf     }
1a97     void query(int id,int x){
537e         int y=0;  int res =0;
7ecf         for (int i=30;i>=0;i--){
0c9f             int t = x&bas[i];
2e46             t>>=i;
32ad             if (nxt[y][!t]){
63b9                 y =nxt[y][!t];
1f38                 res+=bas[i];
8e2e             }else{
f056                 y = nxt[y][t];
95cf             }
95cf         }
181d         if (res==ansv){
a404             if (l[y]<ansl){
50d3                 ansl = l[y];  ansr = id;
95cf             }
8135         }else if (res>ansv){
9429             ansv = res;
12f4             ansl = l[y];
37e9             ansr = id;
95cf         }
95cf     }
1cc7 }trie;
3117 int main(){
bf6d     bas[0] = 1;
1b53     for (int i1=1;i1<=30;i1++)bas[i1] = bas[i1-1]<<1;
3cb5     scanf("%d",&Cas);
3e2f     for (int i=1;i<=Cas;i++){
56d3         trie.init();  trie.insert(0,0);
```

```
        scanf("%d",&n);                                                   cd91
        int sum=0;                                                        4d6a
        for (int j=1;j<=n;j++){                                           ede7
            int ai;                                                       69e6
            scanf("%d",&ai);  sum^=ai;                                    3e9d
            trie.query(j,sum);  trie.insert(j,sum);                       17a6
        }                                                                 95cf
        printf("Case␣#%d:\n%d␣%d\n", i, trie.ansl + 1, trie.ansr);        7351
    }                                                                     95cf
    return 0;                                                             7021
}                                                                         95cf
```

4.2 Cartesian Tree

```
// Created by calabash_boy on 18-7-24.                                    427e
//他的名字是笛卡尔树。                                                      427e
#include<bits/stdc++.h>                                                   302f
using namespace std;                                                      421c
#define OPENSTACK                                                         1585
const int maxn = 1e6+100;                                                 94a1
const int mod = 1e9+7;                                                    5d33
typedef long long LL;                                                     5cad
int stk[maxn],top,sz[maxn];                                               a8dc
int l[maxn],r[maxn],rt,n;                                                 8f18
pair<int,int> a[maxn];                                                    62bd
LL inv[maxn],fac[maxn],inv_fac[maxn];                                     2b49
bool vis[maxn];                                                           dbd8
/* l 左儿子 r 右儿子 rt根*/                                                 ea2f
void build(){                                                             2114
    top=0;                                                                3e5f
    for (int i=1;i<=n;i++) l[i]=r[i]=vis[i] =0;                           4c1f
    for (int i=1;i<=n;i++){                                               6dbf
        int k = top;                                                      8077
        while (k&&a[i]<a[stk[k-1]])k--;                                   14fa
        if (k) r[stk[k-1]] = i;                                           004e
        if (k<top) l[i] = stk[k];                                         90d1
        stk[k++] =i;top = k;                                              c046
    }                                                                     95cf
    for (int i=1;i<=n;i++) vis[l[i]] = vis[r[i]] =1;                      791b
    for (int i=1;i<=n;i++){                                               6dbf
        if (!vis[i]){                                                     794b
            rt = i;                                                       cf39
6173              break;
95cf          }
95cf      }
95cf  }
a89a  LL power(LL x,LL y){
0aee      LL res =1;
db1a      while (y){
349b          if (y&1)res = res*x%mod;
af39          y>>=1;
df96          x = x*x%mod;
95cf      }
244d      return res;
95cf  }
0f81  inline LL C(int n,int m){
54dd      return fac[n]*inv_fac[m]%mod*inv_fac[n-m]%mod;
95cf  }
f33f  int dfs(int u){
fdf8      sz[u]=1;int ans =1;
fe92      if (l[u])ans=1LL*ans*dfs(l[u])%mod;
429f      if (r[u])ans = 1LL*ans*dfs(r[u])%mod;
2c7a      sz[u]+=sz[l[u]]+sz[r[u]];
b778      return 1LL*ans*C(sz[u]-1,sz[l[u]])%mod;
95cf  }
6e6d  void Main(){
acce      inv[1]=fac[1]=fac[0]=1;
3295      for (int i=2;i<maxn;i++)fac[i] = fac[i-1]*i%mod,inv[i] = inv[mod%i]*(mod-mod
            /i)%mod;
5f9e      inv_fac[maxn-1] = power(fac[maxn-1],mod-2);
c2aa      for (int i=maxn-2;i>=0;i--){
4cf8          inv_fac[i] = inv_fac[i+1]*(i+1)%mod;
95cf      }
d6b7      int T;scanf("%d",&T);
60ca      while (T--) {
cd91          scanf("%d", &n);
6dbf          for (int i = 1; i <= n; i++) {
7681              int x;scanf("%d", &x);
d6d4              a[i] = {-x, i};
95cf          }
7068          build();
b475          printf("%d\n", inv[2] * n % mod * power(fac[n], mod - 2) % mod * dfs(rt)
                  % mod);
95cf      }
95cf  }
3117  int main(){
```

```
#ifdef OPENSTACK                                                     4b95
    int size = 70 << 20; // 256MB                                    90c5
    char *p = (char*)malloc(size) + size;                            9efa
#if (defined _WIN64) or (defined __unix)                             8c82
    __asm__("movq␣%0,␣%%rsp\n" :: "r"(p));                           665b
#else                                                                a8cb
    __asm__("movl␣%0,␣%%esp\n" :: "r"(p));                           355e
#endif                                                               1937
#endif                                                               1937
    Main();                                                          362c
#ifdef OPENSTACK                                                     4b95
    exit(0);                                                         a398
#else                                                                a8cb
    return 0;                                                        7021
#endif                                                               1937
}                                                                    95cf
```

4.3 Chairman Tree

```
// Created by calabash_boy on 18-7-7.                                 427e
// query_kth_element                                                 427e
#include<bits/stdc++.h>                                              302f
using namespace std;                                                 421c
const int maxn=1e5+100;                                              52c1
int a[maxn];int rk[maxn];int pos[maxn];                              b425
int root[maxn];int cnt,m,n,T;                                        15ac
struct Chairman_Tree{                                                6207
    struct Node{int L,R,val;}tree[maxn*500];                         108d
    void init(){                                                     5d53
        memset(root,0,sizeof root);                                  a4f5
        cnt =0;                                                      8766
    }                                                                95cf
    /* 建T0空树 */                                                    94cf
    int buildT0(int l, int r){                                       cf84
        int k = cnt++;                                               64f2
        tree[k].val =0;                                              e9d1
        if (l==r) return k;                                          eb40
        int mid = l+r >>1;                                           b8b7
        tree[k].L = buildT0(l, mid);tree[k].R = buildT0(mid + 1, r); 1e97
        return k;                                                    e27b
    }                                                                95cf
    /* 上一个版本节点P, 【ppos】+=del 返回新版本节点*/                    e965
```

```
3a6b        int update (int P,int l,int r,int ppos,int del){
64f2            int k = cnt++;
1e22            tree[k].val = tree[P].val +del;
eb40            if (l==r) return k;
b8b7            int mid = l+r >>1;
4af7            if (ppos<=mid){
59bb                tree[k].L = update(tree[P].L,l,mid,ppos,del);
1cb7                tree[k].R = tree[P].R;
8e2e            }else{
a8f5                tree[k].L = tree[P].L;
d096                tree[k].R = update(tree[P].R,mid+1,r,ppos,del);
95cf            }
e27b            return k;
95cf        }
4798        int query_kth(int lt,int rt,int l,int r,int k){
9e61            if (l==r) return a[rk[l]];
b8b7            int mid = l+r >>1;
9988            if (tree[tree[rt].L].val-tree[tree[lt].L].val>=k) return query_kth(tree[
                  lt].L,tree[rt].L,l,mid,k);
38e4            else return query_kth(tree[lt].R,tree[rt].R,mid+1,r,k+tree[tree[lt].L].
                  val-tree[tree[rt].L].val);
95cf        }
b0c1    }tree;
56b1    bool cmp(int x,int y){return a[x]<a[y];}
3117    int main() {
1fd9        scanf("%d", &T);
60ca        while (T--) {
ac98            scanf("%d%d",&n,&m);
6dbf            for (int i=1;i<=n;i++){
9a1c                scanf("%d",&a[i]);
f9d0                rk[i]=i;
95cf            }
a475            tree.init();
f0ca            sort(rk+1,rk+1+n,cmp);
8b31            for (int i1=1;i1<=n;i1++){
9b5e                pos[rk[i1]] =i1;
95cf            }
b6a2            root[0] = tree.buildT0(1, n);
8b31            for (int i1=1;i1<=n;i1++){
8294                root[i1] = tree.update(root[i1-1],1,n,pos[i1],1);
95cf            }
3f3a            while (m--){
d32c                int l,r,k;scanf("%d%d%d",&l,&r,&k);
26ab                printf("%d\n",tree.query_kth(root[l-1],root[r],1,n,k));
```

```
            }                                   95cf
        }                                       95cf
        return 0;                               7021
}                                               95cf
```

4.4 KD Tree

```
// Created by calabash_boy on 18-10-6.                427e
#include<bits/stdc++.h>                               302f
using namespace std;                                  421c
typedef long long LL;                                 5cad
const int maxn = 2e5+100;                             eb45
const LL INF = 0x3f3f3f3f3f3f3f3fLL;                  b1ec
int m,n;                                              4d9b
const int demension = 2;                              fc74
struct Hotel{                                         4825
    int pos[demension],id,c;                          b199
}hotel[maxn],kdtree[maxn];                            4922
double var[demension];                                2ece
int split [maxn];int cmpDem;                          8003
bool cmp(const Hotel &a,const Hotel &b){              5cdc
    return a.pos[cmpDem]<b.pos[cmpDem];               b5cd
}                                                     95cf
void build (int l,int r){                             d5af
    if (l>=r)return;                                  2625
    int mid = l+r >>1;                                b8b7
    for (int i=0;i<demension;i++){                    8037
        double ave =0;                                4655
        for (int j=l;j<=r;j++){                       a0d3
            ave+=hotel[j].pos[i];                     70b6
        }                                             95cf
        ave/=(r-l+1);var[i] =0;                       b1eb
        for (int j=l;j<=r;j++){                       a0d3
            var[i]+=pow(hotel[j].pos[i]-ave,2);       27fe
        }                                             95cf
        var[i]/=(r-l+1);                              6e08
    }                                                 95cf
    split[mid] =-1;double maxVar=-1;                  3909
    for (int i=0;i<demension;i++){                    8037
        if (var[i]>maxVar){                           d704
            maxVar = var[i];                          3bdc
            split[mid] =i;                            9c04
        }
    }
    cmpDem = split[mid];
    nth_element(hotel+l,hotel+mid,hotel+r+1,cmp);
    build (l,mid-1);build (mid+1,r);
}
int ansIndex;
LL ansDis;
void query(int l,int r,const Hotel& x){
    if (l>r)return ;
    int mid = l+r >>1;LL dis =0;
    for (int i=0;i<demension;i++){
        dis +=1LL*(x.pos[i]-hotel[mid].pos[i])*(x.pos[i]-hotel[mid].pos[i]);
    }
    if (hotel[mid].c<=x.c){
        if (ansDis == dis && hotel[mid].id<hotel[ansIndex].id){
            ansIndex = mid;
        }else if (dis<ansDis){
            ansDis = dis;
            ansIndex = mid;
        }
    }
    int d = split[mid];
    LL radius = 1LL*(x.pos[d]-hotel[mid].pos[d])*(x.pos[d]-hotel[mid].pos[d]);
    if (x.pos[d]<hotel[mid].pos[d]){
        query(l,mid-1,x);
        if (ansDis>radius){query(mid+1,r,x);}
    }else{
        query(mid+1,r,x);
        if (ansDis>radius){query(l,mid-1,x);}
    }
}
int T;
void input(){
    scanf("%d%d",&n,&m);
    for (int i=0;i<n;i++){
        scanf("%d%d%d",&hotel[i].pos[0],&hotel[i].pos[1],&hotel[i].c);
        hotel[i].id=i;
    }
    build (0,n-1);
}
void solve(){
    Hotel x;
    for (int i=1;i<=m;i++){
        scanf("%d%d%d",&x.pos[0],&x.pos[1],&x.c);
        ansDis = INF;ansIndex =n+1;
        query(0,n-1,x);
        printf("%d␣%d␣%d\n",hotel[ansIndex].pos[0],hotel[ansIndex].pos[1],hotel[
          ansIndex].c);
    }
}
int main(){
    scanf("%d",&T);
    while (T--){
        input();
        solve();
    }
    return 0;
}
```

4.5 Segment Tree

```
// Created by calabash_boy on 18-9-14.
// interval modify & interval query
#include<stdio.h>
using namespace std;
const int maxn = 1e5+100;
typedef long long LL;
int a[maxn];
struct Seg_Tree{
    LL val[maxn*4];LL lazy[maxn*4];
    inline void Up(int x){val[x] = val[x<<1]+val[x<<1|1];}
    inline void Down(int x,int l,int mid,int r){
        if (lazy[x]){
            val[x<<1] += 1LL*lazy[x]*(mid-l+1);
            val[x<<1|1] += 1LL*lazy[x]*(r-mid);
            lazy[x<<1]+= lazy[x];
            lazy[x<<1|1] += lazy[x];
            lazy[x] =0;
        }
    }
    void build (int x,int l,int r){
        lazy[x] =0;
        if (l==r){val[x] = a[l];return ;}
        int mid = l+r >>1;
        build (x<<1,l,mid);build (x<<1|1,mid+1,r);
        Up(x);
    }
    void add(int x,int l,int r,int L,int R,int del){
        if (l>R||r<L)return;
        if (L<=l&&r<=R){
            val[x]+=1LL*del*(r-l+1);
            lazy[x]+=del;
            return;
        }
        int mid = l+r >>1;
        Down(x,l,mid,r);
        add(x<<1,l,mid,L,R,del);add(x<<1|1,mid+1,r,L,R,del);
        Up(x);
    }
    LL query_Sum(int x,int l,int r,int L,int R){
        if (l>R||r<L)return 0;
        if (L<=l&&r<=R)return val[x];
        int mid = l+r >>1;
        Down(x,l,mid,r);
        return query_Sum(x<<1,l,mid,L,R)+query_Sum(x<<1|1,mid+1,r,L,R);
    }
}tree;
char opt[5];int m,n;
int main(){
    scanf("%d%d",&n,&m);
    for (int i=1;i<=n;i++){
        scanf("%d",a+i);
    }
    tree.build(1,1,n);
    while (m--){
        int l,r,v;
        scanf("%s%d%d",opt,&l,&r);
        if (opt[0]=='Q'){
            printf("%I64d\n",tree.query_Sum(1,1,n,l,r));
        }else if (opt[0]=='C'){
            scanf("%d",&v);
            tree.add(1,1,n,l,r,v);
        }
    }
    return 0;
}
```

4.6 AFL(Cactus)

```
// Created by calabash_boy on 18-9-14.
// circle-square-tree Maximum independent set
#include<bits/stdc++.h>
using namespace std;
const int maxn = 1e5+100;
vector<int> E1[maxn],ET[maxn];
int m,n,N,fa[maxn],dp[maxn][2];
int len[maxn],dfn[maxn],dfs_clock;
bool inCircle[maxn];
int dp2[maxn][2];
inline void addEdge1(int x,int y){
    E1[x].push_back(y);
}
inline void addEdgeT(int x,int y){
    ET[x].push_back(y);
}
void input(){
    cin>>n>>m;N =n;
    for (int i=0;i<m;i++){
        int u,v;cin>>u>>v;
        addEdge1(u,v);addEdge1(v,u);
    }
}
void tarjan(int u){
    dfn[u] = ++dfs_clock;
    for (int i=0;i<E1[u].size();i++){
        int v = E1[u][i];
        if (v==fa[u])continue;
        if (!dfn[v]){
            fa[v] = u;tarjan(v);
        }else if (dfn[v]<dfn[u]){
            n++;
            len[n] = dfn[u]-dfn[v]+1;
            fa[n] = v;
            addEdgeT(v,n);
            int temp = u;
            while (temp!=v){
                inCircle[temp] = true;
                addEdgeT(n,temp);
                temp = fa[temp];
            }
        }
95cf      }
aeb9      if (!inCircle[u]){
6225          addEdgeT(fa[u],u);
95cf      }
e88e      dfs_clock--;
95cf  }
662c  void work(int x){
7330      int sz = ET[x].size();
03f3      if (sz==2){
bc63          int son1 = ET[x][0];
e1e3          int son2 = ET[x][1];
ff53          dp[x][0] = dp[son1][0]+dp[son2][0];
95d6          dp[x][1] = max(dp[son1][0]+dp[son2][0],max(dp[son1][0]+dp[son2][1],dp[
                son1][1]+dp[son2][0]));
4f2d          return;
95cf      }
3bde      dp2[0][0] =dp[ET[x][0]][0];dp2[0][1]=0;
e123      for (int i=1;i<sz;i++){
1022          dp2[i][0] = max(dp2[i-1][0],dp2[i-1][1])+dp[ET[x][i]][0];
6ecd          dp2[i][1] = dp2[i-1][0]+dp[ET[x][i]][1];
95cf      }
b6ba      dp[x][0] = dp2[sz-1][0];
cfc2      dp[x][1] = dp2[sz-1][0];
3347      dp2[sz][0]=dp2[sz][1]=0;
ca21      for (int i=sz-1;i>=0;i--){
858a          dp2[i][0] = max(dp2[i+1][0],dp2[i+1][1])+dp[ET[x][i]][0];
6f8c          dp2[i][1] = dp2[i+1][0]+dp[ET[x][i]][1];
95cf      }
5e56      dp[x][1] = max(dp[x][1],max(dp2[0][0],dp2[0][1]));
95cf  }
d714  void dfs(int u){
0799      dp[u][0]=0;dp[u][1]=1;
16e7      if (u>N)dp[u][0]=0;
5ee5      for (int i=0;i<ET[u].size();i++){
f37f          int v = ET[u][i];
5f3c          dfs(v);
2900          if (u<=N){
edd9              dp[u][0]+=max(dp[v][1],dp[v][0]);
2a1b              dp[u][1]+=dp[v][0];
95cf          }
95cf      }
3200      if (u>N)work(u);
95cf  }
3117  int main(){
```

```
    input();                                            2a5c
    tarjan(1);                                          951d
    dfs(1);                                             dcdd
    cout<<max(dp[1][0],dp[1][1])<<endl;                 09a1
    return 0;                                           7021
}                                                       95cf
```

4.7 Segment Tree(Dynamic Memory).cpp

```
// Created by calabash_boy on 18-10-1.                      427e
// CF 1046A                                                 427e
// give n tuple(x,r,p) and k<=20 , calc unordered pair(i,j) 427e
// xi - ri <= xj <= xi + ri                                 427e
// xj - rj <= xi <= xj + rj                                 427e
// |pi - pj| <=k                                            427e
#include <bits/stdc++.h>                                    302f
using namespace std;                                        421c
const int maxn = 1e5+100;                                   52c1
typedef long long ll;                                       4085
struct Node{ int L,R,val; }tree[maxn*200];                  1c06
int cnt;                                                    9f58
struct Segment_Tree{                                        9c29
    int root = 0;                                           e7b0
    int newnode(){                                          ee91
        ++cnt;                                              06cb
        tree[cnt].val = tree[cnt].L = tree[cnt].R = 0;      6598
        return cnt;                                         6808
    }                                                       95cf
    Segment_Tree(){ root = newnode(); }                     1483
    void add(int x,int l,int r,int Pos,int delta){          74ce
        tree[x].val += delta;                               df5d
        if (l == r)return;                                  0eec
        int mid = l+r >>1;                                  b8b7
        if (Pos <= mid){                                    5411
            if (tree[x].L == 0){                            88c7
                tree[x].L = newnode();                      9efd
            }                                               95cf
            add(tree[x].L,l,mid,Pos,delta);                 55fc
        }else{                                              8e2e
            if (tree[x].R == 0){                            e74e
                tree[x].R = newnode();                      ffbb
            }                                               95cf
```

```
492e             add(tree[x].R,mid+1,r,Pos,delta);
95cf         }
95cf     }
30b1     int query(int x,int l,int r,int L,int R){
52df         if (!x)return 0;
b8e7         if (l>R || L>r)return 0;
c450         if (L <= l && r <= R)return tree[x].val;
b8b7         int mid = l+r >>1;
b018         return query(tree[x].L,l,mid,L,R) + query(tree[x].R,mid+1,r,L,R);
95cf     }
329b };
9c0b map<int,Segment_Tree> mp;
9a6f map<int,int> id;
d7af int N;
3117 int main(){
232a     int n,k;
9927     scanf("%d%d",&n,&k);
ad91     vector<tuple<int,int,int> > a(n);
7739     vector<int> nums;
1294     for (int i=0;i<n;i++){
6a6b         int x,r,q;scanf("%d%d%d",&x,&r,&q);
82fb         a[i] = make_tuple(x,r,q);
3bee         nums.push_back(x);
ca6f         nums.push_back(x+r);
4730         nums.push_back(x-r);
95cf     }
19cd     sort(nums.begin(),nums.end());
e5bf     nums.erase(unique(nums.begin(),nums.end()),nums.end());
9e70     for (int i=0;i<nums.size();i++){
9b07         id[nums[i]] = i+1;
95cf     }
34ee     N = nums.size();
4c8a     sort(a.begin(),a.end(),[](const tuple<int,int,int> &a,const tuple<int,int,
           int>&b){
ddfb         return get<1>(a) > get<1>(b);
b251     });
19f3     ll ans =0;
1294     for (int i=0;i<n;i++){
2f4e         int x,r,q;tie(x,r,q) = a[i];
a8aa         int L = id[x-r],R = id[x+r];
af5f         for (int j=q-k;j<=q+k;j++){
7cd6             if (mp.find(j) == mp.end())continue;
8341             Segment_Tree & tree = mp[j];
e7d3             int root = tree.root;
```

```
            ans += tree.query(root,1,N,L,R);    768d
        }                                       95cf
        Segment_Tree & tree = mp[q];            e2c3
        int root = tree.root;                   e7d3
        tree.add(root,1,N,id[x],1);             9252
    }                                           95cf
    cout<<ans<<endl;                            d592
    return 0;                                   7021
}                                               95cf
```

4.8 Rollback UFS

```
//加边删边二部图判定。                           427e
#include <bits/stdc++.h>                        302f
using namespace std;                            421c
const int maxn = 1e5 + 20;                      f374
struct UFS{                                     bd89
    int fa[maxn];                               33ef
    int sz[maxn];                               590c
    int len[maxn];                              6873
    stack<pair<int*,int> > stk;                 65fd
    void init(){                                5d53
        for (int i=1;i<maxn;i++){               e4ba
            fa[i] = i;                          974c
            sz[i] = 1;                          fa1a
            len[i] = 0;                         c008
        }                                       95cf
    }                                           95cf
    UFS(){                                      e034
        init();                                 07e2
    }                                           95cf
    pair<int,int> find(int x){                  fee7
        if (fa[x] == x)return make_pair(x,0);   7eb8
        else{                                   037f
            pair<int,int> ret = find(fa[x]);    2890
            ret.second ^= len[x];               22aa
            return ret;                         ee0f
        }                                       95cf
    }                                           95cf
    // 0 fail                                   427e
    // 1 succ but not update                    427e
    // 2 succ and update                        427e
```

```
41b9     int merge(int x,int y){
7121         int fx,lenx;
5d92         int fy,leny;
9726         tie(fx,lenx) = find(x);
d13a         tie(fy,leny) = find(y);
e94b         if (fx == fy){
4350             return lenx ^ leny;
95cf         }
93ac         if (sz[fx] > sz[fy]){
65b4             swap(lenx,leny);
47d4             swap(x,y);
6c4f             swap(fx,fy);
95cf         }
dfaa         stk.push(make_pair(&sz[fy],sz[fy]));
863a         stk.push(make_pair(&fa[fx],fa[fx]));
a93a         fa[fx] = fy;
24e9         sz[fy] += sz[fx];
3c8a         if (lenx == leny){
5f4d             len[fx] = 1;
8e2e         }else{
7cc4             len[fx] = 0;
95cf         }
ca92         return 2;
95cf     }
831d     void rollback(){
5a7d         for (int i=0;i<2;i++){
503e             int * tar;
d26b             int val;
5b9a             tie(tar,val) = stk.top();
75b6             stk.pop();
9133             (*tar) = val;
95cf         }
427e
95cf     }
5795 }ufs;
58c6 const char* YES = "YES";
a0f7 const char* NO = "NO";
cd1e bool ans[maxn];
23cc struct SegmentTree{
90fc     vector<pair<int,int> > edges[maxn*4];
2161     void put(int x,int l,int r,int L,int R,pair<int,int> e){
d499         if (l > R || L > r)return;
4d29         if (L <= l && r <= R){
5bfc             edges[x].push_back(e);
4f2d             return;
95cf         }
b8b7         int mid = l + r >> 1;
8d76         put(x<<1,l,mid,L,R,e);
36cd         put(x<<1|1,mid+1,r,L,R,e);
95cf     }
8b28     void dfs(int x,int l,int r){
cd24         int succ = true;
8abb         int cnt = 0;
92f7         for (auto e : edges[x]){
0f8b             int x,y;
2bba             tie(x,y) = e;
6848             int ret = ufs.merge(x, y);
ecd5             succ &= ret!= 0;
7c6f             if (!succ){
9102                 for (int i=0;i<cnt;i++)
5e31                     ufs.rollback();
4f2d                 return;
95cf             }
feaf             cnt += ret == 2;
95cf         }
3a0d         if (l == r){
91cd             ans[l] = succ;
9102             for (int i=0;i<cnt;i++)
5e31                 ufs.rollback();
4f2d             return;
95cf         }
b8b7         int mid = l + r >> 1;
7405         dfs(x<<1,l,mid);
b115         dfs(x<<1|1,mid+1,r);
9102         for (int i=0;i<cnt;i++)
5e31             ufs.rollback();
95cf     }
1d91     void debug(int x,int l,int r){
4bde         cerr<<x<<"␣:␣"<<"[␣"<<l<<"␣,␣"<<r<<"␣]"<<endl;
92f7         for (auto e : edges[x]){
54f1             int u,v;
4c70             tie(u,v) = e;
40e5             cerr<<"<␣"<<u<<"␣,␣"<<v<<"␣>"<<endl;
95cf         }
0eec         if (l == r)return;
b8b7         int mid = l + r >> 1;
7dab         debug(x<<1,l,mid);
f599         debug(x<<1|1,mid+1,r);
95cf        }
f7fb    }segtree;
ae0e    map<pair<int,int>,vector<int> > mp;
3117    int main(){
1ed7        int n,q;
9c97        cin>>n>>q;
949d        for (int i=1;i<=q;i++){
54f1            int u,v;
a02c            cin>>u>>v;
fd0e            if (u > v)swap(u,v);
7c88            mp[make_pair(u,v)].push_back(i);
95cf        }
957e        for (auto pr : mp){
9660            vector<int> & ts = pr.second;
1e87            if (ts.size() & 1){
a1b6                ts.push_back(q+1);
95cf            }
a8d5            for (int i=0;i<ts.size();i+=2){
7ff9                int st = ts[i];
ab30                int ed = ts[i+1] - 1;
8188                segtree.put(1, 1, q, st, ed, pr.first);
95cf            }
95cf        }
427e        //  segtree.debug(1,1,q);
c9f8        segtree.dfs(1, 1, q);
949d        for (int i=1;i<=q;i++){
9d1d            puts(ans[i]?YES:NO);
95cf        }
7021        return 0;
95cf    }
```

4.9 Persistent LiChao SegmentTree

```
427e    // Created by calabash_boy on 2019/10/14.
302f    #include<bits/stdc++.h>
421c    using namespace std;
3045    #define int ll
4085    typedef long long ll;
94c8    const int inf = 1e9 + 5;
2726    const int maxn = 80000 + 50;
9958    const int maxq = 160000 + 50;
80b8    struct Node{
```

```
    int lson,rson,k,b;                                                              4ae7
    //y = k * x + b                                                                 427e
    void init(){                                                                    5d53
        lson = rson = 0;                                                            d096
        k = b = inf;                                                                9016
    }                                                                               95cf
}nodes[maxn * 300];                                                                 abd1
int node_cnt = 0;                                                                   07b6
int root[maxn];                                                                     760d
int update(int x,int l,int r,int L,int R,int k,int b){                              55d8
    int now = node_cnt ++;                                                          d06a
    nodes[now] = nodes[x];                                                          2b27
    int mid = l + r >> 1;                                                           b8b7
    if (l == L && r == R){                                                          0b7f
        if (k * mid + b < nodes[now].k * mid + nodes[now].b){                       5513
            swap(k,nodes[now].k);                                                   98b9
            swap(b,nodes[now].b);                                                   0cf6
        }                                                                           95cf
        if (l == r)return now;                                                      11a1
        if (k * l + b < nodes[now].k * l + nodes[now].b){                           419c
            nodes[now].lson = update(nodes[x].lson,l,mid,l,mid,k,b);                e878
        }                                                                           95cf
        if (k * r + b < nodes[now].k * r + nodes[now].b){                           3833
            nodes[now].rson = update(nodes[x].rson,mid+1,r,mid+1,r,k,b);            c91b
        }                                                                           95cf
        return now;                                                                 7d47
    }                                                                               95cf
    if (L <= mid)nodes[now].lson = update(nodes[x].lson,l,mid,L,min(mid,R),k,b);    616e
    if (mid + 1 <= R)nodes[now].rson = update(nodes[x].rson,mid+1,r,max(mid+1,L)    1157
      ,R,k,b);
    return now;                                                                     7d47
}                                                                                   95cf
int n,q;                                                                            1ed7
int a[maxn], b[maxn], c[maxn];                                                      b3f9
vector<int> E[maxn];                                                                b6a2
void dfs(int u,int fa){                                                             312b
    root[u] = update(root[fa],0,inf,0,c[u],b[u],a[u]);                              6db3
    for (int v : E[u]){                                                             2c0f
        if (v == fa)continue;                                                       b6ee
        dfs(v,u);                                                                   e2f7
    }                                                                               95cf
}                                                                                   95cf
ll query(int x,int l,int r,int pos){                                                7488
    int res = nodes[x].k * pos + nodes[x].b;                                        9860
```

```
427e        //cerr<<x<<" "<<l<<" "<<r<<":"<<nodes[x].k<<" "<<nodes[x].b<<":"<<pos<<" "<<res<<endl;
4745        if (l == r)return res;
b8b7        int mid = l + r >> 1;
e2e9        if (pos <= mid && nodes[x].lson )res = min(res,query(nodes[x].lson,l,mid,pos));
956b        else if (pos > mid &&nodes[x].rson ) res = min(res,query(nodes[x].rson,mid+1,r,pos));
244d        return res;
95cf    }
3b32    signed main(){
1908        scanf("%lld%lld",&n,&q);
6a82        for (int i=1;i<=n;i++)scanf("%lld",a+i);
960d        for (int i=1;i<=n;i++)scanf("%lld",b+i);
d3f4        for (int i=1;i<=n;i++)scanf("%lld",c+i);
324a        for (int i=1;i<n;i++){
54f1            int u,v;
1ddb            scanf("%lld%lld",&u,&v);
068b            E[u].push_back(v);
7e77            E[v].push_back(u);
95cf        }
0c79        root[0] = 0;node_cnt ++;nodes[0].init();
99d6        dfs(1,0);
2cc8        while (q--){
499a            int v,t;
a087            scanf("%lld%lld",&v,&t);
bc8e            printf("%lld\n",query(root[v],0,inf,t));
95cf        }
7021        return 0;
95cf    }
```

5 Graph

5.1 Tarjan(BCC of Edge)

```
427e    // Created by calabash_boy on 18-10-10.
302f    #include<bits/stdc++.h>
421c    using namespace std;
52c1    const int maxn = 1e5+100;
5b3f    int first[maxn],nxt[maxn*2],from[maxn*2],des[maxn*2],isBrige[maxn*2],tot;
ff12    int dfn[maxn],low[maxn],dfs_clock;
8c69    int cnt_e[maxn],cnt_n[maxn];int bcc_cnt;
e093    bool ok[maxn];vector <int> ans;int m,n;
453e    inline void addEdge(int x,int y){
71cf        tot++;
56e8        des[tot] =y;from[tot] =x;
6d84        nxt[tot] = first[x];first[x] = tot;
95cf    }
0e91    void input(){
9af0        cin>>n>>m;
356f        for (int i=0;i<m;i++){
17be            int u,v;scanf("%d%d",&u,&v);
ad4e            addEdge(u,v);addEdge(v,u);
95cf        }
95cf    }
312b    void dfs(int u,int fa){
d413        dfn[u] = low[u] = ++dfs_clock;
3ddf        for (int t = first[u];t;t=nxt[t]){
071c            int v = des[t];if (v==fa)continue;
3c64            if (!dfn[v]){
e2f7                dfs(v,u);
7078                low[u] = min(low[v],low[u]);
f611                if (dfn[u]<low[v]){
4639                    isBrige[t] = true;
b158                    if (t&1){isBrige[t+1] = true;}
6c47                    else{isBrige[t-1] = true;}
95cf                }
e138            }else if (dfn[v]<dfn[u]){low[u] = min(low[u],dfn[v]);}
95cf        }
95cf    }
e992    void blood_fill(int x){
ec01        dfn[x] = bcc_cnt;
4bb0        for (int t = first[x];t;t=nxt[t]){
9516            if (isBrige[t])continue;
e8e0            int v = des[t];
7127            if (!dfn[v]){blood_fill(v);}
95cf        }
95cf    }
fd4b    void check(){
a599        for (int i=1;i<=n;i++){cnt_n[dfn[i]]++;}
a7c6        for (int i=1;i<=tot;i++){
7701            if (isBrige[i]) continue;
5746            cnt_e[dfn[des[i]]]++;
95cf        }
41ce        for (int i=1;i<=bcc_cnt;i++){
e64d            if (cnt_n[i]*2==cnt_e[i]){ok[i]=1;}
95cf        }
95cf    }
d880    void output(){
8d09        for (int i=1;i<=tot;i+=2){
7701            if (isBrige[i])continue;
c2ef            if (ok[dfn[des[i]]])ans.push_back((i+1)/2);
95cf        }
e139        sort(ans.begin(),ans.end());
c4d5        cout<<ans.size()<<endl;
263e        for (int i=0;i<ans.size();i++){printf("%d␣",ans[i]);}
95cf    }
9627    void solve(){
c2a0        for (int i=1;i<=n;i++){if (!dfn[i])dfs(i,-1);}
cbec        memset(dfn,0,sizeof dfn);
6dbf        for (int i=1;i<=n;i++){
aa35            if (!dfn[i]){
03f5                bcc_cnt++;
3b53                blood_fill(i);
95cf            }
95cf        }
92ea        check();output();
95cf    }
3117    int main(){
2a5c        input();
ccd1        solve();
7021        return 0;
95cf    }
```

5.2 Tarjan(BCC of Point)

```
427e    // Created by calabash_boy on 18-10-10.
302f    #include<bits/stdc++.h>
421c    using namespace std;
52c1    const int maxn = 1e5+100;
58a9    int first[maxn],des[maxn*2],nxt[maxn*2],tot;
09ab    int bcc_cnt,cnt_n[maxn],cnt_e[maxn],bcc_no[maxn];
ff12    int dfn[maxn],low[maxn],dfs_clock;
8882    int st[maxn*2],top;bool ok[maxn];
5013    vector<int> ans;vector<int> temp;
4d9b    int m,n;
453e    inline void addEdge(int x,int y){
4704        tot++;des[tot] = y;
    nxt[tot] = first[x];first[x] = tot;                6d84
}                                                      95cf
void input(){                                          0e91
    cin>>n>>m;                                         9af0
    for (int i=0;i<m;i++){                             356f
        int u,v;scanf("%d%d",&u,&v);                   17be
        addEdge(u,v);addEdge(v,u);                     ad4e
    }                                                  95cf
}                                                      95cf
void dfs(int u,int fa){                                312b
    dfn[u] = low[u] = ++dfs_clock;                     d413
    for (int t = first[u];t;t=nxt[t]){                 3ddf
        int v = des[t];                                e8e0
        if (v==fa)continue;                            b6ee
        if (!dfn[v]){                                  3c64
            st[top++] = t;dfs(v,u);                    5248
            low[u] = min(low[u],low[v]);               a19f
            if (low[v]>=dfn[u]){                       9cb7
                bcc_cnt++;ok[bcc_cnt] = true;          9d83
                temp.clear();                          1a7e
                while (true){                          1026
                    int tt = st[--top];                87f2
                    temp.push_back((tt+1)/2);          0648
                    if (bcc_no[des[tt]]!=bcc_cnt){     cf0f
                        bcc_no[des[tt]] = bcc_cnt;     aff7
                        cnt_n[bcc_cnt]++;              3e93
                    }else{                             8e2e
                        ok[bcc_cnt] = false;           e551
                    }                                  95cf
                    cnt_e[bcc_cnt]++;                  83bb
                    if (tt==t)break;                   5047
                }                                      95cf
                if (ok[bcc_cnt]&&temp.size()>1){       b114
                    for (int i=0;i<temp.size();i++){   af9b
                        ans.push_back(temp[i]);        90d3
                    }                                  95cf
                }                                      95cf
            }                                          95cf
        }else if (dfn[v]<dfn[u]){                      e245
            st[top++] = t;                             be8d
            low[u] = min(low[u],dfn[v]);               769a
        }                                              95cf
    }                                                  95cf
}                                                      95cf
```

```
9627 void solve(){
c2a0     for (int i=1;i<=n;i++){if (!dfn[i])dfs(i,-1);}
e139     sort(ans.begin(),ans.end());
c4d5     cout<<ans.size()<<endl;
263e     for (int i=0;i<ans.size();i++){printf("%d␣",ans[i]);}
95cf }
3117 int main(){
2a5c     input();
ccd1     solve();
7021     return 0;
95cf }
```

5.3 Tarjan(SCC)

```
302f #include<bits/stdc++.h>
421c using namespace std;
52c1 const int maxn = 1e5+100;
04f1 int m,n,h;int t[maxn];
7560 int first[maxn*2],nxt[maxn*2],des[maxn*2],tot;
eaf3 int dfn[maxn],low[maxn],dft;bool d[maxn];
414b int flag[maxn],cnt[maxn],scc;stack<int> stk;
e50d bool in[maxn];
704e inline void add(int x,int y){
4704     tot++;des[tot] =y;
6d84     nxt[tot] = first[x];first[x] =tot;
95cf }
a4ef void tar(int node){
b081     dfn[node] = low[node] = ++dft;
5782     in[node] = 1;stk.push(node);
e83e     for (int t = first[node];t;t=nxt[t]){
e8e0         int v = des[t];
3c64         if (!dfn[v]){
53e9             tar(v);
9ee1             low[node] = min(low[node],low[v]);
8734         }else if (in[v]){
d1ad             low[node] = min(low[node],dfn[v]);
95cf         }
95cf     }
bb4b     if (dfn[node]==low[node]){
38ac         scc++;
1026         while (true){
6947             int temp = stk.top();
            flag[temp]=scc;  80c2
            in[temp] = 0;  5685
            cnt[scc]++;stk.pop();  b820
            if (temp==node)break;  ea28
        }  95cf
    }  95cf
}  95cf
int main(){  3117
    scanf("%d%d%d",&n,&m,&h);  d994
    for (int i=1;i<=n;i++){scanf("%d",t+i);}  b8ca
    for (int i=0;i<m;i++){  356f
        int u1,u2;scanf("%d%d",&u1,&u2);  4d1b
        if (t[u1]==(t[u2]+1)%h)add(u2,u1);  7ec2
        if (t[u2]==(t[u1]+1)%h)add(u1,u2);  e284
    }  95cf
    for (int i=1;i<=n;i++){if (!dfn[i])tar(i);}  6d72
    for (int i=1;i<=n;i++){  6dbf
        for (int t = first[i];t;t=nxt[t]){  f030
            if (flag[i]==flag[des[t]])continue;  f3e2
            else{d[flag[i]]++;}  a099
        }  95cf
    }  95cf
    cnt[0] =n+1;int ans = 0;  61a1
    for (int i=1;i<=scc;i++){  5176
        if (d[i]==0&&cnt[i]<cnt[ans]){ans = i;}  83aa
    }  95cf
    cout<<cnt[ans]<<endl;  31ae
    for (int i=1;i<=n;i++){  6dbf
        if (flag[i]==ans){cout<<i<<"␣";}  e341
    }  95cf
    cout<<endl;  3251
    return 0;  7021
}  95cf
```

5.4 Dijkstra

```
// Created by calabash_boy on 18-11-13.  427e
// remain k bi-edge such that the most points' dis == min_dis  427e
#include <bits/stdc++.h>  302f
using namespace std;  421c
typedef long long ll;  4085
const ll inf_ll = 0x3f3f3f3f3f3f3f3fll;  1c1d
```

```
a7c7 const int inf = 0x3f3f3f3f;
8856 const int maxn = 300005;
aaaa struct EDGE{int first,second,third;};
47a0 int n,m,k;
04e9 namespace Short_Path_Tree{
db9e     vector<pair<int,int> > Edge[maxn];
727f     bool used[maxn];
b200     void add_edge(int x,int y,int w) {Edge[x].push_back({y,w});}
1e0b     void output(const vector<int> &ans){
90f7         printf("%d\n",(int) ans.size());
69cb         for (int v : ans)printf("%d␣",v);
dcec         puts("");exit(0);
95cf     }
2fb6     void solve(int K){
8c27         vector<int> ans(0);queue<int> Q;
2ad2         used[1] = 1;Q.push(1);
11e5         while (!Q.empty()){
440f             if (ans.size()== K)output(ans);
ff8a             int head = Q.front();Q.pop();
79f8             for (auto pr : Edge[head]){
1ddf                 if (used[pr.first])continue;
5046                 used[pr.first] = 1;
fb50                 ans.push_back(pr.second);
b172                 Q.push(pr.first);
440f                 if (ans.size()==K)output(ans);
95cf             }
95cf         }
25fd         output(ans);
95cf     }
329b };
b049 namespace Dijkstra{
26a7     ll dis[maxn];bool used[maxn];
d92b     vector<EDGE > *Edge;int S,N;
80b8     struct Node{
386c         int x;ll dis;
647a         bool operator < (const Node &other)const{
717e             return other.dis < dis;
95cf         }
329b     };
4826     void init(vector<EDGE>*Edgee,int n,int st){
96ad         Edge = Edgee;S =st;N = n;
95cf     }
ec07     void work(){
2560         memset(dis,inf,sizeof dis);
c124         priority_queue<Node> pq;
b911         dis[S] = 0;pq.push({S,0});
57d6         while (!pq.empty()){
d5d6             Node head = pq.top();pq.pop();
7583             if (used[head.x])continue;
e4b5             used[head.x] = 1;
1a52             for (auto pr : Edge[head.x]){
2fbb                 if (dis[pr.first] > dis[head.x] + pr.second){
d59f                     dis[pr.first] = dis[head.x] + pr.second;
d53e                     pq.push({pr.first,dis[pr.first]});
95cf                 }
95cf             }
95cf         }
95cf     }
c844     void extract_spt(){
5cdb         for (int u=1;u<=N;u++){
79f0             for (auto pr : Edge[u]){
091e                 if (dis[pr.first] == dis[u] + pr.second){
e042                     Short_Path_Tree::add_edge(u,pr.first,pr.third);
95cf                 }
95cf             }
95cf         }
95cf     }
329b };
cae8 vector<EDGE> E[maxn];
3117 int main(){
7ffc     scanf("%d%d%d",&n,&m,&k);
e052     for (int i=1;i<=m;i++){
58ac         int x,y,w;scanf("%d%d%d",&x,&y,&w);
53d8         E[x].push_back({y,w,i});
fd97         E[y].push_back({x,w,i});
95cf     }
080d     Dijkstra::init(E,n,1);
f9c1     Dijkstra::work();
1170     Dijkstra::extract_spt();
734c     Short_Path_Tree::solve(k);
7021     return 0;
95cf }
```

5.5 Dijkstra interval graph

```
427e // CF 786B
```

```
302f  #include<bits/stdc++.h>
421c  using namespace std;
52c1  const int maxn = 1e5 + 100;
0c86  const int N = 10 * maxn;
4085  typedef long long ll;
b049  namespace Dijkstra{
3a06      vector<pair<int,int> > E[N];
e7eb      ll dis[N];
5269      bool used[N];
bb4b      inline void add_edge(int u,int v,int w){
88d1          E[u].push_back(make_pair(v,w));
95cf      }
9fbb      void dijkstra(int S, int N){
69f6          priority_queue<pair<ll,int> > pq;
cd0f          for (int i=1;i<=N;i++){
4d17              dis[i] = 0x3f3f3f3f3f3f3f3fll;
fc61              used[i] = 0;
95cf          }
4fb7          dis[S] = 0;
cd0f          for (int i=1;i<=N;i++){
0f64              pq.push(make_pair(-dis[i],i));
95cf          }
57d6          while (!pq.empty()){
63ef              pair<ll,int> head = pq.top();pq.pop();
c89e              int u; ll dist;
4067              tie(dist,u) = head;
c884              dist *= -1;
9a95              if (used[u])continue;
db27              used[u] = 1;
48e2              for (auto e : E[u]){
33b3                  int v,len;
ccc4                  tie(v,len) = e;
f6e6                  if (dis[v] > dist + len){
078a                      dis[v] = dist + len;
d06d                      pq.push(make_pair(-dis[v],v));
95cf                  }
95cf              }
95cf          }
95cf      }
756f      void output(int n){
6dbf          for (int i=1;i<=n;i++){
b158              printf("%lld␣",dis[i] == 0x3f3f3f3f3f3f3f3fll ? -1:dis[i]);
95cf          }
885d          puts("");
```

```
    }                                                                   95cf
}                                                                       95cf
int n,q,s;                                                              24fc
int cnt;                                                                9f58
struct SegmentTree{                                                     23cc
    int id[maxn*4];                                                     c7e5
    void build(int x,int l,int r,bool up){                              9476
        id[x] = ++cnt;                                                  6281
        if (l == r){                                                    3a0d
            int u = id[x];                                              c35b
            int v = l;                                                  d74c
            if (up)swap(u,v);                                           2d00
            Dijkstra::add_edge(u, v, 0);                                a9ea
            return;                                                     4f2d
        }                                                               95cf
        int mid = l + r >> 1;                                           b8b7
        build(x<<1,l,mid,up);                                           8094
        build(x<<1|1,mid+1,r,up);                                       7d97
        int u = id[x];                                                  c35b
        int v = id[x<<1];                                               dc32
        if (up)swap(u,v);                                               2d00
        Dijkstra::add_edge(u, v, 0);                                    a9ea
        u = id[x];                                                      a419
        v = id[x<<1|1];                                                 e9c6
        if (up)swap(u,v);                                               2d00
        Dijkstra::add_edge(u, v, 0);                                    a9ea
    }                                                                   95cf
    void add_edge(int x,int l,int r,int L,int R, int T, int w, bool up){ 3e8e
        if (l > R || L > r)return;                                      d499
        if (L <= l && r <= R){                                          4d29
            int u = id[x];                                              c35b
            int v = T;                                                  8863
            if (up)swap(u,v);                                           2d00
            Dijkstra::add_edge(u, v, w);                                4c45
            return;                                                     4f2d
        }                                                               95cf
        int mid = l + r >> 1;                                           b8b7
        add_edge(x<<1, l, mid, L, R, T, w, up);                         9083
        add_edge(x<<1|1, mid+1, r, L, R, T, w, up);                     edd2
    }                                                                   95cf
}Down,Up;                                                               dfc9
int main(){                                                             3117
    scanf("%d%d%d",&n,&q,&s);                                           13bb
    cnt = n;                                                            811f
```

```
d237        Down.build(1, 1, n, false);
c1bc        Up.build(1, 1, n, true);
2cc8        while (q--){
aa14            int t,u,l,r,w;
8661            scanf("%d",&t);
8204            if (t == 1){
3b67                int v;
95a1                scanf("%d%d%d",&u,&v, &w);
8637                l = r = v;
96c0                t = 2;
8e2e            }else{
168f                scanf("%d%d%d%d",&u,&l,&r,&w);
95cf            }
163d            if (t == 2){
427e                // u -> [l,r], len = w
63b8                Down.add_edge(1, 1, n, l, r, u, w, true);
8e2e            }else{
427e                // [l,r] -> v, len = w
c4a7                Up.add_edge(1, 1, n, l, r, u, w, false);
95cf            }
95cf        }
3fd3        Dijkstra::dijkstra(s, cnt);
d041        Dijkstra::output(n);
7021        return 0;
95cf    }
```

5.6 Eulor Tour

```
302f    #include<bits/stdc++.h>
421c    using namespace std;
52c1    const int maxn = 1e5 + 100;
a71b    const int maxm = 5e5 + 100;
35b8    int n,m;
03f0    int d[maxn];
427e    //<点，到这个点走的边id>
c49a    vector<pair<int,int> > tour;
37e9    vector<pair<int,int> > E[maxn];
052c    pair<int,int> edge[maxm];
f231    bool used[maxm];
880a    int now[maxn];
5331    void dfs(int u,int e_id){
18c2        for (; now[u] < E[u].size(); now[u] ++){
6003            int v,id;
c7a3            tie(v,id) = E[u][now[u]];
1e6c            if (used[id]) continue;
6be5            used[id] = 1;
038b            dfs(v,id);
95cf        }
4556        tour.push_back(make_pair(u,e_id));
95cf    }
3117    int main(){
ac98        scanf("%d%d",&n,&m);
e052        for (int i=1;i<=m;i++){
e635            int a,b;
a6b8            scanf("%d%d",&a,&b);
4a7b            edge[i] = make_pair(a,b);
7462            E[a].push_back(make_pair(b,i));
2a96            E[b].push_back(make_pair(a,i));
95cf        }
4e9d        dfs(1,-1);
8d42        reverse(tour.begin(), tour.end());
87e7        /*
3977         for (auto pr : tour){
6b5b         int u,id;
2e37         tie(u,id) = pr;
6b68         cerr<<u<<" "<<id<<endl;
95cf         }
f2b5         */
7021        return 0;
95cf    }
```

6 Graph/Tree

6.1 Divide & Conquer of Point

```
427e    //
427e    // Created by calabash_boy on 18-10-6.
427e    //
427e    //求树上长度小于等于k的有向路径数
1915    #include<stdio.h>
54ff    #include<algorithm>
ef2f    #include<cstring>
421c    using namespace std;
bbaa    const int MAX = 1e4+100;
```

```
08a4 const int INF = 0x3f3f3f3f;
0b89 int first [MAX*2];  int des[MAX*2];
3efe int len[MAX*2];  int nxt[MAX*2];
956f int n,k,tot;  int a[MAX];  int sum[MAX];
ecb3 int dp[MAX];  int dis[MAX];  int num,ans;
aa8d bool vis[MAX];  int Sum,Min,Minid;
5d53 void init(){
57d5     memset(first,0,sizeof first);
7ae1     tot =0;  ans =0;
87fb     memset(vis,0,sizeof vis);
95cf }
ce82 inline void add(int x,int y,int z){
71cf     tot++;
3615     des[tot]= y;  len[tot] =z;
6d84     nxt[tot] = first[x];  first[x] = tot;
95cf }
0e91 void input(){
324a     for (int i=1;i<n;i++){
3676         int u,v,w;
95a1         scanf("%d%d%d",&u,&v,&w);
43a8         add(u,v,w);  add(v,u,w);
95cf     }
95cf }
da46 void dfs1(int node,int father){
90d3     sum[node] = 1;  dp[node] = 0;
e83e     for (int t = first[node];t;t = nxt[t]){
e8e0         int v = des[t];
c80a         if (v == father||vis[v]){
b333             continue;
95cf         }
d58d         dfs1(v,node);
cb59         sum[node] += sum[v];
2cf9         dp[node] = max(dp[node],sum[v]);
95cf     }
95cf }
2d8d void dfs2(int node,int father){
4ab1     int temp = max(dp[node],Sum-sum[node]);
d6e3     if (temp<Min){
76f6         Min = temp;  Minid = node;
95cf     }
e83e     for (int t = first[node];t;t = nxt[t]){
e8e0         int v = des[t];
a37f         if (v==father||vis[v]){  continue;  }
253c         dfs2(v,node);
```

```
        }                                                   95cf
}                                                           95cf
int getRoot(int u){                                         6fae
    dfs1(u,0);  Sum = sum[u];                               8e67
    Min = INF;  Minid = -1;                                 3069
    dfs2(u,0);                                              005f
    return Minid;                                           1090
}                                                           95cf
void getDist(int node,int father,int dist){                 4ac1
    dis[num++] = dist;                                      e097
    for (int t = first[node];t;t = nxt[t]){                 e83e
        int v =des[t];                                      e8e0
        if (v == father||vis[v]){  continue;  }             a37f
        getDist(v,node,dist+len[t]);                        6cae
    }                                                       95cf
}                                                           95cf
int calc (int u,int val){                                   97e3
    num=0;  int res =0;                                     9daa
    getDist(u,0,0);                                         d05a
    sort(dis,dis+num);                                      4b02
    int i=0;int j=num-1;                                    e78d
    while (i<j){                                            6f80
        if (dis[i]+dis[j]+2*val<=k){                        e6c0
            res+=j-i;                                       efef
            i++;                                            a42b
        }else{  j--;  }                                     5cd2
    }                                                       95cf
    return res;                                             244d
}                                                           95cf
void solve(int u){                                          ee28
    int root = getRoot(u);                                  b583
    ans +=calc(root,0);  vis[root] = true;                  b2e3
    for (int t = first[root];t;t = nxt[t]){                 235c
        int v = des[t];                                     e8e0
        if (vis[v]){                                        332f
            continue;                                       b333
        }                                                   95cf
        ans-=calc(v,len[t]);                                91fa
        solve(v);                                           a707
    }                                                       95cf
}                                                           95cf
int main(){                                                 3117
    while (scanf("%d%d",&n,&k)!=EOF&&n&&k){                 7666
        init();                                             07e2
```

```
2a5c         input();
1d60         solve(1);
53b1         printf("%d\n",ans);
95cf     }
7021     return 0;
95cf }
```

6.2 Divide & Conquer of Edge

```
427e // Created by calabash_boy on 2019/10/15.
427e // Luogu 5115.SAM + 边分 + 虚树DP: Given S, calculate
427e // \sum_{i<j and LCP(i,j)<=K1 and LCS(i,j)<=K2}{LCS(i,j) * LCP(i,j)}
427e // 最大度数有限制（例如parent树27度），则不需要三度化。
302f #include<bits/stdc++.h>
421c using namespace std;
eb45 const int maxn = 2e5 + 100;
b1f7 char s[maxn],t[maxn];int n,K1,K2;
3e3e struct Suffix_Automaton{
0037     int nxt[maxn*2][26],fa[maxn*2],l[maxn*2];
0db0     int last,cnt;
c75a     Suffix_Automaton(){ clear(); }
1126     void clear(){
8bdb         last =cnt=1;fa[1]=l[1]=0;
9b85         memset(nxt[1],0,sizeof nxt[1]);
95cf     }
e798     void init(char *s){
0bef         while (*s){add(*s-'a');s++;}
95cf     }
681b     void add(int c){
ee06         int p = last, np = ++cnt;
8b9f         memset(nxt[cnt],0,sizeof nxt[cnt]);
97c0         l[np] = l[p]+1;last = np;
b7f5         while (p&&!nxt[p][c])nxt[p][c] = np,p = fa[p];
fdc4         if (!p)fa[np]=1;
037f         else{
5740             int q = nxt[p][c];
d84d             if (l[q]==l[p]+1)fa[np] =q;
037f             else{
2401                 int nq = ++ cnt;
bc67                 l[nq] = l[p]+1;
da26                 memcpy(nxt[nq],nxt[q],sizeof (nxt[q]));
66a6                 fa[nq] =fa[q];fa[np] = fa[q] =nq;
5dc1                 while (nxt[p][c]==q)nxt[p][c] =nq,p = fa[p];
95cf             }
95cf         }
95cf     }
0d52     void extract(vector<int> * E,char *s,int n,int *id,int *dep,int K){
3c9b         int temp = 1;
1294         for (int i=0;i<n;i++){
ac16             temp = nxt[temp][s[i] - 'a'];
7eb8             id[temp] = i + 1;
95cf         }
e3bd         for (int i=2;i<=cnt;i++)E[fa[i]].push_back(i);
7b35         for (int i=1;i<=cnt;i++){
9a67             if (l[i] <= K)dep[i] = l[i];
4cb5             else dep[i] = 0;
95cf         }
95cf     }
0fd0 }sam1,sam2;
9ac5 vector<int> EE1[maxn * 2],E2[maxn*2];
90b0 vector<tuple<int,int,int> > E1[maxn*4];
9ea3 int idd1[maxn * 2],id1[maxn*4];
885f int depp1[maxn * 2],dep1[maxn*4];
0fbb int id2[maxn* 2],dep2[maxn*2];
223d bool can_use[maxn*4];
8adf int edge_cnt = 0;
4e0e int cnt, st[maxn * 2][20], depth[maxn * 2];
a210 int pos2[maxn*2],pos1[maxn*4];
9156 int dfs_clock,l[maxn*2],r[maxn*2];
c2d4 void dfs2(int u,int fa){
bf7e     l[u] = ++dfs_clock;
98f1     st[u][0] = fa;
8ec8     depth[u] = depth[fa] + 1;
c19b     for (int i=1;i<20 && st[u][i-1];i++){
df83         st[u][i] = st[st[u][i-1]][i-1];
95cf     }
8622     for (auto v : E2[u]){
b6ee         if (v == fa)continue;
3e76         dfs2(v,u);
95cf     }
f142     r[u] = dfs_clock;
95cf }
7ed5 int get_lca(int u,int v){
08ee     if (depth[u] < depth[v])swap(u,v);
1534     for (int i=19;i>=0;i--){
f2cc         if (depth[st[u][i]] >= depth[v])u = st[u][i];
95cf     }
c698     if (u == v)return u;
1534     for (int i=19;i>=0;i--){
8ce3         if (st[u][i] != st[v][i]){
aaf6             u = st[u][i];v = st[v][i];
95cf         }
95cf     }
178b     assert(st[u][0] == st[v][0]);
d6f1     return st[u][0];
95cf }
427e //三度化
7d3c int dfs(int u,int fa){
72bd     int now = ++cnt;
63f2     id1[now] = idd1[u];dep1[now] = depp1[u];
8d7f     pos1[id1[now]] = now;
b669     int pre = now;
f4be     for (auto v : EE1[u]){
b6ee         if (v == fa)continue;
71b8         int temp = ++cnt;
72ac         id1[temp] = 0;dep1[temp] = depp1[u];
d721         edge_cnt ++;
d805         E1[pre].push_back(make_tuple(temp,dep1[temp] - dep1[pre],edge_cnt));
4478         E1[temp].push_back(make_tuple(pre,dep1[temp] - dep1[pre],edge_cnt));
1b1e         int vid = dfs(v,u);
d721         edge_cnt ++;
696d         E1[temp].push_back(make_tuple(vid,dep1[vid] - dep1[temp],edge_cnt));
e45d         E1[vid].push_back(make_tuple(temp,dep1[vid] - dep1[temp],edge_cnt));
8dde         pre = temp;
95cf     }
7d47     return now;
95cf }
889f long long ans = 0;
fbd5 int sz[maxn*4];
bc69 int dis[maxn* 4];
8db0 void dfs_dis(int u,int fa,int len){
ac19     dis[u] = len;
6eaa     for (auto e : E1[u]){
7d8f         int v,lll,edge_id;tie(v,lll,edge_id) = e;
20c6         if (v == fa || !can_use[edge_id])continue;
03ed         dfs_dis(v,u,len + lll);
95cf     }
95cf }
3183 void dfs_sz(int u,int fa){
50c0     sz[u] = 1;
```

```
    for (auto e : E1[u]){                                                   6eaa
        int v,len,edge_id;tie(v,len,edge_id) = e;                           5d3a
        if (v == fa || !can_use[edge_id])continue;                          20c6
        dfs_sz(v,u);                                                        4934
        sz[u] += sz[v];                                                     8449
    }                                                                       95cf
}                                                                           95cf
void dfs_edge(int u,int fa,int &e_id,int &uu,int &vv,int &ww,int &max_sz,int    af5f
  tot_node){
    for (auto e : E1[u]){                                                   6eaa
        int v,len,edge_id;tie(v,len,edge_id) = e;                           5d3a
        if (v == fa || !can_use[edge_id])continue;                          20c6
        int max_sz_t = max(sz[v],tot_node - sz[v]);                         5841
        if (max_sz_t < max_sz){                                             6b23
            max_sz = max_sz_t;                                              4d92
            uu = u;vv = v;ww = len;e_id = edge_id;                          8a53
        }                                                                   95cf
        dfs_edge(v,u,e_id,uu,vv,ww,max_sz,tot_node);                        6498
    }                                                                       95cf
}                                                                           95cf
void dfs_node(int u,int fa,vector<int> &nodes){                             11d5
    if (id1[u])nodes.push_back(id1[u]);                                     d7fa
    for (auto e : E1[u]){                                                   6eaa
        int v,len,edge_id;tie(v,len,edge_id) = e;                           5d3a
        if (v == fa || !can_use[edge_id])continue;                          20c6
        dfs_node(v,u,nodes);                                                2f72
    }                                                                       95cf
}                                                                           95cf
int color[maxn * 2];                                                        31b7
int vis[maxn];                                                              d862
long long dp[maxn * 2];                                                     d6e7
long long dp_cnt[maxn*2][2];                                                1207
long long dp_sum[maxn*2][2];                                                2ec5
int stk[maxn*2];                                                            cafd
int fa[maxn*2];                                                             3abd
inline void clear(int x,int type){                                          8345
    dp[x] = 0;vis[x] = type;                                                e238
    for (int c = 0; c < 2;c ++)dp_cnt[x][c] = dp_sum[x][c] = 0;             a8b8
}                                                                           95cf
void DP(vector<int> & nodes_,int ww){                                       74bd
    vector<int> nodes(0);                                                   1112
    for (int x : nodes_){                                                   00c4
        nodes.push_back(pos2[x]);                                           0d7f
    }                                                                       95cf
```

```
4619     for (int x : nodes)clear(x,1);
d5d4     sort(nodes.begin(),nodes.end(),[](int x,int y){
c861         return l[x] < l[y];
b251     });
98e5     int SZ = nodes.size();
8d5c     for (int i=1;i<SZ;i ++){
0378         int temp = get_lca(nodes[i-1],nodes[i]);
ada3         if (!vis[temp]){
f11a             nodes.push_back(temp);
9df9             clear(temp,2);
95cf         }
95cf     }
5557     if (!vis[1]){
1428         nodes.push_back(1);
74de         clear(1,2);
95cf     }
d5d4     sort(nodes.begin(),nodes.end(),[](int x,int y){
c861         return l[x] < l[y];
b251     });
323d     int top = 1;
2894     stk[0] = nodes[0];
031f     for (int i=1;i<nodes.size();i++){
8825         while (l[nodes[i]] > r[stk[top-1]]) top --;
cf1f         fa[nodes[i]] = stk[top-1];
67ba         stk[top++] = nodes[i];
95cf     }
8235     long long anss = 0;
ceb1     for (int i= nodes.size() - 1;i >=0 ;i --){
529a         int u = nodes[i], c = vis[u] == 1? color[id2[u]] - 1: -1;
c997         if (c != -1){
c4fa             long long A = dep1[pos1[id2[u]]] - dis[pos1[id2[u]]];
dbda             dp[u] += A * dp_cnt[u][!c] + dp_sum[u][!c];
18b4             dp[u] -= dp_cnt[u][!c] * ww;
b801             dp_cnt[u][c] ++;
e09a             dp_sum[u][c] += A;
95cf         }
405e         long long temp_ans = dp[u] * dep2[u];
81b7         assert(temp_ans %2 == 0);
516a         anss += temp_ans/2;
d9a7         dp[fa[u]] += dp_cnt[fa[u]][0] * dp_sum[u][1] + dp_cnt[u][0] * dp_sum[fa[u]][1];
041d         dp[fa[u]] += dp_cnt[fa[u]][1] * dp_sum[u][0] + dp_cnt[u][1] * dp_sum[fa[u]][0];
c5a9         dp[fa[u]] -= (dp_cnt[fa[u]][1] * dp_cnt[u][0] + dp_cnt[fa[u]][0] *
             dp_cnt[u][1]) * ww;
         for (int c = 0;c < 2;c ++){                                     4545
             dp_cnt[fa[u]][c] += dp_cnt[u][c];                           96b5
             dp_sum[fa[u]][c] += dp_sum[u][c];                           4009
         }                                                               95cf
     }                                                                   95cf
     ans += anss;                                                        b484
     for (int x : nodes)vis[x] = 0;                                      70c2
}                                                                        95cf
void calc(int uu,int vv,int ww){                                         29be
    vector<int> L(0),R(0),nodes(0);                                      52e9
    dfs_node(uu,0,L);dfs_node(vv,0,R);                                   b258
    for (int x : L){color[x] = 1;nodes.push_back(x);}                    e503
    for (int x : R){color[x] = 2;nodes.push_back(x);}                    4e8a
    DP(nodes,ww);                                                        e5e6
}                                                                        95cf
void dfs(int root){                                                      35ab
    dfs_sz(root,0);                                                      f054
    int tot_node = sz[root];                                             bfcc
    if (tot_node == 1)return;                                            11c5
    int edge_id,uu,vv,ww,max_sz = tot_node + 1;                          18a2
    dfs_edge(root,0,edge_id,uu,vv,ww,max_sz,tot_node);                   232d
    can_use[edge_id] = false;                                            5cf1
    dfs_dis(uu,0,0);dfs_dis(vv,0,0);                                     7e85
    calc(uu,vv,ww);dfs(uu);dfs(vv);                                      ab13
}                                                                        95cf
int main(){                                                              3117
    scanf("%s%d%d",s,&K1,&K2);                                           c11b
    n = strlen(s);                                                       5264
    memcpy(t,s,sizeof s);reverse(t,t + n);                               5f03
    sam1.init(s);sam2.init(t);                                           7dbf
    sam1.extract(EE1,s,n,idd1,depp1,K2);                                 b1ed
    sam2.extract(E2,t,n,id2,dep2,K1);                                    8ada
    for (int i=1;i<= sam2.cnt; i ++){                                    073d
        if (id2[i]){                                                     4ce1
            id2[i] = n + 1 - id2[i];                                     4913
            pos2[id2[i]] = i;                                            97e0
        }                                                                95cf
    }                                                                    95cf
    int root1 = dfs(1,0);int root2 = 1;                                  b333
    dfs2(root2,0);                                                       ac7c
    memset(can_use,true,sizeof can_use);                                 fbab
    dfs(root1);                                                          4267
    cout<<ans<<endl;                                                     d592
```

```
7021     return 0;
95cf }
```

6.3 Heavy Light Decomposition

```
427e // Created by calabash_boy on 18-7-3.
427e //统计路径上标记边的个数
302f #include<bits/stdc++.h>
421c using namespace std;
8e62 const int maxn = 500000+100;
4bc9 int n,q,m,Root;  char s[10];
5f7d struct BIT{
3bf5     int sm[maxn];
cf5a     int lowbit(int _x){return _x&(-_x);}
d5af     void build (int l,int r){
5023         for (int i=l;i<=r;i++)add(i,1);
95cf     }
6142     void add(int x,int val){
dc9a         while (x<=maxn){
9ccc             sm[x]+=val;x+=lowbit(x);
95cf         }
95cf     }
eb61     int sum(int x){
5839         int res =0;
6f1c         while (x){
e64f             res+=sm[x];
e6b6             x-=lowbit(x);
95cf         }
244d         return res;
95cf     }
9fc7     int query_sum(int l,int r){
7789         return sum(r)-sum(l-1);
95cf     }
b0c1 }tree;
9c21 namespace Heavy_Light_Decomposition{
7b14     int first[maxn*2];int nxt[maxn*2];int des[maxn*2];
cd30     int tot,cnt=0;
0d93     int tpos[maxn];int dep[maxn];int top[maxn];
d6bf     int fa[maxn]; int wson[maxn];  int sz[maxn];
f9d3     inline void addEdge(int _u, int _v){
26b9         des[++tot] = _v;
a66a         nxt[tot] = first[_u];
```

```
        first[_u] = tot;                                  593b
    }                                                     95cf
    //统计dep，子树sz，重儿子wson                           427e
    void dfs(int node,int father){                        dd7c
        dep[node] = dep[father]+1;                        c5b1
        fa[node] = father;  sz[node] =1;                  afa3
        for (int t = first[node];t;t = nxt[t]){           e83e
            int v = des[t];                               e8e0
            if (v==father){  continue;  }                 e092
            dfs(v,node);                                  1f8e
            if (sz[v]>sz[wson[node]]){                    acb3
                wson[node] = v;                           44c0
            }                                             95cf
            sz[node]+=sz[v];                              47d5
        }                                                 95cf
    }                                                     95cf
    //node所在链的头是chain                                 427e
    void dfs2(int node,int father,int chain){             aee5
        top[node] = chain;  tpos[node] = ++cnt;           950f
        if (wson[node]){                                  d010
            dfs2(wson[node],node,chain);                  0f73
        }                                                 95cf
        for (int t = first[node];t;t = nxt[t]){           e83e
            int v = des[t];                               e8e0
            if (v==father||v ==wson[node]){  continue;  } b928
            dfs2(v,node,v);                               e6aa
        }                                                 95cf
    }                                                     95cf
    /* s 树根 */                                           c352
    void init(int root){                                  1a86
        dfs(root,0);                                      5136
        dfs2(root, 0, root);                              7cdf
    }                                                     95cf
    int lca(int x,int y){                                 620b
        while (top[x]!=top[y]){                           d2f8
            if (dep[top[x]]<dep[top[y]]){swap(x,y);}      0cc5
            x = fa[top[x]];                               7456
        }                                                 95cf
        if (dep[x]<dep[y])swap(x,y);                      d22b
        return y;                                         c218
    }                                                     95cf
    void modify(int u,int v){                             29cf
        if (fa[u]!=v){  swap(u,v);  }                     733e
        tree.add(tpos[u],-1);                             1e27
```

```
95cf     }
1dc2     int get_sum(int u,int v){
5839         int res =0;
03a1         while (top[u]!=top[v]){
a716             if (dep[top[u]]<dep[top[v]]){  swap(u,v);  }
f1e8             res+= tree.query_sum(tpos[top[u]],tpos[u]);
005b             u = fa[top[u]];
95cf         }
4b1a         if (dep[u]<dep[v]){  swap(u,v); }
cbff         res += tree.query_sum(tpos[v],tpos[u]);
244d         return res;
95cf     }
95cf }
3117 int main(){
cd91     scanf("%d",&n);
324a     for (int i=1;i<n;i++){
17be         int u,v;  scanf("%d%d",&u,&v);
1478         Heavy_Light_Decomposition::addEdge(u, v);
e4e6         Heavy_Light_Decomposition::addEdge(v, u);
95cf     }
90e1     Heavy_Light_Decomposition::init(1);
427e     //维护
1ca5     tree.build(2,n);
ea85     scanf("%d",&q);
3605     q+=n-1;
2cc8     while (q--){
587c         scanf("%s",s);
5d10         if (s[0]=='W'){
3c9e             int x;
ea4e             scanf("%d",&x);
3b50             printf("%d\n",Heavy_Light_Decomposition::get_sum(1,x));
8e2e         }else{
0f8b             int x,y;
a9b3             scanf("%d%d",&x,&y);
a309             Heavy_Light_Decomposition::modify(x,y);
95cf         }
95cf     }
7021     return 0;
95cf }
```

6.4 Virtual Tree

```
427e //
427e // Created by calabash_boy on 18-10-6.
427e //
427e
302f #include <bits/stdc++.h>
421c using namespace std;
5cad typedef long long LL;
40fb const int maxn = 25e4+100;
b1ec const LL INF = 0x3f3f3f3f3f3f3f3fLL;
58a9 int first[maxn],des[maxn*2],nxt[maxn*2],tot;
35b8 int n,m;
667a LL dp[maxn],leng[maxn*2], len[maxn];
e55b int vis[maxn],dep[maxn],fa[maxn];
21fe int sz[maxn],wson[maxn],ttop[maxn],tfa[maxn];int k,h[maxn];
0a19 int stk[maxn],top;int l[maxn],r[maxn],dfs_clock;
a50a inline void addEdge(int x,int y,int w){
71cf     tot++;
a752     des[tot] = y;leng[tot] = w;
6d84     nxt[tot] = first[x];first[x] = tot;
95cf }
827d void dfs(int u,int fath){
84cf     l[u] = ++dfs_clock;sz[u]=1;
3ddf     for (int t = first[u];t;t=nxt[t]){
e8e0         int v = des[t];
9d74         if (v==fath)continue;
62a8         LL w = leng[t];
e4a6         dep[v] = dep[u] + 1;tfa[v]=u;
818a         len[v] = min(len[u],w);
7457         dfs(v,u);sz[u]+=sz[v];
c7eb         if (sz[v]>sz[wson[u]]){wson[u] = v;}
95cf     }
f142     r[u]=dfs_clock ;
95cf }
4707 void dfs2(int u,int chain){
0865     ttop[u]=chain;
d6b4     if (wson[u])dfs2(wson[u],chain);
3ddf     for (int t = first[u];t;t=nxt[t]){
e8e0         int v = des[t];
0c51         if (v==tfa[u]||v==wson[u])continue;
8064         dfs2(v,v);
95cf     }
95cf }
620b int lca(int x,int y){
00da     while (ttop[x]!=ttop[y]){
```

```
6d86         if (dep[ttop[x]]<dep[ttop[y]])swap(x,y);
2df6         x = tfa[ttop[x]];
95cf     }
d22b     if (dep[x]<dep[y])swap(x,y);
c218     return y;
95cf }
4ac9 bool cmp(int x,int y){return l[x]<l[y];}
9627 void solve(){
c93a     scanf("%d",&k);
f3ea     for (int i=0;i<k;i++){
3596         scanf("%d",h+i);
a234         vis[h[i]]=1;dp[h[i]]=0;
95cf     }
f5bb     sort(h,h+k,cmp);
a555     int kk =k;
c701     for (int i=1;i<kk;i++){
4680         int temp = lca(h[i-1],h[i]);
b925         if (!vis[temp])vis[temp]=2,h[k++] =temp,dp[temp]=0;
95cf     }
22a9     if (!vis[1])vis[1]=2,h[k++]=1,dp[1]=0;
f5bb     sort(h,h+k,cmp);
25a6     top=1;stk[0]=h[0];
3ef4     for (int i=1;i<k;i++){
b35a         while (l[h[i]]>r[stk[top-1]])top--;
f930         fa[h[i]] = stk[top-1];
274e         stk[top++] =h[i];
95cf     }
5c52     for (int i=k-1;i>=0;i--){
dca2         if (vis[h[i]]==2)dp[h[i]] = min(dp[h[i]],len[h[i]]);
6a6b         else dp[h[i]] = len[h[i]];
d6ae         dp[fa[h[i]]]+=dp[h[i]];
95cf     }
c682     printf("%lld\n",dp[1]);
f3ea     for (int i=0;i<k;i++){
e3ec         vis[h[i]]=0;
95cf     }
95cf }
3117 int main(){
cd91     scanf("%d",&n);
324a     for (int i=1;i<n;i++){
3676         int u,v,w;
95a1         scanf("%d%d%d",&u,&v,&w);
8796         addEdge(u,v,w);addEdge(v,u,w);
95cf     }
```

```
    len[0] = len[1] = INF;    8694
    dfs(1,-1);dfs2(1,1);    0e9e
    scanf("%d",&m);    aa8d
    while (m--){solve();}    74ed
    return 0;    7021
}    95cf
```

7 Math

7.1 FFT

```
// Created by calabash_boy on 18-6-18.    427e
#include <bits/stdc++.h>    302f
using namespace std;    421c
namespace fft {    e48c
    //attention data type    427e
    typedef long long type;    53f7
    typedef double db;    f7dc
    struct cp {    e718
        db x, y;    ba04
        cp() { x = y = 0; }    cfb3
        cp(db x, db y) : x(x), y(y) {}    f329
    };    329b
    cp operator+(cp a, cp b) { return cp(a.x + b.x, a.y + b.y); }    9f2f
    cp operator-(cp a, cp b) { return cp(a.x - b.x, a.y - b.y); }    624b
    cp operator*(cp a, cp b) { return cp(a.x * b.x - a.y * b.y, a.x * b.y + a.y    36fe
      * b.x); }
    cp conj(cp a) { return cp(a.x, -a.y); }    a0e1
    type base = 1;    6ecb
    vector<cp> roots = {{0, 0}, {1, 0}};    44b9
    vector<type> rev = {0, 1};    3a50
    const db PI = acosl(-1.0);    3f9e
    void ensure_base(type nbase) {    2b5b
        if (nbase <= base) return;    7037
        rev.resize(static_cast<unsigned long>(1 << nbase));    bbb1
        for (type i = 0; i < (1 << nbase); i++) {    89c3
            rev[i] = (rev[i >> 1] >> 1) + ((i & 1) << (nbase - 1));    33a9
        }    95cf
        roots.resize(static_cast<unsigned long>(1 << nbase));    a0ef
        while (base < nbase) {    7acf
            db angle = 2 * PI / (1 << (base + 1));    cd10
            for (type i = 1 << (base - 1); i < (1 << base); i++) {    f864
b824                 roots[i << 1] = roots[i];
90ee                 db angle_i = angle * (2 * i + 1 - (1 << base));
a5d7                 roots[(i << 1) + 1] = cp(cos(angle_i), sin(angle_i));
95cf             }
d27a             base++;
95cf         }
95cf     }
3548     void fft(vector<cp> &a, type n = -1) {
805a         if (n == -1) n = a.size();
2fa3         assert((n & (n - 1)) == 0);
dca5         type zeros = __builtin_ctz(n);
c44f         ensure_base(zeros);
a1b9         type shift = base - zeros;
800c         for (type i = 0; i < n; i++) {
aa3c             if (i < (rev[i] >> shift)) {
669c                 swap(a[i], a[rev[i] >> shift]);
95cf             }
95cf         }
5911         for (type k = 1; k < n; k <<= 1) {
b660             for (type i = 0; i < n; i += 2 * k) {
b247                 for (type j = 0; j < k; j++) {
7dca                     cp z = a[i + j + k] * roots[j + k];
ee2d                     a[i + j + k] = a[i + j] - z;
4da7                     a[i + j] = a[i + j] + z;
95cf                 }
95cf             }
95cf         }
95cf     }
fbc2     vector<cp> fa, fb;
6833     vector<type> multiply(vector<type> &a, vector<type> &b) {
02f0         type need = a.size() + b.size() - 1;
cf09         type nbase = 0;
0c88         while ((1 << nbase) < need) nbase++;
6f7d         ensure_base(nbase);
cb07         type sz = 1 << nbase;
b44d         if (sz > (type) fa.size())
74d8             fa.resize(static_cast<unsigned long>(sz));
46e8         for (type i = 0; i < sz; i++) {
2155             type x = (i < (type) a.size() ? a[i] : 0);
f2d7             type y = (i < (type) b.size() ? b[i] : 0);
140d             fa[i] = cp(x, y);
95cf         }
eb13         fft(fa, sz);
53b1         cp r(0, -0.25 / sz);
6611         for (type i = 0; i <= (sz >> 1); i++) {
3695             type j = (sz - i) & (sz - 1);
f17e             cp z = (fa[j] * fa[j] - conj(fa[i] * fa[i])) * r;
4a23             if (i != j) {
0628                 fa[j] = (fa[i] * fa[i] - conj(fa[j] * fa[j])) * r;
95cf             }
8cd4             fa[i] = z;
95cf         }
eb13         fft(fa, sz);
a834         vector<type> res(static_cast<unsigned long>(need));
4516         for (type i = 0; i < need; i++) {
1653             res[i] = fa[i].x + 0.5;
95cf         }
244d         return res;
95cf     }
3ca7     vector<type> multiply_mod(vector<type> &a, vector<type> &b, type m, type eq
           = 0) {
02f0         type need = a.size() + b.size() - 1;
cf09         type nbase = 0;
0c88         while ((1 << nbase) < need) nbase++;
6f7d         ensure_base(nbase);
cb07         type sz = 1 << nbase;
3292         if (sz > (type) fa.size()) {
74d8             fa.resize(static_cast<unsigned long>(sz));
95cf         }
2f67         for (type i = 0; i < (type) a.size(); i++) {
cfe6             type x = (a[i] % m + m) % m;
7cb0             fa[i] = cp(x & ((1 << 15) - 1), x >> 15);
95cf         }
b1cb         fill(fa.begin() + a.size(), fa.begin() + sz, cp {0, 0});
eb13         fft(fa, sz);
8c71         if (sz > (type) fb.size()) {
14b9             fb.resize(static_cast<unsigned long>(sz));
95cf         }
2cba         if (eq) {
88c2             copy(fa.begin(), fa.begin() + sz, fb.begin());
8e2e         } else {
0ac2             for (type i = 0; i < (type) b.size(); i++) {
ad83                 type x = (b[i] % m + m) % m;
97f9                 fb[i] = cp(x & ((1 << 15) - 1), x >> 15);
95cf             }
5f8e             fill(fb.begin() + b.size(), fb.begin() + sz, cp {0, 0});
e06b             fft(fb, sz);
95cf         }
```

```
d8f2            db ratio = 0.25 / sz;
9cc7            cp r2(0, -1);cp r3(ratio, 0);
0367            cp r4(0, -ratio);cp r5(0, 1);
6611            for (type i = 0; i <= (sz >> 1); i++) {
3695                type j = (sz - i) & (sz - 1);
996e                cp a1 = (fa[i] + conj(fa[j]));
a37e                cp a2 = (fa[i] - conj(fa[j])) * r2;
51fd                cp b1 = (fb[i] + conj(fb[j])) * r3;
ad90                cp b2 = (fb[i] - conj(fb[j])) * r4;
4a23                if (i != j) {
792b                    cp c1 = (fa[j] + conj(fa[i]));
ecde                    cp c2 = (fa[j] - conj(fa[i])) * r2;
18a0                    cp d1 = (fb[j] + conj(fb[i])) * r3;
6ced                    cp d2 = (fb[j] - conj(fb[i])) * r4;
28c4                    fa[i] = c1 * d1 + c2 * d2 * r5;
178d                    fb[i] = c1 * d2 + c2 * d1;
95cf                }
1184                fa[j] = a1 * b1 + a2 * b2 * r5;
87e9                fb[j] = a1 * b2 + a2 * b1;
95cf            }
922b            fft(fa, sz);fft(fb, sz);
a834            vector<type> res(static_cast<unsigned long>(need));
4516            for (type i = 0; i < need; i++) {
9dbc                long long aa = fa[i].x + 0.5;
d335                long long bb = fb[i].x + 0.5;
de5d                long long cc = fa[i].y + 0.5;
67e4                res[i] = (aa + ((bb % m) << 15) + ((cc % m) << 30)) % m;
95cf            }
244d            return res;
95cf        }
2307        vector<type> square_mod(vector<type> &a, type m) {
b845            return multiply_mod(a, a, m, 1);
95cf        }
329b    };
eb45    const int maxn = 2e5+100;
86d1    int n,x;
7608    int a[maxn],sum[maxn],cnt[maxn];
a6aa    vector<long long > A,B,C;
427e    //example:
427e    //f[i] = number of subsequences whose occurence of 1 is i.
427e    //f[i] = \sum_{cnt[j]*cnt[j-i]}
3117    int main(){
a5fe        scanf("%d%d",&n,&x);cnt[0]=1;
6dbf        for (int i=1;i<=n;i++){
```

```
        scanf("%d",a+i);                                60cb
        sum[i] =sum[i-1];                               9a8f
        if(a[i]<x)sum[i]++;                             1229
        cnt[sum[i]]++;                                  6210
    }                                                   95cf
    A.resize(n*2+2);B.resize(n*2+2);                    bb11
    for (int i=0;i<=n;i++){                             0423
        A[n+i] = cnt[i];B[n-i] = cnt[i];                1451
    }                                                   95cf
    C = fft::multiply(A,B);                             284a
    C[n*2]-=n+1;C[n*2]>>=1;                             7cf7
    for (int i=n*2;i<=n*3;i++){ cout<<C[i]<<"␣"; }      d7c0
    return 0;                                           7021
}                                                       95cf
```

7.2 FWT

```
// Created by calabash_boy on 18-8-17.                           427e
//UOJ 310                                                        427e
#include<bits/stdc++.h>                                          302f
using namespace std;                                             421c
typedef long long LL;                                            5cad
const int N = 1048576;;                                          a923
const int MOD = 998244353;                                       5bf2
const int INV2 = (MOD+1)>>1;                                     2003
const int INV4 = 1LL*INV2*INV2%MOD;                              4d4d
int a[N];                                                        ac9d
int n;                                                           5c83
//xor fwt : A[i] = \sigma{-1^([i&j])*a[j]}   [x]:count of 1-bit  427e
void FWT(int *a,int n,int r){                                    3284
    for (int i=1;i<n;i<<=1){                                     65de
        for (int j=0;j<n;j+=(i<<1)){                             2d6f
            for (int k =0;k<i;k++){                              3d77
                int x = a[j+k];int y = a[j+k+i];                 269d
                if (r){                                          f418
                    a[j+k] = (x+y)%MOD;                          a62b
                    a[j+k+i] = (x-y+MOD)%MOD;                    df0f
                }else{                                           8e2e
                    a[j+k] = 1LL*(x+y)*INV2%MOD;                 a36d
                    a[j+k+i] = 1LL*(x-y+MOD)*INV2%MOD;           5b23
                }                                                95cf
            }                                                    95cf
        }
    }
}
LL pow_mod(LL x,LL y){
    LL ret = 1;
    for (;y;y>>=1){if (y&1) ret = ret*x%MOD;x = x*x%MOD;}
    return ret;
}
int main(){
    scanf("%d",&n);
    for (int i=1;i<=n;i++){
        int x;scanf("%d",&x);
        a[x]++;
    }
    FWT(a,N,1);
    for(int i=0;i<N;i++){
        a[i] = (n+2*a[i])%MOD;
        int cnt3 = 1LL*(a[i]+n)%MOD*INV4%MOD;
        int cnt1 = n-cnt3;
        a[i] = pow_mod(3,cnt3);
        if (cnt1&1)a[i] = MOD-a[i];
    }
    FWT(a,N,0);
    printf("%d\n",(a[0]+MOD-1)%MOD);
    return 0;
}
```

7.3 BerlekampMassey

```
// Created by calabash_boy on 18-8-16.
#include<bits/stdc++.h>
#define FOR(i,l,r) for (int i = (l);i<(r);i++)
#define FORD(i,r,l) for (int i= (r);i>(l);i--)
using namespace std;
typedef long long LL;
typedef vector<LL> V;
const int MOD = 1e9+7;
// k 为 m 最高次数 且  a[m] == 1
namespace BerlekampMassey {
    inline void up(LL& a, LL b) { (a += b) %= MOD; }

    V mul(const V& a, const V& b, const V& m, int k) {
        V r; r.resize(2 * k - 1);
        FOR (i, 0, k)
            FOR (j, 0, k)
                up(r[i + j], a[i] * b[j]);
        FORD (i, k - 2, - 1) {
            FOR (j, 0, k)
                up(r[i + j], r[i + k] * m[j]);
            r.pop_back();
        }
        return r;
    }
    LL pow_mod (LL x,LL y){
        LL ret =1;
        for (;y;y>>=1){if (y&1) ret = ret*x%MOD;x = x * x %MOD;}
        return ret;
    }
    LL get_inv(LL x,LL MOD){
        return pow_mod(x,MOD-2);
    }
    V pow(LL n, const V& m) {
        int k = (int)m.size() - 1; assert(m[k] == -1 || m[k] == MOD - 1);
        V r(k), x(k); r[0] = x[1] = 1;
        for (; n; n >>= 1, x = mul(x, x, m, k))
            if (n & 1) r = mul(x, r, m, k);
        return r;
    }
    LL go(const V& a, const V& x, LL n) {
        // a: (-1, a1, a2, ..., ak).reverse
        // x: x1, x2, ..., xk
        // x[n] = sum[a[i]*x[n-i],{i,1,k}]
        int k = (int)a.size() - 1;
        if (n <= k) return x[n - 1];
        V r = pow(n - 1, a);
        LL ans = 0;
        FOR (i, 0, k)
            up(ans, r[i] * x[i]);
        return ans;
    }

    V BM(const V& x) {
        V a = {-1}, b = {233};
        FOR (i, 1, x.size()) {
            b.push_back(0);
            LL d = 0, la = a.size(), lb = b.size();
d228              FOR (j, 0, la) up(d, a[j] * x[i - la  + 1 + j]);
85ae              if (d == 0) continue;
292f              V t; for (auto& v: b) t.push_back(d * v % MOD);
296a              FOR (j, 0, a.size()) up(t[lb - 1 - j], a[la - 1 - j]);
3ead              if (lb > la) {
46e5                  b = a;
f0ce                  LL inv = -get_inv(d, MOD);
b92f                  for (auto& v: b) v = v * inv % MOD;
95cf              }
64bf              a.swap(t);
95cf          }
b24a          for (auto& v: a) up(v, MOD);
5ffd          return a;
95cf      }
bb1a      void sample();
95cf  }
f425  void BerlekampMassey::sample(){
3ddb      V x(6);
26b0      x[0] = 1;x[1] = 2;
dc7c      x[2] = 21;x[3] = 212;
408c      x[4] = 2141;x[5] = 21622;
6243      V a = BerlekampMassey::BM(x);
a849      cout<<"a[n]␣=␣";
0126      for (int i = 0;i<a.size()-2;i++){
844c          cout<<a[i]<<"*a[n-"<<a.size()-1-i<<"]␣+␣";
95cf      }
e0ba      cout<<a[a.size()-2]<<"*a[n-1]"<<endl;
95cf  }
3117  int main(){
47ff      BerlekampMassey::sample();
7021      return 0;
95cf  }
```

7.4 CRT

```
427e  //
427e  // Created by DELL on 2019/2/12.
427e  //luogu 4777
302f  #include<bits/stdc++.h>
421c  using namespace std;
4085  typedef long long ll;
52c1  const int maxn = 1e5+100;
ff57  namespace CRT{
8345      ll ex_gcd(ll a,ll b,ll& x,ll& y){
7d1a          if (b == 0){x = 1;y = 0;return a;}
df10          ll gcd = ex_gcd(b,a%b,x,y);
8737          ll t = x;x = y;y = t - a/b*y;
8be6          return gcd;
95cf      }
40a5      ll mul_mod(ll a,ll b,ll m){
292f          ll res = 0;
ca22          while (b){
90a9              if (b&1){
6d81                  res = (res + a) % m;
95cf              }
ca1f              b >>=1;
06e5              a = a * 2 % m;
95cf          }
244d          return res;
95cf      }
427e      // ans = first + t * second;
427e      // x = second (mod first)
7f60      pair<ll,ll>work(vector<pair<ll,ll> >&es ){
601c          ll ans = es[0].second;
2a60          ll M = es[0].first;
954a          for (int i=1;i<es.size();i++){
c35f              ll a = es[i].first;
27e2              ll b = es[i].second;
d406              ll x,y;
6786              ll gcd = ex_gcd(M,a,x,y);
69fb              ll c = (b - ans %a + a) % a;
1a20              a/=gcd;
e23e              if (c % gcd)return {-1,-1};
5a47              x = (mul_mod(x ,(c / gcd),a) + a)% a;
4108              ans += M * x;
9b2a              M *= a;
324d              ans %= M;
95cf          }
f267          return {ans,M};
95cf      }
95cf  }
6a81  vector<pair<ll,ll> > es;
3117  int main(){
5c83      int n;
cd91      scanf("%d",&n);
1294      for (int i=0;i<n;i++){
```

```
6d1c        ll a,b;
9407        scanf("%lld%lld",&a,&b);
3a4a        es.push_back(make_pair(a,b));
95cf    }
c88b    pair<ll,ll> ans = CRT::work(es);
427e    // cout<<ans.first<<" "<<ans.second<<endl;
ee13    ll x = ans.first;
290b    cout<<x<<endl;
7021    return 0;
95cf }
```

7.5 Linear Sieve

```
302f #include<bits/stdc++.h>
421c using namespace std;
68e4 const int maxn = 1e7+10;
4085 typedef long long ll;
727f bool used[maxn];
efe5 int mu[maxn];
7c8f vector<int> prime;
c882 ll f[maxn];
a0b1 int low[maxn];
22c5 void sieve(int size){
427e     //f:multiplicative function;
7d97     assert(size < maxn);
7f5a     mu[1] = 1;
c6b9     f[1] = 1;
40bd     for (int i=2;i<=size;i++){
efb1         if (!used[i]){
1024             prime.push_back(i);
7171             mu[i] = -1;
427e             //f:TODO
c21b             low[i] = i;
95cf         }
eb1a         for (int j = 0;j < prime.size();j++){
d3c2             ll nxt = 1ll * i * prime[j];
b561             if (nxt > size)break;
6b89             used[nxt] = 1;
073a             if (i % prime[j]){
b9b8                 low[nxt] = prime[j];
66f9                 mu[nxt] = -mu[i];
427e                 //f: mod or not?
                f[nxt] = f[i] * f[prime[j]];                    7225
            }else{                                              8e2e
                low[nxt] = prime[j] * low[i];                   734b
                mu[nxt] = 0;                                    8ec3
                if (low[nxt] != nxt){                           b401
                    //mod or not?                               427e
                    f[nxt] = 1ll * f[low[nxt]] * f[nxt/low[nxt]];  4d18
                }else{                                          8e2e
                    // i = prime[j] ^ k                         427e
                    //f:TODO                                    427e
                }                                               95cf
                break;                                          6173
            }                                                   95cf
        }                                                       95cf
    }                                                           95cf
}                                                               95cf
int main(){                                                     3117
    sieve(1e7);                                                 ff91
    return 0;                                                   7021
}                                                               95cf
```

7.6 Linear Basis

```
/* Generated by powerful Codeforces Tool                        6c13
 * Author: calabash_boy_love_15                                 c7a5
 * Time: 2019-05-15 11:00:02                                    6619
 * Personal Code Template: https://github.com/4thcalabash/ACM-Code-Library  ca63
**/                                                             421d
#include <bits/stdc++.h>                                        302f
using namespace std;                                            421c
int s[maxn];                                                    4c95
int n;                                                          5c83
struct Linear_Basis{                                            2360
    //basis vector                                              427e
    int basis[22];                                              d2e8
    //basis vector in origin data                               427e
    int num[22];                                                36c3
    void clear(){                                               1126
        memset(basis,0,sizeof basis);                           037d
        memset(num,0,sizeof num);                               7b40
    }                                                           95cf
    void ins(int x){                                            2f9f
c7a6            int bk = x;
54c0            for (int i=20;i>=0;i--){
a0f3                if (x & (1<< i)){
e222                    if (!basis[i]){basis[i] = x;num[i] = bk;break;}
370c                    x ^= basis[i];
95cf                }
95cf            }
95cf        }
5bcc        int count(){
8abb            int cnt = 0;
9f1c            for (int i=0;i<=20;i++){
340e                cnt += (basis[i] != 0);
95cf            }
6808            return cnt;
95cf        }
56dd        void debug(){
af23            _debug("basis␣:␣");
9f1c            for (int i=0;i<=20;i++){
dbf5                if (basis[i])_debug("%d␣:␣%d",i,basis[i]);
95cf            }
95cf        }
4a42    }basis;
3117    int main(){
e1b6        cin>>n;
6dbf        for (int i=1;i<=n;i++){
f9af            cin>>s[i];
9f1c            basis.ins(s[i]);
95cf        }
7021        return 0;
95cf    }
```

7.7 Mobius

```
e9ac    /* x in [1,N]; y in [1,M] (x,y) = 1 */
59b9    #include<cstdio>
09f7    #include<vector>
421c    using namespace std;
52c1    const int maxn = 1e5+100;
4085    typedef long long ll;
727f    bool used[maxn];
7c8f    vector<int> prime;
a00a    ll mu[maxn];
void sieve(){                                                    9bc6
    mu[1] = 1;                                                   7f5a
    for (int i=2;i<maxn;i++){                                    82c4
        if(!used[i]){                                            efb1
            prime.push_back(i);                                  1024
            mu[i] = -1;                                          7171
        }                                                        95cf
        for (int j = 0;j<prime.size();j++){                      eb1a
            long long nxt = 1ll* prime[j] * i;                   b70b
            if(nxt >= maxn)break;                                1487
            used[nxt] = 1;                                       6b89
            if (i % prime[j] == 0){                              20cc
                mu[nxt] = 0;                                     8ec3
                break;                                           6173
            }else{                                               8e2e
                mu[nxt] = -mu[i];                                66f9
            }                                                    95cf
        }                                                        95cf
    }                                                            95cf
}                                                                95cf
ll work(int n,int m){                                            8399
    ll ans = 0;                                                  19f3
    int top = min(n,m);                                          78fb
    for (int i=1;i<=top;i++){                                    3d1c
        ans += 1ll * mu[i] * (n/i) * (m/i);                      7d55
    }                                                            95cf
    return ans;                                                  4206
}                                                                95cf
int main(){                                                      3117
    sieve();                                                     5ec4
    int T;                                                       9523
    scanf("%d",&T);                                              1fd9
    for (int Case = 1;Case <= T;Case ++){                        9415
        int a,b,n,m,k;                                           fb8b
        scanf("%d%d%d%d%d",&a,&n,&b,&m,&k);                      cc1c
        if(k == 0){                                              5399
            printf("Case␣%d:␣0\n",Case);                         8acc
            continue;                                            b333
        }                                                        95cf
        n/=k;                                                    0dac
        m/=k;                                                    a94f
        printf("Case␣%d:␣%lld\n",Case,work(n,m) - work(min(n,m),min(n,m))/2);   0d4c
    }                                                            95cf
    return 0;                                                    7021
95cf }
```

8 Others

8.1 Header

```
427e // Created by calabash_boy
b54d #pragma GCC optimize(3)
302f #include <bits/stdc++.h>
421c using namespace std;
426f #ifdef __LOCAL_DEBUG__
59a8 # define _debug(fmt, ...) fprintf(stderr, "\033[91m[%s␣%3d]:␣" fmt "\n\033[0m", \
1a94   __func__,__LINE__, ##__VA_ARGS__)
a8cb #else
0c29 # define _debug(...) (void(0))
1937 #endif
d54b #define PB(x) push_back(x)
8f39 #define rep(i,l,r) for (int i = l,_ = r;i< _;i++)
aa2e #define REP(i,l,r) for (int i=l,_=r;i<=_;i++)
7e99 #define leave(x) do {cout<<#x<<endl;fflush(stdout);return 0;}while (0);
c33e #define untie do{ios::sync_with_stdio(false);cin.tie(nullptr);cout.tie(nullptr);}while (0)
#define range(x) x.begin(),x.end()                   aaca
typedef long long LL;                                5cad
typedef long long ll;                                4085
typedef vector<int> vi;                              76b3
typedef vector<ll> vl;                               3a45
typedef long double db;                              2bc8
typedef pair<int,int> pii;                           3688
typedef pair<ll,ll> pll;                             0d99
const int inf = 0x3f3f3f3f;                          a7c7
const ll inf_ll = 0x3f3f3f3f3f3f3f3fLL;              a744
mt19937 wdy(time(0));                                526f
/************* header ******************/           5862
int main(){                                          3117
    return 0;                                        7021
}                                                    95cf
```

8.2 FORMULA

```
C(n,m)%2 = (n&m)==m                                                       0f11
约瑟夫问题:                                                                674a
F(n,m) = 有n个人(0,1,2,..,n-1), 每次杀掉编号为(x + m) %n的人，最终的幸存者。   760a
F(n,m) = (F(n-1,m) + m ) % n                                              9a71
```