

思路:BFS/FLOYD+二分

通过bfs/floyd求出各店间的最短路径（题解使用floyd），求出最短路径后，面对每次查询，

简单推算可以得之，我们只需要利用等比数列求和公式求相应的答案即可。因此我们可以通

过二分查找，在log级别的时间复杂度内求出相应的答案。

时间复杂度为： $O(n^3+q\log s)$

```
#include<bits/stdc++.h>
using namespace std;
int n,m,u,v,len;
int q,a,b,s;
int ma[110][110];
void floyd()
{
    for(int k=1;k<=n;k++)
        for(int i=1;i<=n;i++)
            for(int j=1;j<=n;j++)
                ma[i][j]=min(ma[i][j],ma[i][k]+ma[k]
[j]);
}
long long mi(int x,int y)
{
    long long s=1;
```

```

        for(int i=1;i≤y;i++)
            s*=x;
        return s;
    }
    bool judge(int x)
    {
        if(x==1)
        {
            if(len>s)
                return 0;
            else
                return 1;
        }
        long long ans=(mi(x,len+1)-x)/(x-1);
        if(ans>s)
            return 0;
        else
            return 1;
    }
    int main()
    {
        cin>>n>>m;
        for(int i=1;i≤n;i++)
            for(int j=1;j≤n;j++)
                if(i≠j)
                    ma[i][j]=0x3f3f3f3f;
        for(int i=1;i≤m;i++)
        {
            cin>>u>>v;
            ma[u][v]=ma[v][u]=1;
        }
        floyd();
    }

```

```
cin>>q;
while(q-->0)
{
    cin>>a>>b>>s;
    len=ma[a][b];
    int l=0,r=(int)pow(2e9,1.0/len);
    int mid=(l+r)/2;
    while(l+1<r)
    {
        if(judge(mid))
            l=mid;
        else
            r=mid;
        mid=(l+r)/2;
    }
    if(judge(r))
        cout<<r<<"\n";
    else
        cout<<l<<"\n";
}
return 0;
}
```