1001

离散化之后，这题等价于两个操作：

1. 区间取 max
2. 计算 (2 × 非0的区间长度和) + $\sum |a_i - a_{i+1}|$

答案的前半部分是个区间覆盖，我们只需要考虑后半部分。

操作 1 可以用 segment tree beats 转化成对区间内的最小值集体增加一个数的操作，且这次操作不会让区间最小值变得比严格次小值更大。

唯一的问题在于，怎么在进行区间最小值修改的同时，更新区间内 $\sum |a_i - a_{i+1}|$。这个只需要维护区间内的最小值的段数，或者更具体地，维护有多少个间隔满足 a_i, a_{i+1} 中恰好只有一个是区间最小值。假设这样的间隔有 k 个，那么一次对最小值加 a 的操作会让答案减少 ak，这样就可以维护了。

时间复杂度 $O(n \log n)$。

1002

考虑一个构造，对于 $i \in [3, n]$，如果 i 是质数，则加上边 $(2, i)$，如果 i 是合数，则加上边 (d, i)，其中 d 是 i 的某一个不等于 $1, i$ 的约数。现在得到的结果是一棵生成树，同时因为选择的所有边都是 i 的最小边，所以这棵树一定是最小生成树。

因此可以得到答案为 $\frac{(n+3)(n-2)}{2} + [2, n]$ 中的质数和。

质数和可以用分段打表、各种数论筛法来求。

1003

结论：任意一种将 a_i 最小的快递最后取的方案均最优。

证明：观察发现，除了最后一个快递，其余快递都需要从 k 出发走到 a_i，再走回 k。因此我们假设取完最后一个快递后仍然需要走回 k ,才可以走回 1 ,然后考虑最后一个快递对答案的影响。

不难发现，若最后一个快递坐标为 x，则答案的该变量为 $-2 \times \max(0, k - x)$。因此取 a_i 最小的快递最优。

时间复杂度 $O(\Sigma m)$。

1004

定义

- V_i 表示权值第 i 大的节点，K_i 表示 V_i 的权值
- L_i 表示权值最大的 i 个节点构成的集合, $L_0 = \emptyset$
- 对于集合 A, $W(A)$ 表示所有的起点 s, 使得 baby volcano 有一个必胜策略可以让棋子经过 A.

我们要求的就是对每一个顶点 s, 找到最小的 i 使得 $s \in W(L_i)$ 并输出 K_i.

然后我们从小到大依次枚举 i, 通过 $W(L_{i-1})$ 来计算 $W(L_i)$. 首先对于 $i = 0$, 显然 $W(L_0) = \emptyset$.

然后如果 $W(L_{i-1})$ 已经算出来了，那么首先令 $W(L_i) := W(L_{i-1})$，然后在这个基础上，从 V_i 出发进行广搜就可以找到 $W(L_i)$ 相对于 $W(L_{i-1})$ 多出来的节点。广搜就是从 V_i 出发在反图上搜索，对于一个 baby volcano 控制的节点，如果他有一条出边属于 $W(L_i)$，那么它就属于 $W(L_i)$, 对于一个 baby evil 的节点，必须要他所有的出边都属于 $W(L_i)$，他才属于 $W(L_i)$。

这个搜索的均摊时间是 $O(n+m)$.

1005

这是一个 nim 游戏，所以只需知道数字 k 的 nim 函数值 $f(k)$ 就可以了。

结论：$f(k)$ 等于 k 的奇质因子个数 + [k 为偶数]。这个结论只要随便打一段表应该都能发现。

证明只要对着归纳就可以了，只需要证明 $f(k) = \mathbf{mex}_{d|k,d>1}(d\text{ 个 } f(k/d)\text{ 异或起来})$。首先，$f(k/d)$ 只在一种情况下可能大于等于 $f(k)$，那就是 d 是 2 的幂次，k/d 是偶数，但是这时因为 d 是偶数，所以异或起来一定是 0。因此，我们可以得到右侧的式子一定小于等于 $f(k)$。

要证明右侧的式子大于等于 $f(k)$，只需要对每一个 $i \in [0, f(k)-1]$，都构造一个 d 使得右侧值为 i。$i = 0$ 的时候可以取 $d = k$，其他情况只需要从 k 的奇质因子中随便取 i 个乘起来，就是一个合法的 d 的构造。

所以问题变成了如何给输入的所有数分解质因数。这只要先筛出 $[1, \sqrt{10^9}]$ 中的所有质数，然后用这些数来试除就可以了。

1006

按照拓扑序从后往前，每次只需验证每个 limits 的左右极限是否相同即可。

1007

不难发现答案就是字符串 s 中出现次数最多的那个字母的出现次数，设这个值为 W^*。

这是因为对于任何一种排列，其最长border chain的最后一个字母肯定是相等的，故答案不会超过 W^*.

然后把出现次数最多的字母都放到最前面就可以达到 W^*

故答案就是 W^*

1008

动态规划。用 `dp[i][l][r][0/1]` 表示现在在第 i 层($y=i$)，但是还没找到第 i 层开着的门，已经探索过的区间是第 l 扇门到第 r 扇门，当前在 l(对应最后一维是0) 还是当前在 r(对应最后一维是1)，此时到终点的期望最短路. 用 `f[i][x]` 表示现在是 $y=i$, 从第 x 们穿出来后的期望最短路。

转移就是枚举一下下一扇门走的是 $l-1$ 还是 $r+1$, 通过最后一维 01 值来算距离，然后有 $p = \frac{k_i}{i-(r-l+1)}$ 找到开着的门，有 $1-p$ 的概率找到关着的门。如果找到了开着的门，就从 `f[i][l-1]` 或者 `f[i][r+1]` 转移过来。否则从 `dp` 转移过来。

`f` 可以直接从 `dp[i+1]` 转移过来

1009

首先介绍一下 bluestein 算法。

设我们需要求出 $V(j) = \Sigma_{i=0}^{n} a_i \omega^{ij}$ 的值。

观察到 $i \times j = \binom{i+j}{2} - \binom{i}{2} - \binom{j}{2}$,将其代入上面的 ω^{ij} 中可以得到:

$$V(j)\omega_n^{\binom{j}{2}} = \sum_{i=0}^{i<n} a_i \omega_n^{-\binom{i}{2}} \omega_n^{\binom{i+j}{2}}$$

这个式子可以看成是$C(x) = \sum \omega_n^{-\binom{i}{2}} x^i$对$B(x) = \sum a_i \omega_n^{\binom{i}{2}} x^i$做一次减法卷积的结果，然后对每一项系数乘上一个对应常数的值。

同样的，由于模数是一个较小的质数 p ，因此假设我们求出了 $\omega_{p-1}^0, \omega_{p-1}^1, \cdots, \omega_{p-1}^{p-2}$ 的点值，也就可以求出 $1 \sim p-1$ 的点值，就解决了多点求值问题。而求出 $\omega_{p-1}^0, \omega_{p-1}^1, \cdots, \omega_{p-1}^{p-2}$ 的点值部分可以采用上面提到的 bluestein 算法解决。

同时，观察不难发现，求解多点求值部分时我们本质上运行了一次 DFT ，这启发我们是否可以使用一次 IDFT 解决多点插值问题。而答案是肯定的。如果我们已经知道 $\omega_{p-1}^0, \omega_{p-1}^1, \cdots, \omega_{p-1}^{p-2}$ 处的点值，则可以通过一次 IDFT 反向求出其系数。

因此，首先我们可以通过下降幂与点值的互相转换在 $O(n\log n)$ 的复杂度内求出 $1 \sim p-1$ 处的点值。然后通过一次 IDFT 求解系数即可。 IDFT 部分仍然可以使用 bluestein 算法解决即可。

时间复杂度 $O(p\log p)$。

1010

直接按照题意模拟

时间复杂度：$O(n)$

1011

可以发现，对于矩阵 K，如果有除了 $K[1][1]$ 以外的数大于 0 的话，那么最后结果一定收敛到全 0

进行一次 $C(A,K)$ 的过程可以看成是 A 中每个元素向左上的一个 3×3 的矩阵进行带权扩散的过程

假设 $K[2][2] = \gamma > 0$，那相当于对于 A 中每个元素 $A[x][y]$，卷积后有 $C[x-1][y-1] += \gamma A[x][y]$，如果 $(x-1, y-1)$ 不在矩阵中的话，那么相当于经过这次卷积后 A 的和减少了 $\gamma A[x][y]$

不如设 A 的和为 sA，那么最大值至少是 $\frac{sA}{n^2}$，然后我们可以知道在经过 $O(n)$ 次卷积后，A 的元素之和会至少减少 $\gamma^n \frac{sA}{n^2}$

所以 $\sum C(A,K)^t[i][j] \le (\sum A[i][j])(1-\frac{\gamma^n}{n^2})^{t/n}$

因为 $1-\frac{\gamma^n}{n^2} < 1$，所以当 $t\to\infty$ 时，$t/n\to\infty$，所以 $\lim_{t\to\infty}\sum C(A,K)^t = 0$

所以当只有 $K[1][1]$ 不等于 0 时，结果就是原矩阵 A，否则就是全 0 矩阵

1012

$|x-y| \le K$ 等价于 $x+K \le y$ 且 $y+K \le x$

所以相当于有这么几个条件：

- $x \le A$
- $y \le B$
- $x+K \le y$
- $y+K \le x$
- $x\ xor\ y \le W$

我们可以考虑从低位往高位确定 x, y 的每一位，进行一个数位 DP

令 $f[bit][xA][yB][xKy][yKx][xyw][cxK][cyK]$ 表示，已经确定了 x, y 的最低的 bit 位的情况下：

- xA 表示是否有 $x \le A$
- yB 表示是否有 $y \le B$
- xKy 表示是否有 $x+K \le y$
- yKx 表示是否有 $y+K \le x$
- xyw 表示是否有 $x\ xor\ y \le W$
- cxK 表示如果只看最低 bit 位的话，$x+K$ 是否产生了进位
- cyK 表示如果只看最低 bit 位的话，$y+K$ 是否产生了进位

转移时，枚举一下 x, y 第 $bit+1$ 位的值，以上这些状态都是自然而然地可以推导的

时间复杂度：$O(\log A)$

1013

对于每个 $f_i(x)$，可以发现他们都是 $f_1(x)$ 的若干阶导的线性组合，所以我们可以把它表示成 $\sum_{j=0}^{n} d_{i,j} f_1(x)^{(j)}$ 的形式

我们不妨设 $g_i(x) = \sum_{j=0}^{n} d_{i,j} x^j$

我们的思路是先求出 $g_n(x) = \sum_{i=0}^{n} d_{n,i} x^i$，然后根据 $g_n(x)$ 去求出 $f_n(x)$

可以发现，因为 $f_i(x) = b_i f_{i-1}(x)' + c_i f_{i-1}(x)$

所以 $g_i(x) = b_i g_{i-1}(x) x + c_i g_{i-1}(x) = (b_i x + c_i) g_{i-1}(x)$

所以 $g_n(x) = \prod_{i=2}^{n} (b_i x + c_i)$

我们可以利用分治 FFT 求出 $g_n(x)$

然后考虑一下如何求 $[x^i] f_n(x)$，（这个符号的意思是这个多项式的 x^i 的系数）

考虑每一阶导对它的贡献，如果是 $[x^j] f_1(x)$ 对它产生贡献的话，因为求一次导会降一个次数，所以一定是求了 $j-i$ 次导之后才会产生这样的贡献

所以可以得出：

$[x^i] f_n(x) = \sum_{j=i}^{n} d_{n,j-i} (\prod_{k=i+1}^{j} k)([x^j] f_1(x))$

令 $B[t] = d_{n,t}$

令 $P[k] = \prod_{i=1}^{k} i$

令 $C[t]=[x^t]f_1(x)$

那么 $[x^i]f_n(x)=\frac{1}{P[i]}\sum_{j=i}^{n}B[j-i]P[j]C[j]$

这是一个下标为减法的卷积形式，通过把下标翻转后可以变成标准的卷积形式，使用 FFT 优化即可

时间复杂度：$O(n\log^2 n)$