IOI2016 中国国家集训队第一次作业
试题泛做表格

江苏省常州高级中学 张志俊

December 25, 2015

目录

1 ANDOOR

题目大意

给定平面直角坐标系内 $w \times h$ 的矩形区域，以及 n 个圆 (x_i, y_i, r_i)。求在矩形区域内所有 n 个圆并的周长。

时间限制：6 s

数据规模

- $1 \le t \le 10^3$，$1 \le n \le 10^3$，$1 \le w, h, r_i \le 10^3$，$0 \le x_i, y_i \le 10^3$，$\sum n_i \le 5 \times 10^3$。

算法分析

首先可以去除被其他圆完整覆盖或者整体处于矩形区域外的所有圆，则对于剩下的任意圆 i 而言，其对并的周长的贡献为所有未被覆盖的圆弧，同时易知任意异于 i 的圆 j 所覆盖的圆弧必然是一段连续的极角区间，并且矩形区域的边界也是如此，因此只需对圆 i 求得所有极角区间的并即可。

时间复杂度：$\Theta(n^2 \log n)$

空间复杂度：$\Theta(n)$

2 ANUDTQ

题目大意

给定一棵 n 个节点的有根树，每个节点的初始权值为 v_i，要求在线处理 m 个以下四类操作：

- $1\ key\ value$：新增一个权值为 $value$ 的节点，其父节点为 key 号点；
- $2\ key\ value$：在以 key 号点为根的子树中，所有节点的权值均加上 $value$；
- $3\ key$：删除以 key 号点为根的整棵子树；
- $4\ key$：在以 key 号点为根的子树中，统计所有节点的权值之和。

时间限制：2 s

数据规模

- $1 \leq n \leq 10^5$，$1 \leq m \leq 10^5$。

算法分析

首先考虑本题的离线版本，则由于所有操作均是以子树为单位的，因此不妨先将所有节点全部加入到树中，然后尝试利用 Dfs 序来进行维护。此时，易知上述四类操作分别表现为：

- 激活序列中的某个位置；
- 对一段连续区间加上同一个数；
- 强制使得一段连续区间失效；
- 统计一段连续区间内的总和。

不难发现，这些操作均可借助线段树来实现。当需要在线处理所有操作时，由于无法提前预知整棵树的形态，因此难以随时确定某棵子树在 Dfs 序中的具体位置。因此，我们考虑改用括号序列来解决本题，即同一标号的节点在该棵子树的首尾各出现一次，则以 x 号点为根的子树在序列中即为以两个 x 为端点的整段区间。于是，本题只需借助平衡树即可完成相应的所有操作。

时间复杂度：$\Theta(m \log n)$

空间复杂度：$\Theta(n + m)$

3 BAKE

题目大意

已知市场上共有 10 种不同商品，每种商品至多有 3 种不同规格。同时，销售区域包括 10 个不同省份，每个省内至多有 20 个不同城市，每个城市至多被划分为 5 个片区。要求处理共计 s 次以下两种事件：

- 增加一份订单，已知订单的商品类型、销售区域、买家性别及年龄、购买数量，其中商品类型和销售区域参数可能部分缺省；
- 查询满足给定参数要求的所有订单中购买数量的总和，其中商品类型和销售区域参数可能部分缺省，并且买家年龄是一段给定区间。

时间限制：$1 \sim 10$ s

数据规模

- $1 \leq s \leq 10^5$。

算法分析

由于题中所有参数的取值范围都极小，因此本题其实是一道简单的模拟题，只需根据输入格式解析出订单信息即可。对于参数部分缺省的问题，则显然可以对每维参数都新增一个标号表示总和。同时，由于询问中的买家年龄是一段连续区间，于是可以借助树状数组来加速。

时间复杂度：$\Theta(s)$

空间复杂度：$\Theta(s)$

4 BB

题目大意

已知有连续 n 个空位，要求从中选择出若干个空位，使得任意连续 m 个空位中至少有 k 个空位被选择。要求最小化所选空位个数，并统计此情形下的合法方案总数。

时间限制：5 s

数据规模

- $1 \leq t \leq 25$，$1 \leq k \leq m \leq 50$，$m \leq n \leq 10^9$。

算法分析

首先考虑最小化所选空位个数，若令 $n = pm + r$（$0 \leq r < m$），则有：

- 当 $r = m - k$ 时，我们不妨将 n 个空位每 m 个分为一组，则第 $p+1$ 组仅有 r 个空位。显然，此种情形下的最优方案即为选择前 p 组中每组内的后 k 个，并且此时最优方案唯一；
- 当 $r < m - k$ 时，则此时的最优方案仍然为 kp，而所有最优方案均需满足任意一组内的前 r 个空位都不选择，于是问题等价于 $n' = p(m-r)$、$m' = m - r$、$k' = k$ 时的情形；
- 当 $r > m - k$ 时，则此时的最优方案为 $k(p+1) - (m-r)$，而所有最优方案均需满足任意一组内的后 $m - r$ 个空位都选择，于是问题等价于 $n' = (p+1)r$、$m' = r$、$k' = k - (m-r)$ 时的情形。

通过上述转化，则我们仅需解决所有满足 $m|n$ 的情形。此时，易知每一组中必然恰好选择了 k 个空位，我们将其在组内所处的位置从大到小排成一列，则选择的所有空位构成一个 $k \times p$ 的矩阵，其中所有元素均在 $[1, m]$ 内，并且每一列的元素单调递减。同时不难证明，选择空位的某种方案合法，当且仅当矩阵中每一行的元素单调不增。

因此，问题转化为求相应的 $k \times p$ 矩阵的方案数。于是，根据杨氏矩阵的勾长公式，可以推知：

$$Ans = \prod_{1 \leq i \leq k, 1 \leq j \leq p} \frac{m + j - i}{k - i + p - j + 1}$$

在上式中，由于分式上下绝大部分元素可以被相互约去，可以发现分别只有 $k(m-k)$ 个不重复元素，因此只需直接计算这部分表达式即可。

时间复杂度：$\Theta(mk)$

空间复杂度：$\Theta(1)$

5 *BICKER

题目大意

给定一张 n 个点的无向完全图，图中任意一条边 (i,j) 有两个非负权值属性 $d(i,j)$ 和 $q(i,j)$，要求将所有 n 个点分为两个非空集合 S 和 T。对于某种划分方案 (S,T)，记 $q(S,T)=\sum_{(i,j)} q(i,j)$ 和 $d(S,T)=\sum_{(i,j)} d(i,j)$ 分别为总损失和总收益，其中 (i,j) 是任意一条跨集合的边。求一种可能的划分方案使得比值 $\frac{q(S,T)}{d(S,T)}$ 尽量小。

时间限制：5 s

数据规模

- $1 \le t \le 30$，$2 \le n \le 500$，$1 \le d,q \le 10^4$，$1 \le v,w \le 10^4$。

算法分析

由于这一模型本身是 $NP-hard$ 问题，且在评分时并不要求每个测试点都给出最优解，于是本题只能从随机化算法等方面进行考虑。鉴于现场表现较好的代码均是卡时结合随机化，因此接下来仅讨论部分实践证明较为有效的随机策略及技巧。

首先，由于给定的评分方式是基于每个数据中所有测试点的综合表现给分，而且允许与参考值有一定的相对误差，因此对于每个测试点我们只需找到某个较为接近最优解的近似解即可。其次，当数据中边权的随机程度较高时，因为不同大小的损失和收益相对交错分布，所以 $|S|$ 的不同取值对比值的影响微乎其微，于是我们可以首先暴力枚举所有 $|S| \le 2$ 的情形，此时便可得到一个较优解，这部分代价为 $\Theta(n^2)$。同时，实践也表明在纯随机的前提下，上述解与最优解都极其接近。

接下来，我们考虑什么样的数据可以使得上述解与最优解存在较大差异。显然，上述解之所以能极其逼近最优解，原因在于损失与收益的分布过于接近，因此接下来我们尝试构造一些极端情形。首先，我们在保证 $|S|$ 与 $|T|$ 差距不是很大的前提下，将所有 n 个点随意分成两个部分作为期望的 S 和 T。为保证最优解的比值较小，任取一个 $(0,\frac{1}{2})$ 内的实数 c，然后对整张图随意连边，其中边权满足：

- 当边 (i,j) 的两个端点同属 S 或 T 时：若连边表示损失 $q(i,j)$，则控制其权值下限为 $(1-c)\cdot MaxValue$；若连边表示收益 $d(i,j)$，则控制其权值上限为 $c\cdot MaxValue$；
- 当边 (i,j) 的两个端点分属不同部分时：若连边表示损失 $q(i,j)$，则控制其权值上限为 $c\cdot MaxValue$；若连边表示收益 $d(i,j)$，则控制其权值下限为 $(1-c)\cdot MaxValue$。

易知，按上述方法构造的初始图的理论最优解与期望由 (S,T) 给出的解几乎不会有太大差距。因此，在这种情形下，我们尝试利用爬山法来进行适当调整。在选取初始解时，为了在之后的随机调整中能遍及尽量多的情形，我们可以适当选取多种有一定差异的 $|S|$。同时，由于一般的随机调整每次仅变动极少的对象，为了加快逼近最优解的速度，我们不妨每次随机一个 $1 \sim n$ 的排列，然后按此顺序依次尝试改变单个对象，并在此过程中立即执行那些使得状态变优的改变。

另外，若对随机函数要求较高，可以尝试诸如梅森旋转等方法生成随机数。（C++11 中有现成的实现可以使用）

上述算法在 $Codechef$ 上可得 $0.997pt$。

时间复杂度：$\Theta(n^2 + CountRandom \cdot n)$
空间复杂度：$\Theta(n^2)$

6 BTREE

题目大意

给定一棵 n 个节点的无根树，且所有边权均为 1。要求处理 q 次询问，每次询问给定 k_i 个特殊点 a_j，以及每个点的覆盖范围 r_j，要求统计被覆盖的总点数。

时间限制：2.5 s

数据规模

- $1 \leq n, q \leq 5 \times 10^4$，$\sum k_i \leq 5 \times 10^5$。

算法分析

首先考虑本题的简化版本，即所有询问均满足 $k_i = 1$ 的情形。此时不难发现，我们只需在点分治的过程中维护到每层分治中心的所有距离信息即可 $\Theta(\log n)$ 处理单次询问。为处理 $k_i > 1$ 的情形，我们考虑建出所有特殊点构成的虚树，则虚树上所有点的实际覆盖范围可以通过 Dfs 求得。于是，虚树上任意两条树边间子树的贡献相互独立。此时，对于虚树上的任意一条树边 (x, y) 而言，若 x 和 y 的覆盖范围相离，则显然只需直接查询 x 和 y 即可。否则，该条树边上必然存在某点 z，使得 z 恰好位于 x 和 y 公共覆盖范围的中点处，则不难证明此部分子树的总贡献即为 x 和 y 的总贡献除去 z 的贡献。因此，任意询问均可被分解为 $\Theta(k_i)$ 次单点查询。

时间复杂度：$\Theta((n + \sum k_i) \log n)$

空间复杂度：$\Theta(n \log n)$

7 BWGAME

题目大意

给定一个 $n \times n$ 的矩阵，已知其中第 i 行仅有 $[l_i, r_i]$ 为黑格，其余均为白格。定义任意一个 $1 \sim n$ 的排列 p 是合法的，当且仅当对任意 $1 \leq i \leq n$ 满足 (i, p_i) 为黑格。定义具有奇数个逆序对的合法排列为奇排列，反之则为偶排列，要求比较奇排列总数和偶排列总数的大小关系。

时间限制：2 s

数据规模

- $1 \leq t \leq 15$，$1 \leq n \leq 10^5$。

算法分析

通过一些观察不难发现，我们不妨将黑格视作 1，而将白格视作 0，则偶排列总数与奇排列总数之差即为该矩阵的行列式值。于是，我们考虑模拟消元的整个过程。此时，若我们每次选取以当前列 i 为左端点的最小右端点所在行作为主行，则易知上述选取主元的方式可以始终保证每行中仅有一段连续的 1。因此，我们对每一列均借助可并堆维护以该列为左端点的所有区间，则消元过程即为堆的合并操作。

时间复杂度：$\Theta(n \log n)$

空间复杂度：$\Theta(n)$

8 CARDSHUF

题目大意

给定一个初始时 $1 \sim n$ 的递增序列，要求依次执行 m 次操作 (a_i, b_i, c_i)：

- 取走当前序列的前 a_i 个数；
- 再取走当前序列的前 b_i 个数；
- 将第一步取走的 a_i 个数按原顺序拼接在当前序列的头部；
- 取走当前序列的前 c_i 个数；
- 将第二步取走的 b_i 个数逆序拼接在当前序列的头部；
- 再将第四步取走的 c_i 个数按原顺序拼接在当前序列的头部。

要求计算出依次执行完所有操作后的序列。

时间限制：6 s

数据规模

- $1 \leq n, m \leq 10^5$

算法分析

本题是一道较为基础的平衡树题，取走前若干个数相当于删除一段区间，拼接一段数相当于插入一段区间，而转为逆序则相当于翻转一段区间，因此本题只需借助 $Splay$ 即可解决。

时间复杂度：$\Theta(m \log n)$

空间复杂度：$\Theta(n)$

9 CBAL

题目大意

给定一个长度为 n 的字符串 p，要求在线处理 q 个询问 $(l, r, type)$：

$$Ans_{l,r,type} = \sum_{s_{i,j}} |s_{i,j}|^{type}$$

其中，$s_{i,j}$ 是 p 中所有满足 $l \le i < j \le r$ 的平衡子串，定义某个字符串是平衡的，当且仅当该字符串中的所有字符可以被平分为两个相等的可重集。

时间限制：2 s

数据规模

- $1 \le t \le 10^5$，$\sum n_i \le 10^5$，$\sum q_i \le 10^5$，$0 \le type \le 2$。

算法分析

首先易知，所谓平衡字符串即每种字符出现次数均为偶数的字符串，因此我们并不关心每种字符的具体出现次数，而只需知晓其奇偶性。于是，若记 $cnt_{i,c}$ 为前 i 个字符中 c 出现次数的奇偶性，则子串 (i, j) 是平衡的当且仅当 $cnt_{i-1,c} = cnt_{j,c}$ 对任意字符 c 成立。此时，若我们将所有字符的奇偶性压位存储，即 $state_i = \sum_c cnt_{i,c} 2^c$，则上述成立条件等价于 $state_{i-1} = state_j$，此处特别定义 $state_0 = 0$。

通过上述转化，问题变为：给定一个长度为 $n+1$ 的整数序列 $state_0, state_1, \ldots, state_n$，要求在线回答 q 个询问 $(l, r, type)$：

$$Ans_{i,j,type} = \sum_{\substack{l-1 \le i < j \le r \\ state_i = state_j}} (j-i)^{type}$$

对于该问题，我们首先可以通过离散化将整数的范围缩小至 $[0, n]$ 以内。其次，由于指数不超过 2，因此我们可以通过分别维护前缀中每类整数相应下标的 0、1、2 次方和，以在线性时间内处理单次询问。对于多组询问，若允许离线处理，则易知我们可以利用莫队算法以 $\Theta((n+q)\sqrt{n})$ 的代价处理所有询问。

对于强制在线的询问，我们同样可以利用类似的想法。假设我们将序列以 k 为块长分块，则整个序列共被分为 $\lceil \frac{n}{k} \rceil$ 块。为方便接下来的表述，记 $begin_i = ik$ 为第 i 块的块头下标，$end_i = (i+1)k - 1$ 为第 i 块的块尾下标，$block_i$ 表示 i 所处块的标号，f_{l_1,r_1,l_2,r_2} 为左端点处于区间 $[l_1, r_1]$、右端点处于区间 $[l_2, r_2]$ 的所有平衡子串的总贡献。在预处理阶段，我们分别求出所有 $pre_{i,j,tp} = Ans_{begin_i,j,tp}$ 和 $suf_{i,j,tp} = Ans_{j,end_i,tp}$，易知此部分总代价为 $\Theta(\frac{n}{k} \cdot n) = \Theta(\frac{n^2}{k})$。

对于每个询问 (l, r, tp)，若 $block_l = block_r$，则上述线性算法即可求得答案，代价为 $\Theta(k)$。否则，若简记 $bl = block_l$，$br = block_r$，$l' = end_{bl}$，$r' = begin_{br}$，则我们可以借助如下等式来计算：

$$Ans_{l,r,tp} = pre_{bl+1,r,tp} + suf_{br-1,l,tp} - pre_{bl+1,r'-1,tp} + f_{l,l',r',r}$$

上述等式证明如下：

$$\begin{aligned}
\text{右边} &= f_{l'+1,r'-1,l'+1,r'-1} + f_{l'+1,r'-1,r',r} + f_{r',r,r',r} \\
&\quad + f_{l,l',l,l'} + f_{l,l',l'+1,r'-1} + f_{l'+1,r'-1,l'+1,r'-1} \\
&\quad - f_{l'+1,r'-1,l'+1,r'-1} \\
&\quad + f_{l,l',r',r} \\
&= f_{l,l',l,l'} + f_{l'+1,r'-1,l'+1,r'-1} + f_{r',r,r',r} \\
&\quad + f_{l,l',l'+1,r'-1} + f_{l'+1,r'-1,r',r} + f_{l,l',r',r} \\
&= f_{l,r,l,r}
\end{aligned}$$

于是，每次询问都只需额外计算 $f_{l,l',r',r}$ 即可。由于 $|l'-l|$ 和 $|r-r'|$ 均不超过 k，因此只需在上述线性算法的基础上稍加修改即可求得 $f_{l,l',r',r}$，单次代价同样为 $\Theta(k)$。

鉴于本题中 n 和 q 同阶，因此当块长取 $k=\sqrt{n}$ 时总代价最低。

时间复杂度：$\Theta((n+q)\sqrt{n})$

空间复杂度：$\Theta(n\sqrt{n})$

10 CHANGE

题目大意

给定 n 种面额为 d_i 的硬币各无限枚，并且保证任意两种面额互质，要求统计凑出恰好 c 单位面额的不同方案总数。

时间限制：3 s

数据规模

- $1 \leq t \leq 5$，$1 \leq n \leq 50$，$1 \leq c \leq 10^{100}$，$1 \leq d_i \leq 500$。

算法分析

首先可以发现，若将 d_i 视作为物品的体积，则本题即等价于完全背包问题。因此，不难想到我们可以利用生成函数来解决本题，则 $Ans = [x^c]F(x)$，其中有：

$$F(x) = \prod_{i=1}^{n}(1 - x^{d_i})^{-1}$$

时间复杂度：$\Theta(n^2 d)$

空间复杂度：$\Theta(nd)$

11 *CHAORNOT

题目大意

给定 m 个非负整数 b_i，要求从中选择尽可能多的非负整数构成序列 $\{a_i\}$，使得 $\{a_i\}$ 中任意三个元素无法构成等差数列。最终评分以序列 $\{a_i\}$ 中的元素个数作为参考。

时间限制：1 s

数据规模

- $1 \leq m \leq 10^5$，$0 \leq b_i \leq 10^5$。

算法分析

由于数据均为随机生成，因此我们考虑按 b_i 从小到大的顺序贪心选取，即若在已有序列中加入当前数不会产生冲突则选取该数，否则忽略该数。此时，易知上述贪心策略显然并不正确，于是我们尝试适当选取某个较大的接受概率 p，然后每次若当前数可以选取，则仅以 p 的概率接受。为使得答案尽可能优，则我们只需重复上述过程若干次即可。

上述算法在 $Codechef$ 上可得 $0.731pt$。

时间复杂度：$\Theta(CountRandom \cdot |\{a_i\}|^2)$

空间复杂度：$\Theta(m)$

12 CHEFBOOK

题目大意

给定 m 个约束条件 $(x_j, y_j, l_j, s_j, t_j)$，要求找到一组非负整数解 p_i 和 q_i，使得对任意一个给定的约束条件均满足 $s_j \leq l_j + p_{x_j} - q_{y_j} \leq t_j$，同时最大化 $\sum_{i=1}^{m} l_j + p_{x_j} - q_{y_j}$。

时间限制：2 s

数据规模

- $1 \leq t \leq 10$，$1 \leq n \leq 100$，$1 \leq m \leq n^2$，$-600 \leq l_j \leq 600$，$-10^3 \leq s_j \leq t_j \leq 10^3$。

算法分析

首先易知，本题中的所有约束条件均可以改写成若干个仅包含两个变量的不等式，且两个变量前的系数必然分别为 1 和 -1。于是不难发现，此时只需借助最小费用最大流即可求得最优解。为求出一组具体的最优解，则我们可以考虑借助差分约束系统。同时，由于利用题中给定的所有约束条件仅能求得一组可行解，因此，对于图中任意一条流量非零的边，则其对应的约束条件均应恰好满足，即只需在差分约束系统中增加相应的限制即可。

时间复杂度：$\Theta(MinCostFlow(n, m))$

空间复杂度：$\Theta(n + m)$

13 CIELQUAK

题目大意

给定一张 $r \times c$ 的网格图，其中每条边均有 p 的概率不存在，并且相互统计独立。要求计算 $(1,1)$ 和 (r,c) 连通的概率。

时间限制：8 s

数据规模

- $1 \leq t \leq 50$，$1 \leq r \leq 8$，$1 \leq c \leq 10^{18}$，$0.1 \leq p \leq 1$。

算法分析

本题显然可以利用最小表示法进行 DP，但由于需要记录轮廓线上所有点与 $(1,1)$ 的连通性，其总状态数可达 3500 级别，因此我们仅能处理 c 较小的情况。此时，通过实践可以发现当 c 趋向于无穷大时，比值 $k = \frac{Ans_{r,c+1}}{Ans_{r,c}}$ 趋向于一定值。于是，我们不妨选取某个适当的阈值 c'，则 $Ans_{r,c} \approx Ans_{r,c'} \cdot k^{c-c'}$。在具体实现时，当 c' 取到 50 级别即可保证精度需求。

时间复杂度：$\Theta(CountState \cdot rc')$

空间复杂度：$\Theta(CountState \cdot rc')$

14 CLONES

题目大意

若 n 元函数 $f_{x_0,x_1,\ldots,x_n}=y$ 的定义域和值域均为 $\{0,1\}$，则将 f 称之为布尔函数。给定以下四种由 n 元布尔函数构成的集合：

- Z：由所有满足 $f_{0,0,\ldots,0}=0$ 的函数构成的集合；
- P：由所有满足 $f_{1,1,\ldots,1}=1$ 的函数构成的集合；
- D：由所有满足 $!f_{x_0,x_1,\ldots,x_n}=f_{!x_0,!x_1,\ldots,!x_n}$ 恒成立的函数构成的集合；
- A：由所有仿射函数构成的集合，其中，定义 f 为仿射函数当且仅当 f 满足：若存在一组自变量使得 $f_{x'_1,\ldots,x'_i=0,\ldots,x'_n}=f_{x'_1,\ldots,x'_i=1,\ldots,x'_n}$，则 $f_{x_1,\ldots,x_i=0,\ldots,x_n}=f_{x_1,\ldots,x_i=1,\ldots,x_n}$ 恒成立。

给定 q 次询问 s_i，其中 s_i 为仅由 Z、P、D、A 构成的集合运算表达式，合法的集合运算为交集、并集、补集、差集，要求分别计算每个表达式结果集合的势。

时间限制：1 s

数据规模

- $1\le n\le 100$，$1\le q\le 100$，$1\le |s_i|\le 100$。

算法分析

由于仅有 4 种不同的初始集合，因此所有 n 元布尔函数可被分为 2^4 类。同时，由给定的集合运算的性质可知，同一类中的所有函数必然同时（不）属于结果集合，因此可以利用 2^4 位二进制数的位运算替代相应的集合运算。于是，问题转化为分别统计每类函数的总数。为方便接下来的表述，记 U 为所有 n 元布尔函数构成的集合，则易知：

- $|U|=2^{2^n}$、$|Z|=|P|=2^{2^n-1}$、$|Z\bigcap P|=2^{2^n-2}$：由于共有 2^n 组不同的自变量，并且对于每组自变量而言均有两种可能的函数值，因此可得 $|U|=2^{2^n}$。对于 Z 而言，由于 $f_{0,0,\ldots,0}$ 必定为 0，而其余函数值均可任意选取，因此可得 $|Z|=2^{2^n-1}$，同理可知 $|P|=2^{2^n-1}$ 以及 $|Z\bigcap P|=2^{2^n-2}$；
- $|D|=2^{2^{n-1}}$、$|D\bigcap Z|=|D\bigcap P|=|D\bigcap Z\bigcap P|=2^{2^{n-1}-1}$：对于 D 而言，由于自变量成对出现，因此可得 $|D|=2^{2^{n-1}}$。同时，由于 $f_{0,0,\ldots,0}$ 和 $f_{1,1,\ldots,1}$ 成对出现，于是可知 $|D\bigcap Z|=|D\bigcap P|=|D\bigcap Z\bigcap P|=2^{2^{n-1}-1}$；
- $|A|=2^{n+1}$、$|A\bigcap Z|=|A\bigcap P|=2^n$、$|A\bigcap Z\bigcap P|=2^{n-1}$：对于 A 而言，易知所有函数值仅由 $f_{0,0,\ldots,0}$ 和所有 $f_{0,\ldots,x_i=1,\ldots,0}$ 决定，因此可得 $|A|=2^{n+1}$。在此基础之上，Z 要求 $f_{0,0,\ldots,0}=0$，而 P 要求 $n+1$ 个基准函数值中恰有奇数个为 1，于是可知 $|A\bigcap Z|=|A\bigcap P|=2^n$ 以及 $|A\bigcap Z\bigcap P|=2^{n-1}$；
- $|A\bigcap D|=2^n$、$|A\bigcap D\bigcap Z|=|A\bigcap D\bigcap P|=|A\bigcap D\bigcap Z\bigcap P|=2^{n-1}$：类似上述情形，$D$ 在 A 的基础之上要求除 $f_{0,0,\ldots,0}$ 以外的 n 个基准函数值中恰有奇数个与 $f_{0,0,\ldots,0}$ 取值不同，于是同理可知 $|A\bigcap D|=2^n$ 以及 $|A\bigcap D\bigcap Z|=|A\bigcap D\bigcap P|=|A\bigcap D\bigcap Z\bigcap P|=2^{n-1}$。

通过上述讨论，则只需利用容斥原理即可求得每类函数的总数。

时间复杂度：$\Theta(\sum |s_i|)$

空间复杂度：$\Theta(|s|)$

15 *CLOSEST

题目大意

给定三维空间中由 n 个点构成的初始点集，要求处理 q 次询问，每次查询初始点集中距离给定点欧几里得距离最近的点。其中，最终评分以求得最优解的比例作为参考。

时间限制：4 s

数据规模

- $1 \leq n, q \leq 5 \times 10^4$。

算法分析

本题不难想到借助 Kd 树来解决，但是由于 Kd 树执行速度略慢，因此需要一定改进。此时，由于本题仅以求得最优解的比例作为评分参考，因此我们宁可舍弃一部分较优解而转为求出少量最优解。于是，我们考虑首先随机选取适当数量的询问暴力求解。此时，由于初始点集中其实存在较多几乎无效的点，即真正有一定概率可能成为最优解的初始点并不多，因此所有询问的最优解基本上较为集中。于是，我们仅保留在选取的询问中成为最优解的初始点，然后对剩余的点集以一定的比例选取，则剩余的所有询问再借助 Kd 树求解即可。

上述算法在 $Codechef$ 上可得 $0.695pt$。

时间复杂度：$\Theta(n \log n + qn^{\frac{2}{3}})$

空间复杂度：$\Theta(n + q)$

16 CNTDSETS

题目大意

对于 n 维空间中的任意点集 S，定义该点集的直径为点集中任意两点间距离的最大值，其中距离为切比雪夫距离。求直径恰好为 d 的本质不同的整点集，两个点集被视为本质不同当且仅当不能通过平移变换相互转化。

时间限制：0.3 s

数据规模

- $1 \leq t \leq 10$，$1 \leq n \leq 10^3$，$1 \leq d \leq 10^9$。

算法分析

首先，为了处理本质不同的限制，易知我们可以将所有点每维坐标的范围都限定在 $[0, d]$ 内，则直径恰好为 d 的限制等价于要求至少存在一维坐标满足：点集中同时存在该维坐标为 0 和 d 的点。同时，由于直径恰好为 d 的点集不便于统计，不难想到转化为统计直径不超过 d 的点集与直径不超过 $d-1$ 的点集。此处，为了解决重复计数的问题，我们规定合法的点集必须满足：对于每一维坐标，点集中均存在至少一个点 x，使得 x 的该维坐标为 0。

接下来，我们讨论直径不超过 d 的合法点集方案数。考虑记性质 p_i 为存在第 i 维坐标为 0 的点，则利用容斥原理即可计算出相应的方案数。此外，由于所有性质并不存在本质差异，因此利用组合数与快速幂即可加速处理过程。

时间复杂度：$\Theta(n^2 + tn \log n)$

空间复杂度：$\Theta(n^2)$

17 CNTHEX

题目大意

给定长度分别为 $1 \sim n$ 的木棍各无限根，要求从中选择 6 根木棍拼成一个严格六边形，并且满足：

- 最长的一根木棍长度至少为 l；
- 剩余的 5 根木棍长度均至多为 x；
- 同种长度的木棍不超过 k 根。

时间限制：6 s

数据规模

- $2 \le l \le n \le 10^9$，$n - l \le 100$，$1 \le x < l$，$1 \le k \le 5$。

算法分析

首先，由于最长木棍的种类数范围较小，我们不妨枚举最长木棍的长度并分别统计，因此接下来我们仅讨论最长木棍长度恰好为 n 的情形。此时，由于本题中有对于同种木棍数量的限制，于是易知木棍不同长度的组合共有 $1+1+1+1+1$、$2+1+1+1$、$2+2+1$、$3+1+1$、$3+2$、$4+1$ 和 5 共 7 种情形。因此，若我们分别统计出上述 7 种情形下的合法方案数，则对于给定的 k 显然可以直接求得答案。

此时，对于上述的任意一种划分方案 $5 = p_1 + p_2 + \cdots + p_k$ 以及所选长度集合 $\{c_1, c_2, \ldots, c_k\}$ 而言，我们统计满足下列性质的方案总数：

- 所选木棍长度 c_i 允许存在重复，但必定满足 $1 \le c_i \le x$；
- 对于任意 $1 \le i \le k$，选择恰好 p_i 根长度为 c_i 的木棍；
- $\sum_{i=1}^{k} p_i c_i \le n$。

不难发现，满足上述性质的恰为所有非法方案，而总方案数显然为 x^k，于是合法方案数即为两者之差。同时，上述 7 种情形相互间的重复计数问题显然可以利用容斥原理解决，而具体的容斥系数只需手动计算即可。而对于剩余木棍长度至多为 x 的限制，则显然也可以再次利用容斥原理来解决。因此，对于某种具体的划分方案 $5 = p_1 + p_2 + \cdots + p_k$ 而言，消除上限后的问题等价于求解以下不定方程的整数解组数：

$$p_1 x_1 + p_2 x_2 + \cdots + p_k x_k + y = n \qquad x_i > 0, y \ge 0$$

易知，若将所有 x_i 的取值范围调整为非负整数，则有：

$$p_1 x_1 + p_2 x_2 + \cdots + p_k x_k + y = n - 5 \qquad x_i, y \ge 0$$

此时，对于上述的 7 种具体情形，我们不难分别推得以相应的未知数作为求和下标的多重求和式，其中仅涉及 n 的多项式以及与 n 有关的组合数。于是，通过一些简单的计

算化简多重求和式，在上述任意一种情形下，我们均可以推得一个便于快速计算的简洁表达式。

时间复杂度：$\Theta((n-l)\log n)$

空间复杂度：$\Theta(1)$

18 CONNECT

题目大意

给定一张 $n \times m$ 的网格图，其中每个网格 (i, j) 内恰有一个 $[-1, nm]$ 的整数 $a_{i,j}$，并且选取网格 (i, j) 的代价为 $v_{i,j}$。要求选择一个不包含 -1 的四连通块，并且满足该四连通块内包含至少 k 种不同的正整数，最小化选取网格的总代价。

时间限制：7 s

数据规模

- $1 \leq n, m \leq 15$，$1 \leq k \leq 7$，$1 \leq v_{i,j} \leq 10^5$。

算法分析

首先考虑本题的简化版本，即网格内出现的不同正整数恰好仅有 k 种，则此时即为经典的斯坦纳树问题，并且单次求解的代价为 $\Theta(3^k nm)$。因此，由于 k 的范围较小，我们尝试将出现的所有正整数均随机映射到 $[1, k]$，则易知此时的最优解必然不可能优于原问题的最优解，并且最坏情况下恰好求得原问题最优解的概率为 $\frac{k!}{k^k}$。于是，我们不妨多次随机映射并求得新问题的最优解，然后取其中的最优解作为原问题的最优解即可。在具体实现时，由于 $(1 - \frac{k!}{k^k})^{1000} < 1\%$，因此随机次数只需达到 10^3 级别即可。

时间复杂度：$\Theta(CountRandom \cdot 3^k nm)$

空间复杂度：$\Theta(2^k nm)$

19 COOLNUM

题目大意

给定 t 组询问 n_i，分别求不超过 n_i 的最大 *cool* 数和超过 n_i 的最小 *cool* 数。定义十进制整数 x 是 *cool* 数当且仅当，存在 k（$1 \le k \le 3$）个不同的十进制数位 $1 \le p_1 < \cdots < p_k \le |x|$，使得 x 整除 $(s-d)^d$，其中 s 表示 x 所有数位上的数字和，d 表示选择的 k 个数位上的数字和。

时间限制：8 s

数据规模

- $1 \le t \le 10^5$，$1 \le n_i \le 10^{1000}$，$\sum |n_i| \le 4 \times 10^6$。

算法分析

显然，若 x 在十进制表示下非零数位不超过 3 位，则 x 必然是 *cool* 数。我们不妨将此类 *cool* 数称为平凡的，而其余 *cool* 数则称为非平凡的。接下来，我们对这两种情形分别进行讨论。

平凡 *cool* 数

此时，若 x 的非零数位少于 3 位，则易知 x 和 $x+1$ 必然都是平凡 *cool* 数。否则，记 x 从最高位起的第 3 个非零位为 i，则不难证明 $\lfloor \frac{x}{10^i} \rfloor \cdot 10^i$ 和 $(\lfloor \frac{x}{10^i} \rfloor + 1) \cdot 10^i$ 必然是最接近 x 的两个平凡 *cool* 数。

非平凡 *cool* 数

对于非平凡 *cool* 数 x，记 x 在十进制表示下共有 t 位，则有 $s \le 9t$ 以及 $d \le 27$。同时，由 $x|(s-d)^d$ 可知 $10^{t-1} \le x \le (s-d)^d$，即 $10^{t-1} \le (9t)^{27}$，因此可得 $t \le 77$。于是，上述限制表明非平凡 *cool* 数至多不超过 10^{77}，则其各数位和 s 不超过 693。同时，由于 $x|(s-d)^d$，因此 x 必然是 s'^{27}（$s' \le 693$）的因子。此时，我们不妨首先通过暴力枚举该范围内可能的 x 预处理出所有非平凡 *cool* 数，实践表明其总数不足 4×10^4，因此每次询问时只需进行二分即可。

时间复杂度：$\Theta(\sum |n_i| \log CountNonTrivial)$

空间复杂度：$\Theta(CountNonTrivial \cdot |n|)$

20 COT5

题目大意

给定一棵初始时为空的大根 $Treap$，要求处理 n 个下列操作：

- $0\ k\ w$：插入一个关键字为 k 且权重为 w 的新节点，保证任意时刻不存在关键字或权重相同的节点；
- $1\ k$：删除当前树中关键字为 k 的节点；
- $2\ k_u\ k_v$：询问当前树中关键字分别为 k_u 和 k_v 的节点间的距离。

时间限制：1 s

数据规模

- $1 \leq n \leq 2 \times 10^5$。

算法分析

由于在执行操作的过程中，$Treap$ 的形态在不断变化，因此考虑将查询两点间距离分解为查询两点的最近公共祖先和查询给定点的深度两种询问。接下来，我们依次解决这两种类型的询问操作。

查询两点的最近公共祖先

易知 $Treap$ 是一棵权重满足堆性质的二叉搜索树，因此所有节点的关键字在先序遍历下有序。于是不难推知，节点 x 和 y 的最近公共祖先即为先序遍历中 $[x, y]$ 内具有最大权重的节点。同时，由于本题并未强制使用在线算法，因此该部分可以将关键字离散后借助线段树来维护，而不存在的关键字只需赋予权重 $-\infty$ 即可。

查询给定点的深度

由于查询给定点 x 的深度等价于查询有多少个点 y 是 x 的祖先，因此接下来我们考虑对此类点进行计数。此时，可以证明点 y 是点 x 的祖先，当且仅当点 y 的权重是先序遍历中 $[x, y]$ 内的最大值，证明如下：

- 首先考虑对于固定的先序遍历还原 $Treap$ 的形态，则每次必然将选择其中权重最大的节点作为树根，并将其两侧的其他所有点分别划入左右子树递归处理；
- 于是，假设先序遍历中 $[x, y]$ 内具有最大权重的节点为 z，若 $z \neq y$，则当 z 成为某棵子树根时，x 和 y 必将被分入不同的子树；
- 否则，若 $z = y$，则 y 将是 $[x, y]$ 内最先成为某棵子树根的节点，因此 y 必然是 x 的祖先。

利用上述结论，则在先序遍历中，点 x 所有祖先的权重在 x 左右两侧分别形成一个严格递增序列。此时，不妨以位于 x 右侧的所有祖先为例，我们考虑在线段树中增加维护一

个值 max_r，表示在该节点所表示区间内严格递增序列的长度，并且保证此序列必然以该区间的首元素为起始。于是，借助 max_r 和之前维护的区间最大值，则单次修改或者询问操作的代价均为 $\Theta(\log^2 n)$。同时，对于所有位于 x 左侧的祖先而言，显然只需类似的再增加维护值 max_l 即可。

时间复杂度：$\Theta(n \log^2 n)$

空间复杂度：$\Theta(n)$

21 COUNTARI

题目大意

给定一个长度为 n 的正整数序列 $\{a_i\}$，要求统计等差三元组的总数。其中，定义等差三元组为满足 $a_j - a_i = a_k - a_j$ 的任意三元组 (i, j, k)（$1 \le i < j < k \le n$）。

时间限制：3 s

数据规模

- $3 \le n \le 10^5$，$1 \le a_i \le 3 \times 10^4$。

算法分析

由于本题要求等差三元组的下标 i、j 和 k 必须递增，因此我们尝试以 l 为块长对序列 $\{a_i\}$ 进行划分。此时，若我们考虑强制令 j 位于当前块中，则可以分为以下三种情形分别解决：

- 若 i 和 k 均位于当前块中，则此种情形下我们暴力枚举 j 和 k，而相应的 i 的可行方案数则只需在顺序扫描时维护计数器即可；
- 若 i 和 k 中恰有一个位于当前块中，则此时不妨暴力枚举 j 和另一个当前块内的元素，于是只需在预处理时求得以块为单位的前（后）缀计数器即可进行统计；
- 若 i 和 k 均位于当前块之外，则此时总贡献为 $\sum_{x+y=2a_j} pre_x \cdot suf_y$，其中 pre 和 suf 分别为以块为单位的前缀计数器和后缀计数器。此时，不难发现上式呈现卷积的形式，则我们只需对每一块分别进行一次 FFT 即可。

对于上述的前两类情形，易知其单块的代价为 $\Theta(l^2)$，而对于上述第三类情形，单次 FFT 的代价为 $\Theta(a \log a)$。于是，易知当取 $l = \sqrt{a \log a}$ 时总代价最低。

时间复杂度：$\Theta(n\sqrt{a \log a})$

空间复杂度：$\Theta(n + a)$

22 CUCUMBER

题目大意

给定 b 个 $n \times n$ 的矩阵 Q_i，对任意无序对 (u, v)，定义矩阵 $C_{u,v} = Q_u Q_v^{\mathrm{T}}$。对于任意矩阵 $C_{u,v}$ 而言，定义 $1 \sim n$ 的排列 p 满足 $C_{u,v}$ 的要求，当且仅当序列 $\{C_{u,v}(i, p_i)\}$ 中存在至少一个奇数。同时，定义矩阵 $C_{u,v}$ 满足要求，当且仅当恰好存在奇数个排列 p 满足 $C_{u,v}$ 的要求。要求统计满足要求的矩阵 $C_{u,v}$ 总数。

时间限制：1.301 s

数据规模

- $1 \le t \le 4 \times 10^3$，$1 \le n \le 60$，$2 \le b \le 8 \times 10^3$，$1 \le Q_{i,j,k} \le 100$，$\sum b_i \le 8 \times 10^3$。

算法分析

首先易知，由于本题中只关注矩阵中元素的奇偶性，因此所有矩阵运算都可在模 2 意义下进行。当 $n = 1$ 时，$C_{u,v}$ 仅存在一个元素 $C_{u,v}(1,1) \equiv Q_u(1,1)Q_v(1,1)$，因此 $C_{u,v}$ 满足要求当且仅当 $Q_u(1,1) \equiv Q_v(1,1) \equiv 1$，于是所求答案即为 $\binom{k}{2}$，其中 k 为满足 $Q_i(1,1) \equiv 1$ 的矩阵 Q_i 总数。

接下来，我们讨论 $n \ge 2$ 的情形。为方便表述，记 $cnt_{u,v}$ 为满足 $C_{u,v}$ 要求的排列 p 总数，$cnt'_{u,v}$ 为不满足 $C_{u,v}$ 要求的排列 p 总数，则由于此时 $n!$ 必为偶数，因此 $cnt_{u,v} \equiv n! - cnt'_{u,v} \equiv cnt'_{u,v}$，于是满足要求的矩阵 $C_{u,v}$ 总数等价于 $\sum_{(u,v)}(cnt'_{u,v} \bmod 2)$。此时，若定义新矩阵 $D_{u,v}(i,j) \equiv C_{u,v}(i,j) + 1$，则不难证明 $cnt'_{u,v} \equiv perm(D_{u,v})$，其中 $perm(D_{u,v})$ 表示矩阵 $D_{u,v}$ 的积和式。同时，由于模 2 的特殊性质，矩阵 $D_{u,v}$ 的积和式 $perm(D_{u,v})$ 在模 2 意义下同余其行列式 $det(D_{u,v})$。

为进一步加速算法，我们对所有矩阵 Q_i 均在其右侧拼上一列 $\vec{x} = (1, 1, \ldots, 1)^{\mathrm{T}}$ 构成 $n \times (n+1)$ 的新矩阵 P_i，则易知 $D_{u,v} \equiv P_u P_v^{\mathrm{T}}$。根据*Cauchy–Binet* 公式可知，

$$det(D_{u,v}) \equiv det(P_u P_v^{\mathrm{T}}) \equiv \sum_{i=1}^{n+1} det(P_{u,i}) det(P_{v,i}^{\mathrm{T}}) \equiv \sum_{i=1}^{n+1} det(P_{u,i}) det(P_{v,i})$$

其中，$P_{u,i}$ 表示 P_u 去除第 i 列后的矩阵。因此，若已经求得所有 $det(P_{u,i})$，则利用位运算即可在 $\Theta(b^2)$ 的代价内计算出答案。此时，若利用*Gauss–Jordan* 消元法将 P_u 化为行最简形，则可以推知：

- 若 P_u 列秩小于 n，易知任意 $P_{u,i}$ 均为奇异矩阵，即任意 $det(P_{u,i})$ 均为 0；
- 若 P_u 列秩等于 n，则 P_u 等价于在单位矩阵 I 中插入一列 $\vec{x_k}$。此时，易证 $det(P_{u,i}) = 1$ 当且仅当，$i = k$ 或者 $\vec{x_k}(i) = P_u(i,k) = 1$。

于是，若同样利用位运算加速，则单次消元过程代价降为 $\Theta(n^2)$。

时间复杂度：$\Theta(bn^2 + b^2)$

空间复杂度：$\Theta(bn)$

23 CUSTPRIM

题目大意

给定三元组 (a,b,c)，其中 $c=11$ 或 $c=24$。定义三元组 (a_1,b_1,c_1) 和 (a_2,b_2,c_2) 的乘法法则如下：

- $s=(a_1a_2+b_1b_2+c_1c_2)+(a_1b_2+b_1a_2)+(c_1+c_2)$；
- $t=\lfloor\frac{s}{2}\rfloor+16(c_1+c_2)-c_1c_2$；
- $A=t-2(a_1b_2+b_1a_2)-(a_1c_2+c_1a_2)+33(a_1+a_2)+(b_1b_2-a_1a_2)$；
- $B=t-5(a_1b_2+b_1a_2)-(c_1b_2+b_1c_2)+33(b_1+b_2)+(2b_1b_2+4a_1a_2)$；
- 若 s 为偶数，则乘法运算结果为 $(A-540,B-540,24)$；
- 若 s 为奇数，则乘法运算结果为 $(A-533,B-533,11)$。

定义零元 z 是对任意三元组 x 满足 $z*x=z$ 的三元组。定义单位元 u 是对任意三元组 x 满足 $u*x=x$ 的三元组。定义一个三元组为素数，当且仅当其不能被分解为两个非零元、非单位元的三元组的乘积。要求判定给定三元组是否为素数。

时间限制：5 s

数据规模

- $1\le t\le 10^4$，$-10^7\le a,b\le 10^7$，$c=11,24$。

算法分析

本题是一道极具难度的抽象代数题，证明过程非常复杂，其大致证明思路如下：

- 设 S 是所有合法三元组构成的集合，则可以证明存在从 S 到欧几里得域 $Z[\omega]$ 的同构 $\phi(a,b,c)=(33-2a-c)+(b-a)\omega$，其中 ω 是满足方程 $\omega^2=\omega-3$ 的一个解 $\omega=\frac{1+\sqrt{-11}}{2}$。因此，判定三元组 (a,b,c) 是否为素数的问题可以转化为判定欧几里得域 $Z[\omega]$ 中 $a+b\omega$ 是否为素数的问题。
- 经过一系列证明可得，对于欧几里得域 $Z[\omega]$ 中的任意元素 $a+b\omega$，若其不是整数，则其为素数当且仅当其距离函数 $f(a+b\omega)=a^2+ab+3b^2$ 为素数。否则，$a+b\omega$ 为素数当且仅当，$|a|$ 为素数，且 $|a|=2$ 或者 -11 是模 $|a|$ 意义下的二次非剩余，其中 $|a|\neq 11$。

借助上述结论，则我们只需利用 $Miller-Rabin$ 等素性检测算法即可。

时间复杂度：$\Theta(PrimalityTest)$

空间复杂度：$\Theta(1)$

24 DAGCH

题目大意

给定一张 n 个点 m 条边的有向图，其中所有节点的标号是以 1 号点为根进行 Dfs 时的遍历顺序。定义节点 x 对节点 y 是重要的，当且仅当 $x < y$ 且存在一条路经 $v_0 = x, v_1, \ldots, v_k = y$，满足 $v_i > y$ 对任意 $0 < i < k$ 成立。定义节点 x 对节点 y 是最重要的，当且仅当 x 是对 y 重要的所有点中标号最小的点。要求处理 q 次询问，每次统计点 p_i 对多少个点是最重要的。

时间限制：2 s

数据规模

- $1 \leq t \leq 10$，$2 \leq n \leq 10^5$，$n - 1 \leq m \leq 2 \times 10^5$，$1 \leq q \leq 10^5$。

算法分析

由于所有点的标号是以 1 号点为根进行 Dfs 时的遍历顺序，则 Dfs 树上点 x 的父节点必然对 x 是重要的，因此限制条件 $x < y$ 的存在与否并不影响最重要节点的判定。于是，通过简单的分析不难发现，所谓最重要的节点其实就是半必经点。因此，本题我们只需利用 $Lengauer - Tarjan$ 算法即可解决。

时间复杂度：$\Theta((n + m) \log n + q)$

空间复杂度：$\Theta(n + m)$

25 *DELNMS

题目大意

给定由 n 个整数构成的序列 $\{a_i\}$，要求通过尽可能少的合法操作将整个序列恰好完全删除。其中，一次合法的操作 (v,t) 定义为：若记 k 为满足 $v+kt \le n$ 的最大非负整数，则必须满足 $a_v = a_{v+t} = a_{v+2t} = \cdots = a_{v+kt}$。每当执行一次合法操作 (v,t) 后，则 $a_v, a_{v+t}, \ldots, a_{v+kt}$ 均被删除，并且由其右侧剩余的元素通过平移填补出现的空位。最终评分以删除操作的次数作为参考。

时间限制：1 s

数据规模

- $1 \le n \le 10^5$，$1 \le a_i \le 10^5$。

算法分析

首先，本题一种显然可行的策略为倒序将所有元素依次删除。在此基础之上，为方便处理，我们依然考虑倒序执行删除操作。此时，若当前序列末端的元素 a_x 仅出现一次，则显然只能对其单独执行一次删除操作。否则，若记元素 a_x 前一次出现为 a_y，则我们尝试将 a_x 和 a_y 同时删除，此时必有该次删除操作的公差 $t = x - y$。因此，我们考虑不断找到元素 a_x 前一次出现的位置，直到其下标序列不再满足公差为止，则仅需一次删除操作即可将这些元素全部删除。其中，由于操作过程中元素对应下标可能发生改变，则我们可以借助树状数组来维护。

上述算法在 $Codechef$ 上可得 $0.546pt$。

时间复杂度：$\Theta(n \log n)$

空间复杂度：$\Theta(n)$

26 DEVLOCK

题目大意

求有多少个允许有前导 0 的 n 位十进制整数 x，满足：

- x 能被 p 整除；
- x 十进制表示的各个数位之和不超过 m。

要求对任意 $m = i$ 分别求出相应方案数模 998244353 的结果，其中 i 满足 $0 \le i \le m$。

时间限制：$5 \sim 15$ s

数据规模

- $1 \le n \le 10^9$，$1 \le p \le 16$，$1 \le m \le 1.5 \times 10^4$。

算法分析

由于题中 p 的范围远小于 n 的范围，因此不难想到将十进制位 z 按照 $r_z = 10^z \bmod p$ 分类，则由其循环节可以快速获知每类十进制位的个数 cnt_i。此时，若记 $f_{i,j,k}$ 为所有第 i 类十进制位上的数模 p 余 j 且总和为 k 的方案数，则所有数位的总方案数即为：

$$Ans_{j,k} = \sum_{\substack{j_0+j_1+\cdots+j_{p-1}\equiv j \\ k_0+k_1+\cdots+k_{p-1}=k}} \prod_i f_{i,j_i,k_i}$$

对于上式而言，由于 p 较小而数字总和所在维呈现卷积的形式，因此我们可以分 p 次合并答案，则每次只需执行 $\Theta(p)$ 次 NTT 即可，这部分总代价为 $\Theta(p^2 m \log m + p^3 m)$。

为计算 $f_{i,j,k}$ 的取值，易知对于固定的 i 而言，任意 k 只有当对应于 $j = ik \bmod p$ 时才有解，因此可以简化为求所有 $f'_{i,k}$。同时，不难发现 $f'_{i,k}$ 可以等价为关于数字和的生成函数 $G_i(x) = (\sum_{i=0}^{9} x^i)^{cnt_i}$ 中 x^k 项前的系数，此部分同样可以利用倍增 NTT 来实现。

时间复杂度：$\Theta(pm \log m \log n + p^2 m \log m + p^3 m)$

空间复杂度：$\Theta(pm)$

27 DGCD

题目大意

给定一棵 n 个节点的无根树，以及初始时每个节点的权值 v_i。要求在线处理 q 次操作，其中修改操作为对某条路径上所有点权均加上同一个值，询问操作则为询问某条路径上所有点权的最大公因数。

时间限制：2 s

数据规模

- $1 \leq n \leq 5 \times 10^4$，$1 \leq q \leq 5 \times 10^4$，$1 \leq v_i \leq 10^4$。

算法分析

首先考虑本题在链上的简化情形，此时需要支持区间加法以及查询区间最大公因数，则受到辗转相除法的启发，不难推知：

$$\begin{aligned} gcd(a_1, a_2, \ldots, a_k) &= gcd(gcd(a_1, a_2, \ldots, a_{k-1}), gcd(a_{k-1}, a_k)) \\ &= gcd(gcd(a_1, a_2, \ldots, a_{k-1}), gcd(a_{k-1}, a_k - a_{k-1})) \\ &= gcd(a_1, a_2, \ldots, a_{k-1}, a_k - a_{k-1}) \\ &= gcd(a_1, a_2 - a_1, a_3 - a_2, \ldots, a_k - a_{k-1}) \end{aligned}$$

因此，我们考虑对序列进行差分，则易知可以借助线段树进行维护，每个节点上只需记录其所表示区间的首尾元素，以及所有相邻元素之差的最大公因数即可。同时，区间加法可以利用懒标记来完成，下传标记时则只需更新首尾元素即可。于是，对于本题而言，我们只需利用树链剖分即可将树上问题转化为链上的子问题。

时间复杂度：$\Theta(m \log^2 n)$

空间复杂度：$\Theta(n)$

28 DIFTRIP

题目大意

给定一棵 n 个点的有根树，要求统计不相似的可行路径条数。其中，定义一条可行路径为从树上某个点 x 到其某个祖先 y 的路径。同时，定义两条路径是相似的，当且仅当其长度相等，并且路径上相同次序的两个点度数均对应相等。

时间限制：1 s

数据规模

- $1 \leq n \leq 10^5$。

算法分析

不难发现，若将每个点的度数视作为该点的字符，则问题等价于给定一棵字典树，统计其中不同子串的总数。于是，我们显然可以借助后缀自动机来解决本题。在具体实现时，由于本题中字符集大小可能达到 $\Theta(n)$ 级别，因此我们不妨在后缀自动机的每个节点上利用平衡树仅保留存在的所有转移边。

时间复杂度：$\Theta(n \log n)$

空间复杂度：$\Theta(n)$

29 DIVIDEN

题目大意

给定平面直角坐标系中的一个 n 度角，要求通过尺规作图将其 n 等分。

时间限制：1 s

数据规模

- $0 < n < 360$。

算法分析

由于要求 n 等分给定的 n 度角，则问题等价于通过尺规作图构造出 1° 角。此时，若不考虑给定的 n 度角，我们不难想到利用等边三角形构造出 30° 角。同时，由于黄金分割比 $\phi = \frac{\sqrt{5}-1}{2}$ 可以仅借助尺规作出，因此只需利用黄金三角形便能构造出 36° 角。于是，通过对两角之差 6° 作角平分线即可构造出 3° 角。此时，可以证明当且仅当 $3|n$ 时无解，否则我们显然可以通过迭代构造出 1° 角。

时间复杂度：$\Theta(n)$

空间复杂度：$\Theta(1)$

30 DIVISORS

题目大意

给定正整数 b 和 x，统计满足条件的正整数 n 的个数。其中，定义 n 满足要求当且仅当，存在正整数 $n < d \leq b$，使得 $d|nx$。

时间限制：7 s

数据规模

- $1 \leq t \leq 40$，$1 \leq b \leq 10^{12}$，$1 \leq x \leq 60$。

算法分析

首先不难发现，当 $x = 1$ 时显然无解，于是接下来我们仅考虑 $x > 1$ 的情形。由于本题中 d 的范围较大，因此我们不妨考虑 $k = \frac{nx}{d}$（$1 \leq k < x$），则所有约束条件等价为 $k|nx$ 以及 $k \geq \frac{nx}{b}$。于是，若记 S_k 为满足以下条件的所有 n 构成的集合：

- $k|nx$ 并且 $k \geq \frac{nx}{b}$；
- 不存在 $k < k' < x$，使得 $k'|nx$。

则易知 $Ans = \sum_{i=1}^{x-1} |S_i|$。由于 x 的范围较小，因此我们尝试枚举 k 并分别统计 $|S_k|$。此时，若记 $c_k = \frac{k}{gcd(k,x)}$，则 n 必须满足 $c_k|n$ 以及 $n \leq \frac{kb}{x}$。因此，若令 $m = \frac{n}{c_k}$，则有 $m \leq \frac{kb}{c_k x}$。于是，由于 n 和 m 一一对应，针对 n 的计数可以完全等价于针对 m 的计数。此时，上述第二个约束条件便可以转化为 $\frac{k'}{gcd(k',c_k x)}|m$，则由于 k、x 和 b 均为已知，于是只需利用容斥原理即可解决。

在具体实现时，由于存在较多重复状态以及冗余状态，因此我们每次在已有状态的基础之上新加入一条性质 $P_{k'}$，并对产生的所有相同状态进行合并。除此之外，由于本题时间限制较紧，需要一定的常数优化方可解决本题。

时间复杂度：$\Theta(x^2 \cdot CountState)$

空间复杂度：$\Theta(\tau(x) \cdot CountState)$

31 DOMNOCUT

题目大意

给定一张 $n \times m$ 的网格，定义合法的多米诺覆盖染色方案为使得每个网格有且仅有一个相邻网格与其同色的染色方案。同时，定义多米诺覆盖染色方案的割为一条水平或者竖直的直线，使得整个网格被分成的两部分均为合法的多米诺覆盖染色方案，即该条直线不得将同色的相邻网格分割开。要求给出一种合法的多米诺覆盖染色方案，并在最小化割的总数的前提下同时最小化所用颜色的总数。

时间限制：4 s

数据规模

- $1 \le t \le 3 \times 10^3$，$1 \le n, m \le 500$，$\sum n_i m_i \le 2 \times 10^6$。

算法分析

本题是一道较为繁琐的构造题，首先易知当且仅当 n 和 m 均为奇数时无解，因此接下来我们默认仅有 n 可能为奇数，并且当 n 为偶数时假定 $n \le m$。此时，若记 $f_{n,m}$ 为 $n \times m$ 的多米诺覆盖染色方案中割数的下限，则可以推知：

$$f_{n,m} = \begin{cases} \max\{m+n-2-\frac{nm}{4}, 0\} & n \text{ 为偶数} \\ \max\{\frac{m}{2}+n-2-\frac{(n-1)m}{4}, 0\} & n \text{ 为奇数} \end{cases}$$

证明如下：

- 由于 nm 为偶数，因此任意一条可能的割线 l 分成的两部分网格数必然奇偶性相同。此时，若两部分网格数均为奇数，则称 l 为奇割线，反之则为偶割线；
- 易知任意一块 1×2 的同色多米诺骨牌必然恰好被一条割线经过，并且任意一条奇割线至少经过 1 块骨牌，而任意一条非合法割的偶割线至少经过 2 块骨牌，于是有 $\frac{nm}{2} \ge cnt_{odd} + 2(cnt_{even} - f_{n,m})$，即 $f_{n,m} \ge \frac{cnt_{odd}}{2} + cnt_{even} - \frac{nm}{4}$；
- 此时，若 n 为偶数，则有 $cnt_{odd} = 0$ 以及 $cnt_{even} = n + m - 2$；若 n 为奇数，则有 $cnt_{odd} = \frac{m}{2}$ 以及 $cnt_{even} = n + \frac{m}{2} - 2$。于是，只需分别将 cnt_{odd} 和 cnt_{even} 代入上式计算并化简即可。

利用上述结论，我们尝试分情况讨论并构造出达到上述下界的合法方案。

$n = 1$ 且 $m = 2k$（$k \ge 1$）

由于 $f_{n,m} = k - 1$，并且最小颜色数为 1 当且仅当 $m = 2$，否则最小颜色数为 2，因此一种可行的最优解形如：

$$aabbaa\cdots$$

$n = 2$ 且 $m = 2k$（$k \ge 1$）

此时 $f_{n,m} = k$，并且最小颜色数显然为 2，于是一种可行的最优解形如：

$$aabbaa\cdots$$
$$bbaabb\cdots$$

$n = 2$ **且** $m = 2k + 1$ **(** $k \geq 1$ **)**

此时 $f_{n,m} = k$，并且最小颜色数必为 3，于是一种可行的最优解形如：

$$caabbaa\cdots$$
$$cbbaabb\cdots$$

$n = 3$ **且** $m = 4$

此种情形下直接暴力枚举即可，则有 $f_{n,m} = 1$ 以及最小颜色数为 3，一种可行的最优解为：

$$ccab$$
$$abab$$
$$abcc$$

$n = 3$ **且** $m = 2k$ **(** $k \geq 3$ **)**

此时 $f_{n,m} = 1$，并且最小颜色数为 3，于是一种可行的最优解形如：

$$aabbccaabb\cdots$$
$$bccaabbcc\cdots x$$
$$baabbccaa\cdots x$$

其中，为使得 x 存在可行的颜色，则由于上述构造方式的特殊性，我们必然可以通过改变第三行最后一块横放骨牌的颜色来解决。

$n = 4$ **且** $m = 2k + 1$ **(** $k \geq 2$ **)**

此时 $f_{n,m} = 1$，并且最小颜色数为 3，于是一种可行的最优解形如：

$$cbbccaabbcc\cdots$$
$$caabbccaabb\cdots$$
$$bbccaabbcc\cdots x$$
$$aabbccaabb\cdots x$$

其中，由于前两行与后两行的所有横放骨牌采用一致的染色顺序，因此 x 必然存在可行的颜色。

$n = 4$ **且** $m = 4$

此种情形下直接暴力枚举即可，则有 $f_{n,m} = 2$ 以及最小颜色数为 3，一种可行的最优解为：

$$aaba$$
$$ccba$$
$$abcc$$
$$abaa$$

$n = 4$ **且** $m = 2k$ **(** $k \geq 3$ **)**

此时 $f_{n,m} = 2$，并且最小颜色数为 3，于是一种可行的最优解形如：

$$aabbccaabb\cdots$$
$$bbccaabbcc\cdots$$
$$caabbccaa\cdots x$$
$$cbbccaabb\cdots x$$

其中，类似于 $n = 3$ 且 $m = 2k$（$k \geq 3$）的情形，我们同样可以通过改变第四行最后一块横放骨牌的颜色以使得 x 存在可行的颜色。

$n = 6$ **且** $m = 6$

此种情形下直接暴力枚举即可，则有 $f_{n,m} = 1$ 以及最小颜色数为 3，一种可行的最优解为：

$$aacaac$$
$$bbcbbc$$
$$caacaa$$
$$cbbcbc$$
$$acaabc$$
$$acbbaa$$

$n = 2k' + 1$ **且** $m = 2k$ **(** $k' \geq 2$ **,** $k \geq 3$ **)**

在此种情形下，首先可知 $f_{n,m} = 0$ 以及最小颜色数为 3。此时，我们首先考虑 $n = 5$ 且 $m = 6$ 的情形，则一种可行的最优解为：

$$aacaac$$
$$cbcbbc$$
$$cbaacb$$
$$bccbcb$$
$$baabaa$$

于是，接下来我们尝试在此基础之上每次扩展两行，则我们以右下角的横放骨牌 a 为突破口可得：

$$aacaac$$
$$cbcbbc$$
$$cbaacb$$
$$bccbcb$$
$$baabac$$
$$acbcac$$
$$acbcbb$$

不难发现，上述方法在扩展行时总是可行的。同时，为对列进行扩展，则我们只需类似的以右上角的竖放骨牌 c 为突破口即可。

$n = 2k'$ **且** $m = 2k$ **(** $k' \geq 3$ **,** $k \geq 4$ **)**

在此种情形下，首先可知 $f_{n,m} = 0$ 以及最小颜色数为 3。此时，我们首先考虑 $n = 6$ 且 $m = 8$ 的情形，则一种可行的最优解为：

$$aaccbccb$$
$$bcaabaab$$
$$bcbccbcc$$
$$aabaabab$$
$$bccbccab$$
$$baabaacc$$

于是，不难发现此时依然只需利用类似于上种情形的方法进行扩展即可。

利用上述构造方式，则每种情形下给出的方案均已达到理论下界。

时间复杂度：$\Theta(nm)$

空间复杂度：$\Theta(nm)$

32 EASYEX

题目大意

给定一个共有 k 面的均匀骰子，并且相互独立地随机抛掷 n 次该骰子。定义 a_i 为抛掷到第 i 面的次数，求 $\prod_{i=1}^{l} a_i^f$ 的期望。

时间限制：$5 \sim 20$ s

数据规模

- $1 \le t \le 50$，$0 < n, k \le 10^9$，$0 < f \le 10^3$，$0 < lf \le 5 \times 10^4$。

算法分析

由于本题涉及到期望，因此我们考虑 ln 个 0/1 随机变量 $x_{i,j}$，其中 $x_{i,j} = 1$ 当且仅当第 j 次恰好抛掷到第 i 面，则有 $a_i = \sum_{j=1}^{n} x_{i,j}$，即 $\prod_{i=1}^{l} a_i^f = \prod_{i=1}^{l} (\sum_{j=1}^{n} x_{i,j})^f$。此时，考虑将上式完全展开，则其中每一项均为若干随机变量之积，并且该项值非零当且仅当该项涉及的所有随机变量值均为 1。于是，对于展开式中的任意一项而言，若同时存在 a, b 和 c，使得 $x_{a,c}$ 和 $x_{b,c}$ 均被涉及，则该项的值必然恒为 0，因此在接下来的讨论中我们均忽略所有此类项。与此同时，由于所有随机变量取值均为 0 或者 1，因此随机变量 $x_{i,j}$ 的任意正整数次幂均恒等于 $x_{i,j}$ 的取值。于是，对于展开式中任意涉及恰好 t 个不同随机变量的一项而言，该项值为 1 的概率显然为 k^{-t}。此时，若记全集 $U = \{i \in \mathbb{N}_+ | i \le n\}$，则可以推知：

$$
\begin{aligned}
E(\prod_{i=1}^{l} a_i^f) &= E(\prod_{i=1}^{l} (\sum_{j=1}^{n} x_{i,j})^f) \\
&= E(\sum_{S \subseteq U} \sum_{\substack{S_u \cap S_v = \varnothing \\ S_1 \cup S_2 \cup \cdots \cup S_l = S}} \prod_{i=1}^{l} |S_i|! dp_{f,S_i} \prod_{j \in S_i} x_{i,j})
\end{aligned}
$$

其中，S_i 表示在整个展开式的某一项中，所有随机变量 $x_{i,j}$ 涉及到的所有 j 构成的集合，dp_{f,S_i} 则表示项 $\prod_{j \in S_i} x_{i,j}$ 在 $(\sum_{j=1}^{n} x_{i,j})^f$ 的展开式中的系数，而 $|S_i|!$ 则表示 S_i 中所有元素的任意一种出现顺序均是等价的。此时，通过观察不难发现，其实上式仅与 $|S_i|$ 有关，因此我们令随机变量 $y_i = |S_i|$，则由期望的线性性可以进一步化简得：

$$
\begin{aligned}
E(\prod_{i=1}^{l} a_i^f) &= E(\sum_{y=l}^{\min\{lf,n\}} \binom{n}{y} \sum_{\substack{1 \le y_i \le f \\ y_1 + y_2 + \cdots + y_l = y}} \prod_{i=1}^{l} \binom{y - y_1 - \cdots - y_{i-1}}{y_i} y_i! dp_{f,y_i} x^{y_i}) \\
&= E(\sum_{y=l}^{\min\{lf,n\}} \binom{n}{y} \sum_{\substack{1 \le y_i \le f \\ y_1 + y_2 + \cdots + y_l = y}} y! \prod_{i=1}^{l} dp_{f,y_i} x^{y_i}) \\
&= \sum_{y=l}^{\min\{lf,n\}} \binom{n}{y} y! E(\sum_{\substack{1 \le y_i \le f \\ y_1 + y_2 + \cdots + y_l = y}} \prod_{i=1}^{l} dp_{f,y_i} x^{y_i}) \\
&= \sum_{y=l}^{\min\{lf,n\}} \binom{n}{y} y! k^{-y} [x^y] (\sum_{i=1}^{f} dp_{f,i} x^i)^l
\end{aligned}
$$

于是，若我们已经求得所有 $dp_{f,i}$，则 $(\sum_{i=1}^{f} dp_{f,i}x^i)^l$ 只需利用倍增 FFT 即可求得，因此最后计算答案的代价为 $\Theta(lf)$。而对于 $dp_{f,i}$ 而言，则不难推得以下递推式：

$$dp_{f,i} = dp_{f-1,i-1} + i \cdot dp_{f-1,i}$$

因此，所有 $dp_{f,i}$ 均可在预处理时通过 $\Theta(f^2)$ 递推求得。

时间复杂度：$\Theta(f^2 + lf \log lf \log l)$

空间复杂度：$\Theta(f^2 + lf)$

33 EST

题目大意

给定一个长度为 n 的字符串 s，记由 s 的所有后缀构成的字典树为 T_s。求有多少个不同的长度为 n 的字符串 s'，使得 T_s 和 $T_{s'}$ 同构。其中，判定两棵字典树是否同构时，忽略所有转移边上的字符，即只考虑字典树的形态。

时间限制：5 s

数据规模

- $1 \leq t \leq 10$，$1 \leq n \leq 10^5$。

算法分析

首先不难发现，T_s 中所有叶节点的深度必然是一段连续整数 $n, n-1, \ldots, l+1$，则易知存在某个 i 使得 $lcp(s_{i,n}, s_{n-l+1,n}) = l$，其中 lcp 表示两个字符串的最长公共前缀。同时，由于 T_s 中深度为 i（$l+1 \leq i \leq n$）的叶子节点有且仅由一个，因此前 $n-l$ 个字符之间的相等关系均需保持不变。于是可以推知，若固定 s' 的前 $n-l$ 个字符保持不变的方案数为 Ans，则总方案数为 $Ans \prod_{i=1}^{cnt}(26-i+1)$，其中 cnt 表示 s 中出现的字符种类数。

接下来，不妨假定 $s'_{1,n-l} = s_{1,n-l}$，则为使得 $T_{s'}$ 和 T_s 同构，等价于对任意 $1 \leq i < n-l$ 均要求满足 $lcp(s'_{i,n}, s'_{n-l,n}) = lcp(s_{i,n}, s_{n-l,n})$。同时，由于 $s'_{n-l+1,n}$ 必然是某个 $s'_{i,n}$ 的前缀，于是此时最多只有 $n-l$ 种不同方案。因此，我们只需利用哈希并将可行方案去重即可。

时间复杂度：$\Theta(n \log n)$

空间复杂度：$\Theta(n)$

34 EVILBOOK

题目大意

给定 n 个相互独立的任务 (c_i, m_i)，其中 c_i 为完成该任务所需付出的能力值，m_i 为完成该任务所得的魔法值。已知初始时魔法值为 0，要求通过执行若干次如下两类操作收集到至少 666 点魔法值：

- 若付出 c_i 点能力值完成第 i 个任务，则可获得 m_i 点魔法值，但每个任务至多只能完成一次；
- 若当前拥有至少 x 点魔法值，则可以选择付出 x 点魔法值使用帮助，并选择某个任务 i，使得 c_i 和 m_i 均被除以 3。

要求最小化初始时所需的总能力值。

时间限制：3 s

数据规模

- $1 \le t \le 5$，$1 \le n \le 10$，$10 \le x < 666$，$0 \le c_i, m_i \le 10^9$。

算法分析

由于本题数据范围较小，因此我们考虑进行搜索。通过一些简单的观察，我们不难发现如下一些性质：

- 由于所有任务相互独立，因此只有在完成某个任务之前才有必要对该任务使用帮助；
- 若对第 i 个任务使用帮助次数为 cnt_i，则为获得收益必须满足 $x \cdot cnt_i < \frac{m_i}{3^{cnt_i}}$；
- 在当前魔法值允许的前提下，若对当前任务多使用一次帮助后仍然可以直接满足要求，则为了最小化对能力值的需求，显然应当多使用一次帮助。

通过实践可以发现，在上述所有约束条件下，可行的 cnt_i 非常少，因此只需再借助可行性剪枝和最优性剪枝即可解决本题。

时间复杂度：$\Theta(CountState \cdot n)$

空间复杂度：$\Theta(n)$

35 FIBTREE

题目大意

给定一棵 n 个点的无根树，初始时所有点权均为 0，要求在线处理 m 个下列操作：

- $A\ x\ y$：对树上 x 到 y 之间路径上的相应点权分别加上 $Fibonacci$ 数列的对应项；
- $QS\ x\ y$：询问以 x 为树根时，以 y 为根的子树中所有点权之和；
- $QC\ x\ y$：询问树上 x 到 y 之间路径上的所有点权之和；
- $R\ x$：将树上所有点权均还原至第 x 次操作后的情形。

时间限制：5 s

数据规模

- $1 \leq n, m \leq 10^5$。

算法分析

由于本题中同时涉及链上求和与子树求和，因此显然考虑利用树链剖分解决。于是，接下来我们主要解决对一段连续区间加上 $Fibonacci$ 数列的问题。为方便处理，我们不妨令 $Fibonacci$ 数列的初值为 $f_0 = f_1 = 1$，则不难推知其通项公式为：

$$f_i = \frac{5+\sqrt{5}}{10}(\frac{1+\sqrt{5}}{2})^i + \frac{5-\sqrt{5}}{10}(\frac{1-\sqrt{5}}{2})^i$$

此时，由于 $383008016^2 \equiv 5 \pmod{10^9+9}$，因此可以推得 $Fibonacci$ 数列在模 10^9+9 意义下的通项公式为：

$$f_i \equiv 138300803 \times 691504013^i + 861699207 \times 308495997^i$$

利用上述通项公式，则每次修改操作都可以被等价分解为两次区间加等比数列的操作。由于所加等比数列的公比只有两种，于是易知只需在线段树中分别维护懒标记来表示首项即可。同时，对于可持久化部分而言，由于修改操作相互独立且易于合并，因此可以通过标记永久化的方式减小空间开销。

时间复杂度：$\Theta(m \log^2 n)$

空间复杂度：$\Theta(m \log^2 n)$

36 FINDSEQ

题目大意

给定一个长度为 n 的整数序列 $\{a_i\}$，以及一个 $1 \sim 5$ 的排列 $(b_0, b_1, b_2, b_3, b_4)$。要求找到任意一个合法的下标序列 $(i_0, i_1, i_2, i_3, i_4)$，满足 $0 \le i_0 < i_1 < i_2 < i_3 < i_4 < n$，并且使得 $(a_{i_0}, a_{i_1}, a_{i_2}, a_{i_3}, a_{i_4})$ 与 $(b_0, b_1, b_2, b_3, b_4)$ 各自序列内部的相对大小关系完全一致。

时间限制：$9 \sim 14$ s

数据规模

- $1 \le t \le 60$，$5 \le n \le 10^3$。

算法分析

首先，由于本题中始终只关心整数相互间的大小关系，因此不妨通过离散化将序列中的整数全部映射到范围 $[1, n]$ 以内。其次，为了使得决策区间互不干扰，我们考虑暴力枚举 i_1 和 i_3，然后将剩余的 $\{b_0, b_2, b_4\}$ 按从小到大的顺序依次记为 b_l、b_m、b_r。为方便接下来的表述，记 $sum_{i,j}$ 为前 i 个整数中不超过 j 的整数个数，记 $premin_{i,j}$ 为 $a_1 \sim a_i$ 中不小于 j 的最小值，记 $premax_{i,j}$ 为 $a_1 \sim a_i$ 中不大于 j 的最大值，类似的分别记 $sufmin_{i,j}$ 和 $sufmax_{i,j}$ 为 $a_i \sim a_n$ 中相应的值。接下来，我们分为以下三种情形进行讨论：

- 情形一：若 $m = 2$，则不失一般性可以假设 $l = 0$ 且 $r = 4$。此时，若记 a_{i_1} 和 a_{i_3} 的取值要求 a_{i_l} 下限为 s 且 a_{i_r} 上限为 t，则易知 a_{i_l} 和 a_{i_r} 必然应当取 $premin_{i_1-1,s}$ 以及 $sufmax_{i_3+1,t}$。为判断是否存在可行的 a_{i_m}，则我们只需利用 sum 值即可 $\Theta(1)$ 判定；
- 情形二：若 $l = 2$，则不失一般性可以假设 $r = 0$ 且 $m = 4$。此时，类似于情形一，我们首先可以参考 a_{i_1} 和 a_{i_3} 选取前缀中尽可能大的值作为 a_{i_r}，则对于固定的 a_{i_r} 而言，a_{i_m} 的取值也应当尽可能大，最后再利用 sum 值来判定是否存在可行的 a_{i_l}；
- 情形三：若 $r = 2$，则不失一般性可以假设 $l = 0$ 且 $m = 4$。此时，类似于情形一，我们首先可以参考 a_{i_1} 和 a_{i_3} 选取前缀中尽可能小的值作为 a_{i_l}，则对于固定的 a_{i_l} 而言，a_{i_m} 的取值也应当尽可能小，最后再利用 sum 值来判定是否存在可行的 a_{i_r}。

通过上述讨论，若我们求得了一组可行的 $(a_{i_0}, a_{i_1}, a_{i_2}, a_{i_3}, a_{i_4})$，则只需再线性扫描一遍即可得到对应的下标序列。

时间限制：$\Theta(n^2)$

空间限制：$\Theta(n^2)$

37 FLYDIST

题目大意

给定一张 n 个点 m 条边的简单无向图，要求对边权进行适当的增减，使得对于任意一条边 (u,v)，满足 u 和 v 之间的最短路径即为该边的边权 $d_{u,v}$。要求最小化总增减量，并以分数形式给出精确值。

时间限制：1 s

数据规模

- $2 \leq n \leq 10$，$1 \leq m \leq 45$，$1 \leq d_{u,v} \leq 20$。

算法分析

要使得 (u,v) 即为 u 和 v 之间的最短路，则 $d_{u,v}$ 应不超过 u 和 v 之间任意路径的长度。不难证明，这等价于对任意包含 (u,v) 的无弦简单环，都满足 $d_{u,v}$ 不超过环长的一半。因此，我们可以对所有约束条件建立线性规划模型，则要求最小化目标函数 $\sum_{(i,j)} |\Delta_{(i,j)}|$。为去除目标函数中的绝对值，通常可以将每个变量均分解为 $\Delta^+_{(i,j)} - \Delta^-_{(i,j)}$。同时，为使得初始解可行，我们不妨利用对偶线性规划来解决。由于简单无向图中无弦简单环的总数上限约为 $\Theta(n^3)$ 级别，因此我们可以直接利用单纯形算法解决本题，具体实现时可以用实数来完成线性规划的整个过程，最后再枚举出最有可能的分母即可。

时间复杂度：$\Theta(Simplex(n^3, m))$

空间复杂度：$\Theta(n^3 m)$

38 FN

题目大意

给定 $Fibonacci$ 数列的初值 $f_0 = 0$ 以及 $f_1 = 1$，在模 p 意义下求 c 在数列中第一次出现的位置。

时间限制：$0.5 \sim 15$ s

数据规模

- $1 \leq t \leq 100$，$11 \leq p \leq 2 \times 10^9$；
- $p \bmod 10$ 为完全平方数。

算法分析

首先易知 $Fibonacci$ 数列通项为：

$$f_i = \frac{1}{\sqrt{5}}(\phi^i - (-\phi)^{-i})$$

其中，$\phi = \frac{1+\sqrt{5}}{2}$。此时，借助二次剩余即可将上式在模 p 意义下进行简化。因此，若令 $x = \phi^i$，则上式即为关于 x 的二次方程，于是只需再次借助二次剩余计算求根公式即可求得 x 的所有可能解。此时，利用 $Baby - step - giant - step$ 的思想即可解得 i 的所有可能解。在具体实现时，二次剩余的求解可以利用Cipolla 算法。

时间复杂度：$\Theta(\sqrt{p})$

空间复杂度：$\Theta(\sqrt{p})$

39 FNCS

题目大意

给定一个长度为 n 的序列 $\{a_i\}$，以及 n 个函数 $f_i = \sum_{j=l_i}^{r_i} a_j$。要求处理 q 个下列操作：

- $1\ x\ y$：将 a_x 的值修改为 y；
- $2\ u\ v$：询问 $\sum_{i=u}^{v} f_i$ 的值。

时间限制：2.5 s

数据规模

- $1 \le n, q \le 10^5$，$1 \le a_i, y \le 10^9$，$1 \le l_i \le r_i \le n$，$1 \le x \le n$，$1 \le u \le v \le n$。

算法分析

本题中由于查询的均为一段连续区间中的函数值之和，因此我们考虑对所有函数以 k 为块长进行分块。此时，若记 $cnt_{i,j}$ 为第 i 块内所有函数覆盖 a_j 的次数，记 sum_i 为当前第 i 块内所有函数值之和，则易知对于每次修改操作 x 而言，任意 sum_i 的变化量即为 $\Delta_{a_x} \cdot cnt_{i,x}$，于是单次修改操作的代价为 $\Theta(\frac{n}{k})$。

接下来处理询问操作，则所有对于完整函数块的询问均可以 $\Theta(1)$ 回答。因此，对于每次询问操作而言，区间首尾处的非完整函数块内的函数值只需再借助树状数组维护即可，于是单次询问操作的代价为 $\Theta(k \log n)$。此时，易知应取 $k = \sqrt{\frac{n}{\log n}}$ 使得总复杂度最优。

时间复杂度：$\Theta((n+q)\sqrt{n \log n})$

空间复杂度：$\Theta(n\sqrt{n \log n})$

40 GERALD07

题目大意

给定一张 n 个点 m 条边的无向图，要求处理 q 次询问 (l_i, r_i)，查询当仅保留编号在 $[l_i, r_i]$ 内的所有边时，图中连通块的个数。

时间限制：8 s

数据规模

- $1 \leq t \leq 10^3$，$1 \leq n, m, q \leq 2 \times 10^5$，$\sum n_i, \sum m_i, \sum q_i \leq 2 \times 10^5$。

算法分析

为方便讨论，不妨记 T_i 为仅由编号在 $[i, m]$ 内的所有边构成的生成森林，则询问 (l_i, r_i) 等价于查询 T_{l_i} 中至多有多少条编号不超过 r_i 的边。因此，不难想到将每条边的编号视作边权，则此时令 T_i 为赋予边权后的最小生成森林，于是所有 T_i 形态均是唯一的，则只需借助 Lct 即可构建出所有 T_i，而计数部分可以借助树状数组解决。

时间复杂度：$\Theta(m \log n + q \log m)$

空间复杂度：$\Theta(n + m)$

41 GNUM

题目大意

给定两个长度为 n 的正整数序列 $\{a_i\}$ 和 $\{b_i\}$，每次操作要求选取一个四元组 (i,j,p,q)，并且满足：

- $a_i < b_j$ 并且 $gcd(a_i, b_j) > 1$；
- $b_p < a_q$ 并且 $gcd(a_q, b_p) > 1$；
- $gcd(gcd(a_i, b_j), gcd(a_q, b_p)) > 1$。

对于任意两次操作而言，要求有序二元组 (i,j) 不能相同，(p,q) 也不能相同。求满足上述条件的最多连续操作次数。

时间限制：1 s

数据规模

- $1 \le t \le 10$，$1 \le n \le 400$，$1 \le a_i, b_i \le 10^9$。

算法分析

首先，由于 $a_i < b_j$ 和 $a_i > b_j$ 不可能同时满足，则易知本题可以建立二分图模型：将所有满足条件一的二元组视为左点集，将所有满足条件二的二元组视为右点集，则条件三即为连边条件。此时，最多连续操作次数即为上述二分图的最大匹配数。然而，由于二分图中边数难以承受，因此考虑建立若干虚拟节点，以此表示序列 $\{a_i\}$ 和 $\{b_i\}$ 中出现的所有数的质因子，则易证对于任意二元组 (i,j) 而言，只需将该二元组和所有表示 $gcd(a_i, b_j)$ 的质因子的虚拟节点连边即可，此时该图的最大流即为所求答案。由于 10^9 以内的任意整数至多包含 9 个不同的质因子，因此新图的点数和边数均可以承受。

时间复杂度：$\Theta(MaxFlow(n^2, n^2))$

空间复杂度：$\Theta(n^2)$

42 GRAPHCNT

题目大意

给定一张 n 个点 m 条边的有向图，要求统计合法无序二元组 (x,y) 的总对数。其中，定义无序二元组是 (x,y) 合法的，当且仅当存在从 1 到 x 的路径 p_x 和从 1 到 y 的路径 p_y，满足 p_x 和 p_y 不存在除 1 号点以外的公共点。

时间限制：2 s

数据规模

- $1 \leq n \leq 10^5$，$0 \leq m \leq 5 \times 10^5$。

算法分析

通过简单的证明不难发现，无序二元组 (x,y) 合法，当且仅当 x 和 y 在支配树上最近公共祖先为 1 号点，即 x 和 y 不存在除 1 以外的其他公共必经点。因此，本题只需利用经典的 $Lengauer-Tarjan$ 算法构建出支配树即可解决，具体实现时应当注意将无法从 1 号点到达的无效点及时剔除，并且在实现并查集部分时应当避免按秩合并。最后，我们给出上述结论的简要证明。

必要性

上述结论的必要性显然：若 x 和 y 在支配树上存在异于 1 号点的公共祖先 z，则任意一条从 1 到 x 的路径 p_x 必然经过 z，任意一条从 1 到 y 的路径 p_y 也必然经过 z，因此 p_x 和 p_y 必然存在公共点 z。

充分性

上述结论的充分性我们尝试借助网络流模型来证明。首先，仅保留原图中对于点 x 和点 y 有效的点和边构成图 $G_{x,y}$。对于任意有向边 $<u,v>$ 而言，此边有效当且仅当存在某条形如 $1 \to u \to v \to x$ 或者 $1 \to u \to v \to y$ 的简单路径。对于任意点 i 而言，所谓对点 x 和点 y 有效当且仅当点 i 同时满足：

- 存在从 1 到 i 的路径；
- 存在从 i 到 x 或者从 i 到 y 的路径。

接下来，我们构建网络流图。对于 $G_{x,y}$ 中的任意点 i，将其拆点为 i 和 i'，并且连接一条容量为 1 的边 $<i,i'>$。特别地，边 $<1,1'>$ 的容量设为 ∞。对于 $G_{x,y}$ 中的任意有向边 $<u,v>$, 连接一条容量为 ∞ 的边 $<u',v>$。最后，将点 1 视作源点，并新增虚拟汇点 t，连边 $<x',t>$ 和 $<y',t>$，容量均为 ∞。

此时，若能证明 $<1,t>$ 最大流至少为 2，则由建图方式可知，必然存在至少两条从 1 到 t 的不相交路径。同时，又由于从 1 到 t 的流量必然经过 $<x,x'>$ 或者 $<y,y'>$，因此 $<1,t>$ 最大流至多为 2，并且必然分别流经 x 和 y。于是，这两单位流量经过的路径便对应于原图中从 1 到 x 和从 1 到 y 的不相交路径。

为证明 $<1,t>$ 最大流至少为 2，根据最大流-最小割定理易知，这等价于求证 $<1,t>$ 最小割至少为 2。因此，由于该图最小割肯定为正，我们假设其最小割为 1。根据上述的

建图方式可知，只有边 $< i, i' >$ 的容量为 1，其中 i 是某个异于 1 的点。鉴于 $< x', t >$ 和 $< y', t >$ 的容量均为 ∞，我们可以断定 x' 和 y' 必然与 t 处于同一点集中。于是，最小割 $< i, i' >$ 的存在表明 i 是从 1 到 t 的必经点，那么 i 也必然是从 1 到 x 和从 1 到 y 的必经点，这与最初的假设矛盾。

综合上述证明，若点 x 和点 y 在支配树上不存在除 1 以外的其他公共祖先，则必然存在两条从 1 到 x 和从 1 到 y 的不相交路径。

时间复杂度：$\Theta((n+m)\log n)$

空间复杂度：$\Theta(n+m)$

43 GTHRONES

题目大意

给定 n 种正整数 u_i，其中第 i 种正整数恰有 c_i 个。先手玩家可以任选一个正整数作为当前数，然后从后手玩家开始轮流操作，其中每步操作要求在剩余所有数中任选一个正整数 v，使得 v 和当前数 u 满足下列条件之一：

- $v|u$ 且 $\frac{u}{v}$ 为质数；
- $u|v$ 且 $\frac{v}{u}$ 为质数。

若某个玩家不存在可行操作，则该玩家失败，否则 v 成为新的当前数。要求判定当两名玩家均采取最优策略时，先手玩家是否必胜。若先手必胜，要求选择的初始正整数的最小可行值。

时间限制：1 s

数据规模

- $1 \leq n \leq 500$，$1 \leq u_i \leq 10^{18}$，$1 \leq c_i \leq 10^9$。

算法分析

首先不难发现，由于每步操作均使得当前数质因子个数的奇偶性发生改变，因此根据质因子个数的奇偶性即可建立二分图模型 G，其中 c_i 即为每个点的匹配数上限。此时，不难证明 x 为先手必胜点当且仅当，存在某个 G 的最大匹配 M，使得 x 不是 M 中的匹配点。因此，我们只需在求得某个最大匹配 M 后从所有未匹配点开始进行遍历即可。在具体实现时，建图部分需要借助 $Miller-Rabin$ 算法进行素性检测。

时间复杂度：$\Theta(n^2 \cdot PrimalityTest + MaxFlow(n, n^2))$

空间复杂度：$\Theta(n^2)$

44 HAMILG

题目大意

给定一张 n 个点 m 条边的无向图，首先由玩家一选定起点，然后玩家二和玩家一轮流进行移动，直到不能移动者为负。其中，每次移动必须移动到某个从未经过的相邻点。要求统计对于玩家一而言，有多少个点作为起点有必胜策略。

时间限制：$2 \sim 5$ s

数据规模

- $1 \leq t \leq 100$，$1 \leq n \leq 2 \times 10^3$，$n-1 \leq m \leq \binom{n}{2}$，$\sum m_i \leq 10^6$。

算法分析

首先易知，由于是由玩家二率先进行移动，因此对于玩家一而言，拥有必胜策略的起点即为图中所有先手必败点。接下来我们证明，点 x 为先手必败点当且仅当，存在该图的某个最大匹配 M，使得点 x 不在 M 上。

必要性

若点 x 必然处在最大匹配上，则显然不存在以点 x 为起点且长度为偶数的交错路径。因此，若以点 x 为起点，则无论后手如何移动，先手每次必然可以沿着匹配边移动，即点 x 为先手必胜点。于是，所有必然处在最大匹配上的点一定不是先手必败点。

充分性

若存在该图的某个最大匹配 M，使得点 x 不在 M 上，则对于任意与点 x 相邻的点 y 而言，点 y 必然在 M 上，否则 M 便不是该图的最大匹配。因此，对于去除点 x 后的新图而言，点 y 必然处在新图的最大匹配上，即点 y 为新图的先手必胜点，于是点 x 即为原图的先手必败点。

借助上述结论，则本题只需利用带花树算法即可解决。

时间复杂度：$\Theta(MaxMatch(n))$

空间复杂度：$\Theta(n+m)$

45 HYPER

题目大意

定义 3-均匀超图为每条超边均连接三个不同点的图，定义 3-超树为去掉任意一条超边后均不连通的 3-均匀超图。给定点数 n，统计恰好具有 n 个点的 3-超树总数。

时间限制：1 s

数据规模

- $1 \leq t \leq 15$，$3 \leq n \leq 17$。

算法分析

由于本题数据范围较小，因此可以直接本地暴力打表后提交。而至于本地打表，由于无需过分计较效率，于是可以通过状压当前所有连通块的大小解决。

时间复杂度：$\Theta(1)$

空间复杂度：$\Theta(1)$

46 *KALKI

题目大意

给定平面直角坐标系中的 n 个点 (x_i, y_i)，要求构造一棵生成树 T，并且最小化其代价 $c = \max_i\{c_i\}$。其中，定义 $r_i = \max_{(i,j)\in T}\{d_{i,j}\}$，则 $c_i = \sum_j [r_j \geq d_{j,i}]$。

时间限制：5 s

数据规模

- $1 \leq t \leq 100$，$3 \leq n \leq 400$，$-2 \times 10^3 \leq x_i, y_i \leq 2 \times 10^3$。

算法分析

由于本题保证数据随机，因此我们应当尽量使得 T 中的边不能过长。于是，我们不妨交替地沿着 X 轴和 Y 轴将当前点集均分，并递归处理两侧点集，然后再连接两侧点集中最靠近分割线处的两点即可。

上述算法在 $Codechef$ 上可得 $0.137pt$。

时间复杂度：$\Theta(n \log^2 n)$

空间复杂度：$\Theta(n)$

47 KNGHTMOV

题目大意

给定平面直角坐标系中的起点 $(0,0)$ 和终点 (x,y)，以及两种操作 (a_x,a_y) 和 (b_x,b_y)。每步操作允许从当前点 (u,v) 转移到 $(u+a_x,v+a_y)$ 或者 $(u+b_x,v+b_y)$，但不得停留在 k 个给定的障碍点 (px_i,py_i) 上。要求统计从起点到达终点的方案数，或者判定存在无数组不同的可行方案。

时间限制：9 s

数据规模

- $1\le t\le 5$，$0\le k\le 15$，$-500\le x,y,a_x,a_y,b_x,b_y,px_i,py_i\le 500$。

算法分析

本题具体实现并不复杂，但是边界细节较多。若分别记 $\vec{a}=(a_x,a_y)$、$\vec{b}=(b_x,b_y)$、$\vec{t}=(x,y)$，则我们分为以下两种情形进行讨论：

- 若 $\vec{a}$ 与 $\vec{b}$ 不共线，则易知所有有效点都能以 $\vec{a}$ 和 $\vec{b}$ 为基底唯一分解。此时，只需按照新坐标依次扫描所有障碍点，并利用组合数进行容斥即可；
- 若 $\vec{a}$ 与 $\vec{b}$ 共线，则只有当 $\vec{t}$ 也与 $\vec{a}$、$\vec{b}$ 共线时才可能有解。此时，若 $\vec{a}$ 和 $\vec{b}$ 中存在零向量，则只需特判即可，否则所有点均可被投影到一维坐标上，并且问题转化为判定决策序列中是否存在环以及统计路径条数，这些均可通过简单的迭代完成。

由上述讨论可知，本题边界情形较多，代码中应当注意细节的实现。

时间复杂度：$\Theta(|x||y|)$

空间复杂度：$\Theta(|x||y|)$

48 LEBOXES

题目大意

给定 n 个盒子，其中第 i 个盒子内有 $p_i\%$ 的概率为 v_i 元，而有 $1-p_i\%$ 的概率为一颗钻石。现有 m 种物品各一件，其中购买第 j 件物品需要恰好花费 c_j 元以及 d_j 颗钻石。若在打开所有盒子后必然采取最优策略，即购买尽可能多的物品，求期望能够购买到的物品件数。

时间限制：4 s

数据规模

- $1 \le t \le 5$，$2 \le n \le 30$，$1 \le m \le 30$，$1 \le v_i, c_j \le 10^7$，$0 \le d_j \le 30$，$0 \le p_i \le 100$。

算法分析

由于本题中 n 和 m 的范围均较小，因此不难想到利用 $Meet-in-the-middle$ 的思想。首先，考虑将 n 个盒子分为 k 个和 $n-k$ 个两部分，并分别暴力枚举出所有 2^k 和 2^{n-k} 种可能的状态。接下来，我们尝试合并两部分的所有状态。此时，由于在购买时必然采取最优策略，因此不妨记 $dp_{i,j}$ 为利用至多 i 颗钻石购得 j 件物品所需的最少钱数，则显然所有 dp 值均可在预处理时求得，该部分代价为 $\Theta(nm^2)$。于是，我们考虑枚举第一部分中可能的每种状态，则只需执行 $\Theta(nm)$ 次二分即可求得其总贡献。在具体实现中，取 $k=\frac{n}{3}$ 时总复杂度较低。

时间复杂度：$\Theta(nm^2+2^{\frac{n}{3}}n^2m+2^{\frac{2n}{3}}n)$

空间复杂度：$\Theta(nm+2^{\frac{2n}{3}})$

49 LECOINS

题目大意

给定 n 种类型的硬币各无限枚，其中第 i 种硬币面值为 v_i 且颜色为 c_i。给定 q 组询问 s_i，查询至多可以由多少种不同颜色的硬币凑出面额 s_i。

时间限制：4 s

数据规模

- $1 \le n \le 30$，$1 \le q \le 2 \times 10^5$，$1 \le v_i \le 2 \times 10^5$，$1 \le c_i \le 10^9$，$1 \le s_i \le 10^{18}$。

算法分析

首先不考虑颜色种类数，我们令 $m = v_n$，并且记 f_i 为当前所有硬币能够拼凑出的面额 x 中，满足 $x \equiv i \pmod m$ 的最小值，则能够拼凑出给定面额 s 当且仅当 $f_{s \bmod m} \le s$。于是，易知 f 可以利用类似于最短路的过程求得，但由于转移路径中存在环，因此 $Bellman–Ford$ 算法最坏情形下需要迭代 m 轮。此时，若单独考虑第 i 种硬币，则该类转移边在图中必然构成 $gcd(v_i, m)$ 个长度为 $\frac{m}{gcd(v_i,m)}$ 的有向环。同时，由于硬币拼凑面额的过程与顺序无关，因此对于每个独立的环而言，我们只需以当前环上的最小值为起点，顺序遍历整个环并更新答案即可，此部分总代价为 $\Theta(nv)$。

接下来考虑颜色种类数的问题，则类似的我们可以记 $f_{i,j}$ 为在至多使用 i 种颜色的硬币的前提下，当前所有硬币能够拼凑出的面额 x 中，满足 $x \equiv j \pmod m$ 的最小值。于是，我们不妨对于同种颜色的硬币同时处理，并且先执行跨层的所有转移，再在同层内部执行转移。此时，为保证转移前的值不被覆盖，则我们只需倒序执行所有跨层转移即可。

时间复杂度：$\Theta(n^2v + nq)$

空间复杂度：$\Theta(nv)$

50 LPARTY

题目大意

给定一个 n 元布尔函数，已知有且仅有给定的 m 组输入对应的函数值为 1，要求用若干条合法的规则归纳该函数的取值，并且最小化所选规则的总代价。其中，定义一条规则为一个长度为 n 的三进制串，其中第 i 位的取值表示输入中的第 i 个变量必须取值为 $0/1$ 或者无限制，则一条规则是合法的，当且仅当所有满足该条规则的输入对应的函数值均为 1，并且定义选取该条规则的代价为对输入变量的限制条数。同时，一种合法的方案要求所有不满足任意一条所选规则的输入对应的函数值均为 0。

时间限制：1.2 s

数据规模

- $1 \leq t \leq 120$，$1 \leq n \leq 5$，$0 \leq m \leq 10^3$。

算法分析

本题是一道限制较紧的搜索优化题。首先不难发现，可能的规则总数至多为 3^n。同时，若两条可行规则之间存在包含关系，则显然可以去除较多的冗余规则。此时，我们考虑暴力枚举剩余规则的状态，则利用可行性剪枝以及最优性剪枝可以极大地加速搜索过程。在具体实现时，我们可以采用 Bfs 的方式生成所有可能的状态。

时间复杂度：$\Theta(CountState)$

空间复杂度：$\Theta(CountState)$

51 LUCKYDAY

题目大意

给定一个初值为 $s_1 = a$ 和 $s_2 = b$ 的递推数列 $s_i = (x \cdot s_{i-1} + y \cdot s_{i-2} + z) \bmod p$ $(i \geq 3)$，其中 p 为一质数。给定 q 组询问 $[l_j, r_j]$，要求统计满足 $s_k = c$（$l_j \leq k \leq r_j$）的总数。

时间限制：5 s

数据规模

- $1 \leq t \leq 2$，$2 \leq p \leq 10007$，$1 \leq q \leq 2 \times 10^4$，$1 \leq l_j \leq r_j \leq 10^{18}$。

算法分析

首先不难发现，由本题中给定的为线性递推式，于是显然可以转化为如下矩阵形式：

$$\begin{bmatrix} s_i \\ s_{i+1} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ y & x & z \\ 0 & 0 & 1 \end{bmatrix}^{i-1} \begin{bmatrix} s_1 = a \\ s_2 = b \\ 1 \end{bmatrix}$$

此时，由于模数 p 为质数，因此转移矩阵 T 在模 p 意义下奇异，当且仅当 $det(T) = -y$ 在模 p 意义下不存在逆元，即 $y = 0$。于是，我们接下来分为两种情形进行讨论。

$y = 0$

在此种情形下，递推式简化为 $s_i = (x \cdot s_{i-1} + z) \bmod p$，则易知其周期必然不超过 p，于是只需暴力求得其周期即可处理所有询问。

$y \neq 0$

此时，假设我们已经求得整个序列的周期为 t，即满足 $T^t(a, b, 1)^{\mathrm{T}} = (a, b, 1)^{\mathrm{T}}$ 的最小正整数解。同时，由于所有询问中的 c 均为同一值，于是假设已知所有 $T^{t-1}(a, b, 1)^{\mathrm{T}} = (c, i, 1)^{\mathrm{T}}$（$0 \leq i < p$）的最小正整数解 t_i，则 t_i 即为函数值二元组 (c, i) 首次出现的下标。

于是，对于任意询问 $[l_j, r_j]$ 而言，首先可以分解为两个相互独立的询问 $[1, l_j - 1]$ 和 $[1, r_j]$。因此，对于任意询问 $[1, x]$ 而言，所有完整周期的总贡献为 $m \cdot \lfloor \frac{x}{t} \rfloor$，其中 m 为一个完整周期中出现的函数值二元组 (c, i) 的不同种类数。而对于剩余的非完整周期，则只需利用二分即可解决，于是单次询问的代价为 $\Theta(\log p)$。

最后，我们处理所有形如 $T^t(a, b, 1)^{\mathrm{T}} = (a', b', 1)^{\mathrm{T}}$ 的最小正整数解。此时，利用 $Baby - step - giant - step$ 的思想，我们不妨令 $t = uk + v$，则有：

$$\begin{bmatrix} 0 & 1 & 0 \\ y & x & z \\ 0 & 0 & 1 \end{bmatrix}^{v} \begin{bmatrix} a \\ b \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ y & x & z \\ 0 & 0 & 1 \end{bmatrix}^{-uk} \begin{bmatrix} a' \\ b' \\ 1 \end{bmatrix}$$

因此，我们考虑将所有 $T^v(a, b, 1)^{\mathrm{T}}$（$0 \leq v < k$）哈希存储，则每次询问只需暴力枚举 u 并进行验证即可。其中，$T^{-1} \equiv det(T)^{-1} T^* \pmod p$，$T^*$ 为 T 的伴随矩阵。

由于整个序列的周期至多为 p^2，并且 u 的上界为 $\frac{p^2}{k}$，于是该部分总代价为 $\Theta(k+p\cdot\frac{p^2}{k})=\Theta(k+\frac{p^3}{k})$。因此，易知当取 $k=p^{\frac{3}{2}}$ 时总代价最低。

时间复杂度：$\Theta(p^{\frac{3}{2}}+q\log p)$

空间复杂度：$\Theta(p^{\frac{3}{2}})$

52 LYRC

题目大意

给定由 w 个单词 p_i 构成的字典，以及由 n 个片段 s_i 构成的文本。求字典中的每种单词在文本中的出现总次数。

时间限制：3 s

数据规模

- $1 \le w \le 500$，$1 \le |p_i| \le 5 \times 10^3$，$1 \le n \le 100$，$1 \le |s_i| \le 5 \times 10^4$。

算法分析

本题是经典的多模式串匹配问题，只需借助 AC 自动机即可解决。对于统计出现总次数的问题，自动机中的节点 i 每被遍历到一次，都相当于节点 $i, fail_i, fail_{fail_i}, \ldots$ 所代表的字符串出现次数均增加一次。因此，当由所有模式串构成的自动机对所有片段都执行一次之后，只需将 $fail$ 指针视作连边，然后遍历一次即可得到每个字符串的真实出现次数。

时间复杂度：$\Theta(|\Sigma| \cdot \sum |p_i| + \sum |s_i|)$

空间复杂度：$\Theta(|\Sigma| \cdot \sum |p_i| + \sum |s_i|)$

53 MAGIC

题目大意

给定一张 n 个点 m 条边的简单无向图，初始时两名玩家分别位于 1 号点和 2 号点，且各有 p 点能力值。两名玩家交替依次执行如下游戏步骤：

- 当前玩家可以沿着已有无向边移动到任意一个可达点，若移动后与另一玩家位于同一点处，则当前玩家获胜；
- 当前玩家必须在现有图中增加恰好一条当前不存在的无向边，若不存在可行方案则当前玩家失败；
- 若当前玩家现有能力值为正，则其可以选择通过消耗一点能力值移动到图上任意一点。

要求判定在两名玩家均采取最优策略的前提下的获胜方。

时间限制：5 s

数据规模

- $1 \leq t \leq 100$，$2 \leq n \leq 7777$，$0 \leq m \leq 10^4$，$0 \leq p \leq 10^4$。

算法分析

通过一定的分析可以发现，本题中的博弈可以采取贪心策略，且易知如下一些简单结论：

- 若两名玩家的初始位置处于同一连通块中，则显然先手必胜；
- 在最优策略下，显然玩家会尽可能新建连通块内的边，而不会主动合并不同连通块。因此，我们仅需关心所有连通块的大小以及各个连通块内剩余边数的总和；
- 于是，在最后分出胜负的前一回合，整张图必然由两张完全子图构成。

基于上述结论，则我们可以借助奇偶性来判定胜负：

- 若 n 为奇数，则分出胜负时图中的总边数奇偶性固定，因此只需根据图中的初始边数即可判定胜负；
- 若 n 为偶数，则图中的初始边数决定了先手最优策略下最后两个连通块的奇偶性，于是只需结合初始位置所在连通块的奇偶性即可判定胜负。

时间复杂度：$\Theta((n+m)\alpha(n))$

空间复杂度：$\Theta(n)$

54 MARTARTS

题目大意

给定一张左右点集各有 n 个点的完全二分图，其中每条边 (i,j) 均有两个权值 $a_{(i,j)}$ 和 $b_{(i,j)}$。定义任意一组完备匹配 M 的权值为 $(u=\sum_{e\in M\setminus\{e'\}}a_e, v=\sum_{e\in M\setminus\{e'\}}b_e)$，其中，$e'$ 为 M 中在最大化 $v-u$ 的前提下最大化 v 的某条匹配边。要求给出一组合法的完备匹配，并且在最大化 $u-v$ 的前提下最大化 u。

时间限制：5 s

数据规模

- $1\le n\le 100$，$0\le a_{(i,j)},b_{(i,j)}<10^{12}$。

算法分析

首先易知，当 $u-v$ 的值固定后，最大化 u 和最大化 v 其实是等价的。因此，对于某个固定的完备匹配 M 而言，则 e' 显然应在最大化 $c_{e'}=a_{e'}-b_{e'}$ 的前提下最大化 $b_{e'}$。于是，我们不妨给所有边 e 赋予新的权值二元组 $(c_e=a_e-b_e,b_e)$，则不难发现这类似于最大权匹配模型。

为方便接下来的表述，我们不妨定义新权值二元组上的全序关系 R 为：$(c,b)\ R\ (c',b')$ 当且仅当 $c<c'$ 或者 $c=c'$ 且 $b>b'$。于是，我们考虑按 R 的逆序暴力枚举 e'，则当前图中仅保留所有满足 $e\ R\ e'$ 的边 e。此时，为强制令 e' 出现在 M 中，则只需暂时令 e' 的权值为 $(\infty,0)$ 即可，而此种情形下最优方案的权值即为 $M\setminus\{e'\}$ 中所有匹配边的权值之和。

在上述算法中不难发现，我们仅需在动态维护最大权匹配时支持修改某条边权和删除某条边。此时，易知修改某条边权后仅会影响至多一组匹配点，于是我们仅需重新增广该组匹配点中的左点即可。同时，由于删除某条边可以等价为将该条边权修改为 $(-\infty,0)$，因此我们同样可以采用类似的方法进行维护。在具体实现时，由于边权均为二元组的形式，因此所有顶标和边权的增减以及大小比较均需由二元组代替。

时间复杂度：$\Theta(n\cdot MaxMatch(n))$

空间复杂度：$\Theta(n^2)$

55 MATCH

题目大意

给定一张 $n \times m$ 的完全二分图，其中每条边 (i, j) 以 $p_{i,j}$ 的概率出现，求该二分图最大匹配的期望。

时间限制：3 s

数据规模

- $1 \leq n \leq 5$，$1 \leq m \leq 100$，$0 \leq p_{i,j} \leq 1$。

算法分析

由 $Hall$ 定理可知，二分图 $G = (V_l, V_r, E)$ 存在大小为 $|V_l|$ 的匹配，当且仅当 $|V_S| \geq |S|$ 对任意 $S \subseteq V_l$ 均成立，其中 V_S 为与 S 相邻的右点集合。于是可以推知，二分图 G 存在大小为 x 的匹配当且仅当，存在 $V_l' \subseteq V_l$ 使得其子图 $G' = (V_l', V_r, E')$ 满足 $Hall$ 定理，并且 $|V_l'| \geq x$。

由于左侧点数极少，因此不妨考虑对 V_l 的所有子集进行状压，用 2^n 个二进制位分别表示每个左点集合是否满足 $Hall$ 定理。为方便接下来的讨论，记 $f_{i,j}$ 为所有左点集合状态为 i 的子图 $G_j = (V_l, V_j, E_j)$ 的出现概率，其中 V_j 表示前 j 个右点构成的集合，记 $next_{i,j}$ 为左点集合状态 i 新增一个右点 x 后的新左点集合状态，其中要求与 x 相邻的左点集合为 j，记 $pr_{i,j}$ 为第 i 个右点相邻的左点集合恰好为 j 的概率。此时，不难推知下列关于 f 的转移方程：

$$f_{i,j} = \sum_{i'} \sum_{k} [next_{i',k} = i] f_{i',j-1} pr_{j,k}$$

于是，易知 $Ans = \sum_i f_{i,m} match_i$，其中 $match_i$ 为左点集合状态为 i 的二分图的最大匹配。此时，通过实践发现合法的左点集合状态十分稀少，因此只需对所有合法状态哈希存储即可解决本题。

时间复杂度：$\Theta(2^n nm + CountState \cdot 2^n(n+m))$

空间复杂度：$\Theta(2^n m + CountState \cdot (2^n + m))$

56 MAXCIR

题目大意

已知三点 a、b、c 的坐标，给定 n 个向量 $p_i = (x_i, y_i)$，要求选择至多 k 个向量作用于点 a，最大化三角形 abc 的周长。

时间限制：3 s

数据规模

- $1 \le n \le 500$，$0 \le k \le n$。

算法分析

首先不难发现，由于仅可能改变 a 的坐标，因此最大化三角形 abc 的周长等价于最大化 $|ab|+|ac|$。此时，我们考虑证明一定存在两个实数 u 和 v，使得在最大化 $f(a) = u \cdot a_x + v \cdot a_y$ 的同时即可最大化 $|ab| + |ac|$：

- 若记最优解为 a'，则所有可行解均位于一个以 b 和 c 为焦点且经过的 a' 的椭圆内；
- 此时，作该椭圆在 a' 处的切线，则易知该条直线方程中 x 和 y 前的系数即为一组满足条件的 u 和 v。

于是，对于固定的 u 和 v 而言，由于 $f(a') = f(a) + \sum_{i \in Ans} f(p_i)$，则我们只需在所有给定向量中选取前 k 大的正贡献即可。此时，由上述选择方式可知，最优解中所选的向量集仅与所有向量间贡献的相对大小关系以及各自的正负性有关。因此，我们不妨求出所有可能令 $f(p_i) = f(p_j)$ 或 $f(p_i) = 0$ 的方向向量 (u, v)，则划分出的任意一个极角区间内均有唯一可能的最优解。于是，我们只需在顺序扫描所有极角区间的同时维护所有给定向量的相对大小关系即可，则单次查询操作仅需进行一次二分。在具体实现时，由于本题精度要求较高，因此在求平方根时不能计算，而应利用下式：

$$\sqrt{x} = [\sqrt{x}] + \{\sqrt{x}\} = [\sqrt{x}] + \frac{x - [\sqrt{x}]^2}{[\sqrt{x}] + \sqrt{x}}$$

其中，上式右侧的 $\sqrt{x}$ 可以直接进行计算。

时间复杂度：$\Theta(n^2 \log n)$

空间复杂度：$\Theta(n^2)$

57 *MAXRECT

题目大意

给定 $h \times w$ 的整数矩阵 $a_{i,j}$，要求选取一个非空行号集合 R 和一个非空列号集合 C，最大化 $\sum_{i \in R, j \in C} a_{i,j}$。

时间限制：2 s

数据规模

- $1 \leq h, w \leq 300$，$|a_{i,j}| \leq 10^9$。

算法分析

由于本题中给定对象为二维矩阵，因此若已知 R 或 C 其中一个集合，则另一个集合的最优解只需一次遍历即可求得。于是，我们不妨首先固定 R，并求得此种情形下 C 的最优解，然后再固定 C，并求得此种情形下 R 的最优解。此时，若我们利用上述方式不断迭代 R 和 C，则当 R 和 C 均不再变化时即求得一个局部最优解。因此，我们以适当的概率随机设定多组初始集合 R 和 C，并分别求得相应的局部最优解即可。

上述算法在 $Codechef$ 上可得 $0.997pt$。

时间复杂度：$\Theta(CountRandom \cdot hw)$

空间复杂度：$\Theta(hw)$

58 MINESREV

题目大意

给定一个 $r \times c$ 的扫雷游戏局面，已知所有雷的分布情况。定义逆扫雷游戏如下：

- 初始时所有网格均呈已知状态，每次操作可以任选一个已知状态的网格 (x, y)；
- 若 (x, y) 是雷区，则该步操作仅将 (x, y) 变为未知状态；
- 若 (x, y) 是空地，则对于任意已知网格 (x', y')，(x', y') 将变为未知状态当且仅当存在某个网格 (u, v)，使得在正常扫雷游戏中点击 (u, v) 将同时开启 (x, y) 和 (x', y')；
- 当所有网格均变为未知状态时，游戏结束。

给定 t 组初始游戏局面，要求分别最小化逆扫雷游戏中的操作次数。

时间限制：5 s

数据规模

- $1 \leq t \leq 50$，$1 \leq r, c \leq 50$。

算法分析

为方便讨论，我们不妨首先像正常扫雷游戏中一样计算出每个网格相邻的雷区数量 $cnt_{i,j}$，并将满足 $cnt_{i,j} = 0$ 的空地 (i, j) 称为自由的。于是，易知在正常扫雷游戏中，每次点击自由空地 (i, j) 开启的区域是 (i, j) 所在的自由空地连通块，以及该连通块边界上的非自由空地。因此，对于逆扫雷游戏中的网格 (i, j)，若 (i, j) 是雷区或者非自由空地 (i, j) 相邻网格中也不存在自由空地，则 (i, j) 必然需要单独一次操作。否则，若 (i, j) 是自由空地则只能消除其所在的连通块，而当 (i, j) 是非自由空地时，以其为边界的所有连通块均会被消除。同时，由八连通的性质可知，非自由空地 (i, j) 相邻的自由空地至多只会属于两个不同的连通块，因此每次操作至多也只会消除两个连通块。

此时，不难想到对自由空地连通块建立匹配模型，存在边 (i, j) 当且仅当连通块 i 和 j 的边界上存在公共的非自由空地。于是，由于每次操作应当消除尽量多的连通块，则在该模型中等价于求一般图的最大匹配，因此只需利用带花树算法即可。

时间复杂度：$\Theta(MaxMatch(rc))$

空间复杂度：$\Theta(rc)$

59 MISINT2

题目大意

定义一个仅由小写英文字母构成的字符串 s 是特殊的，当且仅当 $(s_1, s_2, s_3, \ldots) = (s_2, s_4, s_6, \ldots, s_1, s_3, s_5, \ldots)$。给定 t 组询问 (l_i, r_i)，要求分别统计长度在 $[l_i, r_i]$ 内的特殊字符串总数。

时间限制：10 s

数据规模

- $1 \le t \le 5$，$1 \le l \le r \le 10^{10}$，$r - l \le 5 \times 10^4$。

算法分析

通过一些尝试不难发现，当字符串长度为 n 时，题中所给的映射即为置换 $p_n(i) \equiv 2i \pmod{n'}$, 其中 n' 为不小于 n 的最小奇数，因此长度为 n 的特殊字符串总数为 26^{f_n}，其中 f_n 表示置换 $p_n \equiv 2n \pmod{n'}$ 中环的总数，则易知 $f_{2n} = f_{2n+1} - 1$，于是我们只需对所有奇数进行计算。同时，对于某个奇数 n 而言，可以推知 $f_n = \sum_{d|n} \frac{\phi(d)}{ord_d(2)}$，其中 $ord_d(2)$ 表示 2 在模 d 意义下的阶，证明如下：

- 对于 $1 \le i \le n$ 而言，易知同余方程 $i \equiv 2^j i \pmod{n}$ 的最小正整数解 j 即为置换 p_n 中 i 所在的环长。
- 若令 $d = gcd(i, n)$，则 $i = i'd$，此时有 $i' \equiv 2^j i' \pmod{\frac{n}{d}}$。由于 $gcd(i', \frac{n}{d}) = 1$，进一步可得 $2^j \equiv 1 \pmod{\frac{n}{d}}$，于是 j 即为 $ord_{\frac{n}{d}}(2)$。
- 易证 $gcd(i, n) = d$ 在 $[1, n]$ 内共有 $\phi(\frac{n}{d})$ 个解，并且这些数在置换中所处的环长均为 $ord_{\frac{n}{d}}(2)$，因此该种类型的环共有 $\frac{\phi(\frac{n}{d})}{ord_{\frac{n}{d}}(2)}$ 个，则置换中的总环数即为 $\sum_{d|n} \frac{\phi(d)}{ord_d(2)}$。

由阶的性质以及 $Euler$ 定理可知，$ord_n(2)$ 必然是 $\phi(n)$ 的因子。与此同时，当 x 和 y 互质时，易证 $ord_{xy}(2) = lcm(ord_x(2), ord_y(2))$，因此只需求得给定范围内所有数的质因数分解式即可解决本题。

时间复杂度：$\Theta((r - l) \log r)$

空间复杂度：$\Theta(r - l)$

60 MONOPLOY

题目大意

给定一棵 n 个点的有根树，初始时每个点的颜色各不相同，要求处理 q 次下列操作：

- $O\ u$：将点 u 到树根的路径上所有点的颜色改为一种从未使用过的颜色；
- $q\ u$：询问以 u 为根的子树中所有点到树根的路径长度的平均值，其中路径长度定义为路径上所有边权之和，并且任意一条树边的边权为 1 当且仅当其两个顶点异色，否则边权为 0。

时间限制：2 s

数据规模

- $1 \leq t \leq 15$，$1 \leq n, q \leq 10^5$，$\sum n_i, \sum q_i \leq 2 \times 10^5$。

算法分析

首先不难发现，其实本题中的修改操作即为 Lct 中的 $Access$ 操作，而路径长度则等价于路径上经过的虚边总条数。于是，由于 $Access$ 操作均摊仅会造成 $\Theta(\log n)$ 次虚实边转换，因此我们只需模拟 Lct 的过程即可将每次修改操作分解为 $\Theta(\log n)$ 次的虚实边转换。此时，我们考虑任意一点 v 对询问点 u 的贡献：

- 若 (fa_v, v) 为实边，则 v 必然不产生贡献；
- 若 v 在以 u 为根的子树中，则其贡献为 $size_v$，其中 $size_v$ 表示以 v 为根的子树中的总点数；
- 若 v 为 u 的祖先，则其贡献为 $size_u$，否则其贡献为 0。

在上述方法中，不难发现我们可以将子树内的总贡献与祖先的总贡献分开统计。同时，由于所有修改以及询问操作仅涉及链上的操作，因此我们同样可以借助 Lct 来解决。

时间复杂度：$\Theta(q \log^2 n)$

空间复杂度：$\Theta(n)$

61 PARADE

题目大意

给定一张 n 个点 m 条边的带权有向图，要求选取若干条链或者环，链和环允许自交或者相交，定义一种合法方案的总代价如下：

- 对于每条选取的链（环），代价为该条链（环）的长度，重复经过的边长度应被累计；
- 对于每条选取的链，额外产生 c 单位代价；
- 对于每个未被任何一条链（环）覆盖的点，额外产生 c 单位代价。

一种合法方案的总代价为上述三类代价的总和，要求处理 k 次询问，每次询问对于给定的 c_i 计算最小总代价。

时间限制：4 s

数据规模

- $2 \leq n \leq 250$，$1 \leq m \leq 3 \times 10^4$，$1 \leq k \leq 10^4$，$1 \leq v_i, c_i \leq 10^4$。

算法分析

首先考虑对于单次询问的情形，为处理额外代价 c，不妨对任意两个不同点 i 和 j 都连一条长度为 c 的边 $< i, j >$，对任意点 i 都连一条长度为 c 的自环 $< i, i >$。于是，所有选取的链都形成环，且未被覆盖的点等价于选取该点的自环。因此，问题转化为选取若干个环覆盖所有点，并最小化总环长。

不难发现，上述问题可以建立网络流模型：对于任意点 i 拆点为 i 和 i'，并连接一条下限为 1、费用为 0 的边 $< i, i' >$，对于原图中任意边 $< x, y >$，连接一条容量为 ∞、费用为 $v_{(x,y)}$ 的边 $< x', y >$，则该网络流模型的最小费用可行环流即为答案。

接下来考虑多次询问的情形，首先由下界费用流的处理方式可知，上述网络流模型必然恰好需要增广 n 次。由于建图方式保证每次增广时必然存在长度为 c 的增广路径，则易知最后的连续若干次增广路径长度为 c。因此，我们考虑暂时去除图中新加的长度为 c 的边和自环，并对该图增广 n 次，则对于单次询问 c_i，所有长度超过 c_i 的增广路径都可以被长度为 c_i 的增广路径代替。同时，由于增广路径的长度单调不减，于是每次询问时只需二分即可。

时间复杂度：$\Theta(MinCostFlow(n, m) + k \log n)$

空间复杂度：$\Theta(n + m)$

62 PARSIN

题目大意

给定 n、m 和 x，要求计算下式：

$$\sum_{\substack{k_1,k_2,\ldots,k_m\in\mathbb{N}\\ k_1+k_2+\cdots+k_m=n}} \sin k_1x\cdot\sin k_2x\cdots\sin k_mx$$

时间限制：1.5 s

数据规模

- $1\le t\le 10$，$1\le m\le 30$，$1\le n\le 10^9$，$0\le x\le 2\pi$。

算法分析

为方便表述，不妨记 $f_{i,j}$ 表示当 $m=i$ 且 $n=j$ 时上述求和式的值，则不难推知：

$$f_{i,j}=\sum_{k_i=0}^{j}\sin k_ix\cdot f_{i-1,j-k_i}$$

显然，由于 n 的取值范围较大，直接利用上式求和的代价难以承受。此时，利用三角函数的基本性质，我们可以对上式进行如下简化：

- 当 $k_i=0$ 时，则有 $\sin k_ix=\sin 0=0$，于是此部分贡献为 0；
- 当 $k_i=1$ 时，则此部分贡献为 $\sin x\cdot f_{i-1,j-1}$；
- 当 $k_i\ge 2$ 时，则利用三角函数的基本性质可以推知：

$$\begin{aligned}\sin k_ix&=\sin(k_i-1+1)x\\&=\sin(k_i-1)x\cdot\cos x+\sin x\cdot\cos(k_i-1)x\\&=2\sin(k_i-1)x\cdot\cos x+\sin x\cdot\cos(k_i-1)x-\sin(k_i-1)x\cdot\cos x\\&=2\cos x\cdot\sin(k_i-1)x-\sin(k_i-2)x\end{aligned}$$

于是，此部分总贡献为 $2\cos x\cdot f_{i,j-1}-f_{i,j-2}$。

通过上述的简化，则递推式化简为：

$$f_{i,j}=\sin x\cdot f_{i-1,j-1}+2\cos x\cdot f_{i,j-1}-f_{i,j-2}$$

于是，由于 m 的取值范围较小，因此不难想到利用矩阵乘法来加速计算过程：

$$\begin{bmatrix}f_{1,n-1}\\f_{2,n-1}\\\vdots\\f_{m,n-1}\\f_{1,n}\\f_{2,n}\\\vdots\\f_{m,n}\end{bmatrix}=\begin{bmatrix}0&0&\cdots&0&1&0&\cdots&0\\0&0&\cdots&0&0&1&\cdots&0\\\vdots&\vdots&\ddots&\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&0&0&0&\cdots&1\\-1&0&\cdots&0&2\cos x&0&\cdots&0\\0&-1&\cdots&0&\sin x&2\cos x&\cdots&0\\\vdots&\vdots&\ddots&\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&\cdots&-1&0&0&\cdots&2\cos x\end{bmatrix}^{n-1}\begin{bmatrix}f_{1,0}=0\\f_{2,0}=0\\\vdots\\f_{m,0}=0\\f_{1,1}=\sin x\\f_{2,1}=0\\\vdots\\f_{m,1}=0\end{bmatrix}$$

借助上述转移矩阵，则只需利用矩阵快速幂即可解决本题。

时间复杂度：$\Theta(m^3 \log n)$

空间复杂度：$\Theta(m^2)$

63 PRIMEDST

题目大意

给定一棵 n 个节点的无根树，所有边权均为 1。随机选择树上相异的两点，求两点间距离为质数的概率。

时间限制：5 s

数据规模

- $2 \leq n \leq 5 \times 10^4$。

算法分析

首先，由于距离为质数的限制并没有一些有用的性质能为解题过程提供便利，于是我们退而求其次，转而求出每种距离的点对数。根据以往做题的经验，不难想到此题可以利用点分治来解决，因此我们每次仅需统计所有经过当前层分治重心的路径。

假设当前层分治重心为 c，若以 c 作为当前连通块的根，则易知经过 c 且长度为 d 的路径条数为 $\sum_{i<j}\sum_l s_{i,l}s_{j,d-l}$，其中 i 和 j 为当前连通块中与 c 相邻的点，$s_{x,y}$ 表示以 x 为根的子树中深度为 y 的点数。此时，通过观察可以发现该式的第二维呈现为卷积的形式，因此不难想到通过 FFT 来加速计算过程。不过，即使利用前缀和优化，此过程也需要执行 k 次长度为 w 的 FFT，其中 k 为重心 c 的度数，w 为当前连通块的大小，该复杂度在某些极限情形下可能达到 $\Theta(n^2 \log n)$。

为解决上述问题，我们发现其瓶颈在于分治重心度数过大，从而造成执行 FFT 的次数过多。由于本题的统计对象为路径长度，因此我们可以想到常见的多叉树转化为二叉树的方法，即：对于任意连接超过两个儿子的节点 x，在 x 的所有儿子中任选一个定作左子树，其余所有儿子都连向同一个新建的虚拟节点作为右子树，并且 x 与虚拟节点间的边权赋为 0，然后递归处理左右子树即可。在完成上述转化后，由于分治重心度数最多为 3，则单次代价为 $\Theta(w \log w)$，于是总代价便在可承受范围之内。

时间复杂度：$\Theta(n \log^2 n)$

空间复杂度：$\Theta(n)$

64 QPOINT

题目大意

给定 n 个相离的简单多边形，其中第 i 个简单多边形点数为 k_i。要求在线处理 q 次询问，每次查询某个给定点位于哪个简单多边形内，或者判定不在任何简单多边形内。

时间限制：3 s

数据规模

- $1 \leq n \leq 10^5$，$3 \leq k_i \leq 3 \times 10^5$，$\sum k_i \leq 3 \times 10^5$，$1 \leq q \leq 10^5$。

算法分析

首先考虑本题的离线版本，此时对于任意询问 (x', y') 而言，由于所有简单多边形均相离，因此所有横跨直线 $x = x'$ 的线段之间必然存在严格全序关系。于是，若 (x', y') 的正下方恰有偶数条线段，则 (x', y') 显然不在任何简单多边形内，否则其必然位于正下方紧邻线段所属的简单多边形内。此时，若将所有简单多边形顶点的横坐标离散化，则易知相邻横坐标之间的全序关系必然恒定不变。因此，我们考虑沿 x 正半轴方向进行扫描线，则线段的全序关系可以借助平衡树进行维护。对于强制在线询问而言，则只需借助可持久化平衡树即可解决。在具体实现时，需要注意特判所有简单多边形的顶点以及所有竖直方向上的线段。

时间复杂度：$\Theta((\sum k_i + q) \log \sum k_i)$

空间复杂度：$\Theta(\sum k_i \log \sum k_i)$

65 QPOLYSUM

题目大意

给定一个 d 次多项式 $P(x)$ 在模 m 意义下的前 $(d+1)$ 项，即 $P(0), P(1), \ldots, P(d)$。给定 n 和 q，要求计算：

$$(\sum_{i=0}^{n-1} P(i) \cdot q^i) \bmod m$$

时间限制：4 s

数据规模

- $1 \leq t \leq 5 \times 10^3$，$1 < m < 10^{18}$，$1 \leq n < 10^{10^5}$，$0 \leq d < 2 \times 10^4$，$\sum |n_i| \leq 10^5$，$\sum (d_i + 1) \leq 2 \times 10^4$。

算法分析

本题是一道较为复杂的数学推导题，若令 $G(n) = \sum_{i=0}^{n-1} P(i) \cdot q^i$，则可以通过数学归纳法证明：存在某个 d 次多项式 $F(x)$，使得 $G(n) = F(n) \cdot q^n - F(0)$。然后只需借助 $F(n)$ 的递推式即可求得 $F(x)$ 的前 $(d+1)$ 项，于是只需借助拉格朗日插值法即可求得 $F(n)$。

时间复杂度：$\Theta(|n| + d)$

空间复杂度：$\Theta(|n| + d)$

66 QRECT

题目大意

给定一个平面直角坐标系，要求依次处理 q 次以下三类操作：

- $I\ x_1\ y_1\ x_2\ y_2$：插入一个平行于坐标轴的矩形，并且以 (x_1, y_1) 和 (x_2, y_2) 为一对顶点；
- $D\ index$：删除在第 $index$ 次插入操作中插入的矩形；
- $Q\ x_1\ y_1\ x_2\ y_2$：询问当前有多少个矩形与给定矩形 $(x_1, y_1) - (x_2, y_2)$ 相交。

时间限制：2 s

数据规模

- $1 \le q \le 10^5$，$1 \le x_1 \le x_2 \le 10^9$，$1 \le y_1 \le y_2 \le 10^9$。

算法分析

由于本题并没有强制使用在线算法，因此显然我们可以对整个操作序列进行 Cdq 分治，则所有询问操作都在所有修改操作之后。同时，利用补集转化思想，每次询问时可以改为查询与给定矩形不相交的矩形总数。于是，给定矩形的四条边界将整个平面共分成 $3 \times 3 = 9$ 块区域。此时，所有整体位于左（右）侧一列的矩形只需在排序后顺序扫描即可进行统计，而所有整体位于上（下）方一行的矩形也可用类似方法进行统计。对于四个角块上被重复统计的矩形总数而言，则可以借助树状数组进行维护。

时间复杂度：$\Theta(q \log^2 q)$

空间复杂度：$\Theta(q)$

67 QTREE

题目大意

给定一棵 n 个节点的环套树的初始边权，要求处理 q 个以下两种操作：

- $f\ u\ v$：将环套树上 u 和 v 之间最短路径上的所有边权全部取相反数；
- $?\ u\ v$：询问环套树上 u 和 v 之间最短路径上的最大连续子段和。

时间限制：2 s

数据规模

- $1 \leq n, q \leq 10^5$。

算法分析

首先考虑本题在链上的简化版本，则易知最大连续子段和的维护只需借助线段树即可实现。对于区间取相反数而言，则只需将线段树中每个节点记录的信息分别维护相应的最大值和最小值即可。

此时，由于本题中维护的信息易于合并，因此我们只需利用树链剖分即可解决树上的情形。最后，为解决环上的问题，我们不妨任选一处将整个环展开成链，则只需再单独借助一棵线段树进行维护即可。

时间复杂度：$\Theta(n + q \log^2 n)$

空间复杂度：$\Theta(n)$

68 QTREE6

题目大意

给定一棵 n 个点的有根树，初始时所有节点均为白色，要求处理 m 个以下两类操作：

- $0\ u$：询问点 u 所在的同色连通块的大小；
- $1\ u$：将点 u 反色。

时间限制：1 s

数据规模

- $1 \leq n, m \leq 10^5$。

算法分析

首先，由于点 u 所在的同色连通块分为上下两部分，而向上延伸的部分较为难以统计。因此，我们考虑每次询问时均找到点 u 所在同色连通块的最高点 v，显然此部分只需借助 Lct 即可实现。于是，接下来我们仅需考虑对所有点 u 维护 $white_u$ 和 $black_u$ 两个值，分别表示从点 u 的子节点向下延伸出的白（黑）色连通块大小的总和，且此处不考虑点 u 的具体颜色。此时，对于询问操作而言，则找到点 v 后即可直接得到答案。对于修改操作，若以白色变为黑色为例，设点 x 是点 u 祖先中从下往上的第一个黑色节点，则对于链 (x, u) 上的所有点 z（不包括点 u），$white_z$ 均需减少 $white_u + 1$。同样地，设点 y 是点 u 祖先中从下往上的第一个白色节点，则对于链 (y, u) 上的所有点 z（不包括点 u），$black_z$ 均需增加 $black_u + 1$。易知，上述所有操作只需在 Lct 上维护懒标记即可解决。

时间复杂度：$\Theta(m \log n)$

空间复杂度：$\Theta(n)$

69 QUERY

题目大意

给定一棵 n 个点的无根树，初始时所有点权均为 0，要求在线支持 m 个下列操作：

- $c\ x\ y\ a\ b$：对树上 x 到 y 之间路径上的相应点权加上一个首项为 a、公差为 b 的等差数列；
- $q\ x\ y$：询问树上 x 到 y 之间路径上所有点的点权和；
- $l\ x$：退回到第 x 次修改操作后的状态。

时间限制：5 s

数据规模

- $1 \leq n, m \leq 10^5$，$0 \leq a, b \leq 10^3$。

算法分析

首先考虑本题的简化版本，即：在链上执行所有操作，且没有切换历史版本的操作。此时，问题相当于对某些区间加上等差数列，以及查询某些区间的权值和。由于所有的修改操作相互独立，并且任意次修改操作均可合成为一次等价修改操作，因此只需在线段树上维护懒标记即可解决。于是，因为单个修改操作也可以被分解为任意个等价操作，所以当操作在树上进行时，只需利用树链剖分即可完成链上相应的所有操作。

最后，切换历史版本的操作决定了需要对数据进行可持久化，此处也就是对维护重链的线段树进行可持久化。此时，为了减少可持久化时的空间开销，我们可以利用标记永久化的思想，以避免下传标记所带来的额外空间开销。

时间复杂度：$\Theta(m \log^2 n)$

空间复杂度：$\Theta(m \log^2 n)$

70 RANKA

题目大意

给定一张 9×9 的围棋棋盘，由两名玩家轮流落子，并且允许玩家在某次落子时弃权。要求构造一个长度为 n 的合法对弈序列，其中允许提子，但必须满足“禁止全局同形再现”的规则。

时间限制：1 s

数据规模

- $1 \leq n \leq 10^4$。

算法分析

由于本题中允许玩家在某次落子时弃权，因此我们可以近似任意选取黑白子的数量。于是，我们利用提子进行如下构造：

- 落子位置从 $(1,1)$ 开始以行优先的方式循环遍历；
- 黑白双方每 80 子进行一次交替，即先由黑方从 $(1,1)$ 起连续下 80 手，然后由白方从 $(9,9)$ 起连续下 80 手，再由黑方从 $(9,8)$ 起连续下 80 手，以此类推。

不难发现，上述对弈序列中每 80 手即恰有一次提子，并且黑白方交替进行提子。因此，在“禁止全局同形再现”的规则下，上述对弈序列共可产生 $79 * 81 * 4$ 种不同状态，已经可以满足本题要求。

时间复杂度：$\Theta(n)$

空间复杂度：$\Theta(1)$

71 REALSET

题目大意

给定一个 n 维向量 $\vec{a}=(a_0,a_1,\ldots,a_{n-1})$，定义 $\vec{a}$ 是优美的，当且仅当存在 n 维非零向量 $\vec{b}$，使得 $\vec{a}\cdot\vec{b_i}=0$ 对任意 $0\le i<n$ 成立，其中 $\vec{b_i}=(b_i,b_{(i+1)\bmod n},\ldots,b_{(i+n-1)\bmod n})$。要求判定给定的 $\vec{a}$ 是否是优美的。

时间限制：8 s

数据规模

- $1\le t\le 100$，$1\le n\le 3\times 10^4$，$-10^3\le a_i\le 10^3$。

算法分析

由于 $\vec{a}\cdot\vec{b_i}$ 对任意 $0\le i<n$ 均成立，因此本题等价于判定下列 $n\times n$ 线性方程组是否存在不全为零的非平凡解：

$$\begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ a_{n-1} & a_0 & \cdots & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \cdots & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

此时，由于零向量必然是上述方程组的一组解，则要使得存在不全为零的非平凡解，则必有其系数矩阵行列式为 0，即其秩小于 n。同时，由于循环矩阵的特殊性质，其秩等价于 $n-d$，其中 d 为多项式 $gcd(f(x),x^n-1)$ 的次数，且有 $f(x)=\sum_{i=0}^{n-1}a_ix^i$ 为系数矩阵的相关多项式。因此，利用分圆多项式的特殊性质，要判定该循环矩阵是否奇异，等价于判定是否存在某个 $d|n$，使得 x^d-1 可以整除 $f(x)(x^{\frac{d}{p_1}}-1)(x^{\frac{d}{p_2}}-1)\cdots(x^{\frac{d}{p_k}}-1)$，其中 $p_1,p_2,\ldots,p_k$ 为 d 的所有质因子。于是，我们只需对 n 的所有因数进行判定即可。

时间复杂度：$\Theta(\tau(n)n)$

空间复杂度：$\Theta(n)$

72 RIN

题目大意

给定 n 项课程以及 m 个学期的课表，其中 $x_{i,j}$ 表示在第 j 个学期选修第 i 项课程的收益，若为 -1 则表示该学期不能选修该项课程。给定 k 组约束条件 (a_i, b_i)，表示第 a_i 项课程必须先于第 b_i 项课程选修。要求给出一组满足所有约束条件的课程选修安排，恰好选修每项课程各一次，并且最大化所有课程的平均收益。

时间限制：1 s

数据规模

- $1 \leq n, m \leq 100$，$0 \leq k \leq 100$，$-1 \leq x_{i,j} \leq 100$。

算法分析

首先，为避免特殊处理 $x_{i,j} = -1$ 的情形，则我们不妨直接强制令 $x_{i,j} = -\infty$。此时，由给定约束条件的形式不难想到最大权闭合子图模型：

- 考虑将所有二元组 (i, j) 均视作为点，并且记差分序列 $v_{i,j} = x_{i,j} - x_{i,j-1}$ 为其对应的点权，则显然应连接所有边 $< (i, j), (i, j-1) >$，表示选取 (i, j) 左侧的所有点；
- 对于任意一组约束条件 (a_i, b_i) 而言，我们不妨等价表述为：若选取 (a_i, j) 则必须选取 $(b_i, j+1)$。于是，我们同样可以按上述方式连边。

借助上述建立的模型，则我们只需直接套用最大权闭合子图模型即可解决本题。

时间复杂度：$\Theta(MaxFlow(nm, (n+k)m))$

空间复杂度：$\Theta((n+k)m)$

73 RNG

题目大意

给定 k 阶线性常系数齐次递推关系 $a_i = \sum_{j=1}^{k} c_j a_{i-j} \pmod{104857601}$（$i > k$）的递推系数 c_j 以及初值 a_j（$1 \le j \le k$），要求计算 a_n 的值。

时间限制：15 s

数据规模

- $1 \le k \le 3 \times 10^4$，$1 \le n \le 10^{18}$，$0 \le a_j, c_j < 104857601$。

算法分析

由于本题中给定的线性常系数齐次递推关系具有特征多项式 $f(x) = x^k - \sum_{i=1}^{k} c_i x^{k-i}$，于是易知 $a_n = \sum_{i=1}^{k} [x^{i-1}] g(x) a_i$，其中 $g(x) \equiv x^{n-1} \pmod{104857601, f(x)}$，因此我们可以借助倍增 NTT 并结合多项式除法来解决。

时间复杂度：$\Theta(k \log k \log n)$

空间复杂度：$\Theta(k)$

74 ROC

题目大意

给定一个由 $n \times m$ 的矩阵描述的房间，整个房间是一个完全被墙体包围的四连通块，所有墙体均平行于坐标轴。已知房间内部每个 90° 内角处均恰好站有一个唯一标号的小朋友，小朋友的移动速度均为每单位时间内移动一单位距离，每次交换只能发生在相邻 90° 内角处的两个小朋友之间，且只能紧贴墙体内侧移动，但同一时刻允许多对不相交的小朋友同时进行交换。给定 t 组相互独立的询问，每次询问一对小朋友最快需要多久才能相遇。

给定的房间保证不存在无效墙体，并且额外满足：同一行中所有位于房间内的空地保证连续。

时间限制：1 s

数据规模

- $5 \leq n \leq 2500$，$3 \leq m \leq 2500$，$1 \leq t \leq 10^4$。

算法分析

不难发现，由于整个房间为一个四连通块，且不存在无效墙体，因此若按顺（逆）时针序紧贴墙体内侧移动，则必然可以将所有墙角的小朋友依次连接成一个环。于是，我们不妨将小朋友视为点，任意相邻的小朋友之间连接一条边权为在房间内两者间移动距离的边，则所有询问均可以借助此图解决。

构图

要通过给定的矩阵构出此图，则通常方法便是模拟整个移动过程。假定我们以逆时针方向紧贴墙体内侧移动，则一个合法的状态应当满足：

- 面朝当前移动方向，右侧紧靠墙体；
- 面朝当前移动方向，前方紧邻空地。

为了保证始终处于合法状态，则每次移动后均需调整移动方向：

- 若此时右侧为空地，则移动方向应向右旋转 90°；
- 若此时右侧仍然为墙体，且前方为空地，则保持移动方向不变；
- 否则，不断将移动方向向左旋转 90°，直到前方出现空地为止。

易证，按上述过程必能恰好紧贴墙体内侧移动一周，而遗留的问题便是如何确定所有小朋友的相对位置。由于房间内可能存在宽度为 1 的通道等特殊情形，因此某些位置在整个移动过程中可能被经过不止一次，但每个小朋友显然只应出现一次。于是，我们应当在移动过程中随时结合移动方向来判定当前是否处于墙角，以确定小朋友的相对位置。

询问

由于题中保证同一行中所有房间内的空地都连续，所以墙角总数不会超过 $2n$，即小朋友总数也不超过 $2n$。因此，对于每个询问都暴力枚举相遇点的代价也是可以承受的。同时，若将单个询问分解为顺时针和逆时针两种情形处理，则易证相遇时间关于相遇点具有单峰性，且相遇点越接近总路程的中点相遇时间越短，因此我们也可以通过二分将单次询问的代价降至 $\Theta(\log n)$。

本题综合而言是一道细节较多的模拟题，具体实现时只需注意初始方向的选取、墙角的判定等问题即可。

时间复杂度：$\Theta(nm + t \log n)$

空间复杂度：$\Theta(nm)$

75 SEAARC

题目大意

给定依次排列在一条直线上的 n 个点，第 i 号点具有颜色 a_i。任意一对同色点之间都恰好连了一条圆弧，求有多少对异色圆弧相交。

时间限制：5 s

数据规模

- 对于 10% 的数据，$n \leq 100$；
- 对于 40% 的数据，$|c| \leq 100$；
- 对于 100% 的数据，$1 \leq n \leq 10^5$，$1 \leq |c|, a_i \leq 10^5$。

算法分析

为了方便之后的讨论，不妨记 cnt_i 为第 i 种颜色的总点数，记 $pre_{i,j}$ 为前 i 个点中具有颜色 j 的点数。由于题中所求的相交圆弧情形比较难以处理，因此我们尝试补集转化思想：记所有异色圆弧对总数为 s，形如 $AABB$ 的异色圆弧对数为 s_1，形如 $ABBA$ 的异色圆弧对数为 s_2，形如 $ABAB$ 的异色圆弧对数为 s_3，则易知 $s = s_1 + s_2 + s_3$，即 $s_3 = s - s_1 - s_2$，而 s_3 即为所求。下面依次讨论 s、s_1 和 s_2 的求解。

异色圆弧对总数 s

由于已知每种颜色的点数 cnt_i, 则第 i 种颜色的圆弧数为 $\binom{cnt_i}{2} = \frac{cnt_i(cnt_i-1)}{2}$，于是 $s = \sum_{i<j} \binom{cnt_i}{2}\binom{cnt_j}{2}$，易知这在线性时间内即可求得。

形如 $AABB$ 的异色圆弧对数 s_1

对于形如 $AABB$ 的异色圆弧对，考虑以 i 作为右侧圆弧的左端点，则其对 s_1 的贡献为 $(cnt_{a_i} - pre_{i,a_i})\sum_{j\neq a_i}\binom{pre_{i,j}}{2}$，其中 $(cnt_{a_i} - pre_{i,a_i})$ 表示 i 右侧与其同色的点数，而 $\sum_{j\neq a_i}\binom{pre_{i,j}}{2}$ 则表示 i 左侧与其异色的圆弧数。借助上式，我们同样可以线性求得 s_1。

形如 $ABBA$ 的异色圆弧对数 s_2

通过一些简单的尝试，我们发现形如 $ABBA$ 的异色圆弧对性质较为复杂，难以只用较小的代价进行统计。因此，我们采取平衡规划的思想，对 cnt_i 以 k 为界进行划分并分别处理。

情形一：$cnt_A \geq k$ 由于满足 $cnt_A \geq k$ 的颜色数最多只有 $\frac{n}{k}$ 种，因此这部分的处理较为简单：对于任意一条颜色异于 A 的圆弧 (x, y)，显然其贡献为 $pre_{x,A}(cnt_A - pre_{y,A})$。因此，若已知所有 $pre_{i,A}$，则对于任意一种不同于 A 的颜色 B，只需在顺次扫描所有具有颜色 B 的点的同时，再利用前缀和加速处理即可。整个过程的总代价为 $\Theta(\frac{n}{k} \cdot n) = \Theta(\frac{n^2}{k})$。

情形二：$cnt_A < k$ 且 $cnt_B \geq k$ 这种情形与上种情形相类似，只不过此时只能枚举颜色 B 作为基础。为了方便之后的表述与计算，对于确定的颜色 A，记 pos_i 为第 i 个具有颜色 A 的点，记 $z = cnt_A$，并简记 $pre_{pos_1,B}, pre_{pos_2,B}, \ldots, pre_{pos_z,B}$ 为 $x_1, x_2, \ldots, x_z$。此时，可以得到颜色对 (A, B) 的贡献为：

$$\begin{aligned}
\sum_{i<j} \binom{x_j - x_i}{2} &= \frac{1}{2}\sum_{i<j}(x_j - x_i)^2 - \frac{1}{2}\sum_{i<j}(x_j - x_i) \\
&= \frac{1}{2}(z-1)\sum_i x_i^2 - \sum_{i<j} x_i x_j - \frac{1}{2}\sum_{i<j}(x_j - x_i) \\
&= \frac{1}{2}(z-1)\sum_i x_i^2 - \frac{1}{2}((\sum_i x_i)^2 - \sum_i x_i^2) - \frac{1}{2}\sum_{i<j}(x_j - x_i) \\
&= \frac{1}{2}z\sum_i x_i^2 - \frac{1}{2}(\sum_i x_i)^2 - \frac{1}{2}\sum_{i<j}(x_j - x_i)
\end{aligned}$$

由于上式中每部分的计算都只需对具有同种颜色的点顺序扫描即可，因此整个过程的总代价也是 $\Theta(\frac{n^2}{k})$。

情形三：$cnt_A < k$ 且 $cnt_B < k$ 当两种颜色的出现次数均较小时，则上述的统计方式都不再适用，于是尝试另一种复杂度与同种颜色点数相关的方式。对于一条具有颜色 B 的圆弧 (x, y)，相应的方案数为颜色异于 B 的所有圆弧 (x', y')，且满足 $x' < x$ 及 $y < y'$。此时，如果我们采取类似于扫描线的方式从小到大枚举左端点 x，然后再枚举右端点 y，则方案数的计算可以借助后缀和求得，而加入一条圆弧的操作也能一并进行。由于此处后缀和的维护需要基于数据结构，因此整个过程的总代价为 $\Theta(nk \log n)$。

由于前两种情形的总代价均为 $\Theta(\frac{n^2}{k})$，而最后一种情形的代价为 $\Theta(nk \log n)$，因此，为了使得整个算法的总复杂度最优，易知应取 $k = \sqrt{\frac{n}{\log n}}$。此时，总复杂度为 $\Theta(n\sqrt{n \log n})$。在实际测试中，当 k 取在 10^2 级别时算法表现较好，更精确的取值则与具体实现有一定关联。

时间复杂度：$\Theta(n\sqrt{n \log n})$

空间复杂度：$\Theta(n)$

76 SEAEQ

题目大意

对于任意两个 $1 \sim n$ 的排列 p_1 和 p_2，定义 $f_{p_1,p_2,e}$ 为使得 $p_1(l \dots r)$ 和 $p_2(l \dots r)$ 相似的无序二元组 (l, r) 的总数。其中，定义两个等长的序列相似，当且仅当两个序列中元素的相对大小关系完全一致，并且要求满足单个序列中的逆序对数至多为 e。给定 t 组相互独立的询问 (n_i, e_i)，求当 p_1 和 p_2 均取遍 $1 \sim n_i$ 的所有排列时，f_{p_1,p_2,e_i} 的总和。

时间限制：1 s

数据规模

- $1 \leq t \leq 10^4$，$1 \leq n \leq 500$，$1 \leq e \leq 10^6$。

算法分析

根据题中对于序列相似的定义，通过一些简单的推导不难发现：

$$Ans = \sum_{i=1}^{n} (n - i + 1) f_{i,e} (\binom{n}{i} (n - i)!)^2$$

其中，求和下标 i 表示区间 $[l, r]$ 的长度，$f_{i,e}$ 表示逆序对数至多为 e 的 $1 \sim i$ 的排列数。易知，仅需 $\Theta(n^3)$ 的代价即可预处理出所有 $f_{i,e}$，因此单次询问代价为 $\Theta(n)$。

时间复杂度：$\Theta(n^3 + tn)$

空间复杂度：$\Theta(n^3)$

77 *SEAND2

题目大意

给定一个十进制正整数 a，要求重新排列 a 的各十进制位，最小化 $f_a = \sum_{i=1}^{n} a \bmod b_i$。

时间限制：5 s

数据规模

- $1 \leq t \leq 100$，$1 \leq a \leq 10^{1000}$，$1 \leq n \leq 100$，$1 \leq b_i \leq 10^6$。

算法分析

由于本题中给定的取模并没有较好的性质，因此我们只能考虑进行随机。此时，若预处理出任意十进制位在模所有 b_i 意义下的等价值，则单次随机调整后仅需 $\Theta(n)$ 的代价即可重新计算 f_a。

上述算法在 $Codechef$ 上可得 $0.143pt$。

时间复杂度：$\Theta(CountRandom \cdot n)$

空间复杂度：$\Theta(n|a|)$

78 SEAORD

题目大意

给定 n 个程序以及两台计算机，已知第 i 个程序需要在两台计算机上分别执行 a_i 和 b_i 单位时间。要求每个程序均在两台计算机上各执行恰好一次，并且必须满足：

- 对于同一台计算机而言，同一单位时间内至多只能执行一个程序；
- 对于同一程序而言，同一单位时间内至多只能在一台计算机上执行。

要求给出一组执行所有程序的合法方案，最小化方案的总用时。

时间限制：2 s

数据规模

- $1 \leq n \leq 10^4$，$1 \leq a_i, b_i \leq 10^5$，$\sum n_i \leq 2 \times 10^5$。

算法分析

首先，易知总用时的下限为 $Ans = \max\{\sum_i a_i, \sum_i b_i, \max_i\{a_i + b_i\}\}$，并且可以证明该下限总是可以达到。其中，若存在某个程序 i 使得 $a_i + b_i = Ans$，则显然可以直接构造出一种合法方案。否则，由于合法方案数较多，因此我们只需多次随机所有程序的执行序列并进行检验即可。

时间复杂度：$\Theta(n)$

空间复杂度：$\Theta(n)$

79 SEINC

题目大意

给定两个长度为 n 的序列 $\{a_i\}$ 和 $\{b_i\}$，每次操作允许任意选定 $1 \le i \le j \le n$，并对所有 $i \le k \le j$ 令 a_k 变为 $(a_k + 1) \bmod 4$，求将序列 $\{a_i\}$ 变为 $\{b_i\}$ 的最少操作次数。

时间限制：1 s

数据规模

- $1 \le t \le 10$，$1 \le n \le 10^5$，$0 \le a_i, b_i \le 3$。

算法分析

若记 c_i 为 a_i 处执行操作的总次数，则易知 $c_i \equiv b_i - a_i + 4$。同时，由贪心策略不难推知，此时必有 $Ans = \sum_{i=1}^{n} max\{c_i - c_{i-1}, 0\}$。为方便接下来的表述，不妨记 $d_i = c_i - c_{i-1}$，则可以证明 $|d_i| < 4$：

- 假设 $d_i \ge 4$，则考虑令 c_i 变为 $c_i - 4$，此时 $max\{d_i, 0\}$ 必然减小 4，而 $max\{d_{i+1}, 0\}$ 的增量至多为 4，因此答案至少不会变劣；
- 假设 $d_i \le -4$，则考虑令 c_i 变为 $c_i + 4$，此时 $max\{d_i, 0\}$ 依然为 0，而 $max\{d_{i+1}, 0\}$ 不可能增加，因此答案也不会变劣。

于是，上述结论表明 d_i 至多只有两种取值。与此同时，若要将某个 d_i 减小 4，则必须将某个 d_j（$1 \le j < i$）增大 4。因此，当 $d_i \le 1$ 时，显然没有减小 d_i 的必要；当 $d_i = 2$ 时，若存在某个 $d_j = -3$（$1 \le j < i$），则可以利用其减小答案；当 $d_i = 3$ 时，则必然优先利用其左侧的 -3，否则再利用其左侧的 -2。于是，我们只需在顺序扫描的同时，统计 -3 和 -2 的个数即可。

时间复杂度：$\Theta(n)$

空间复杂度：$\Theta(n)$

80 SHORT

题目大意

给定 n 和 k，统计有多少有序对 (a,b) 满足 $(a-n)(b-n)$ 能够整除 $ab-n$，并且其中 $n<a,b<k$。

时间限制：8 s

数据规模

- $1\le t\le 5$，$0\le n\le 10^5$，$n<k\le 10^{18}$。

算法分析

首先，当 $n=0$ 时易知，区间 (n,k) 内的任意有序对 (a,b) 都满足条件，因此共有 $(k-1)^2$ 种不同方案。于是，在接下来的讨论中，我们均假设 $n\ge 1$，则有 $ab-n=p(a-n)(b-n)$ $(p\ge 2)$。此时，由于 a 和 b 具有对称性，因此我们不妨假设 $a\le b=n+\frac{n(a-1)}{p(a-n)-a}$，则有 $p\le\frac{a^2-n}{(a-n)^2}$。同时，若令 $d=p(a-n)-a$ 是 $n(a-1)$ 的因数，则有 $d\ge a-2n$，于是化简 $a\le b\le n+\frac{n(a-1)}{a-2n}$ 可得 $a\le 2n+\sqrt{2n^2-n}$。因此，我们可以选取某个合适的阈值 n_0，并且当 $a-n\le n_0$ 时枚举 $n(a-1)$ 的因数进行验证，而当 $a-n>n_0$ 时，我们不妨直接枚举 p 进行验证。

时间复杂度：$\Theta(\tau(n)^2n)$

空间复杂度：$\Theta(n)$

81 SHORT2

题目大意

给定一个质数 p，统计有多少对有序二元组 (a,b)，满足 $a,b>p$ 并且 $(a-p)(b-p)|ab$。

时间限制：3 s

数据规模

- $1\le t\le 5$，$1<p<10^{12}$。

算法分析

为方便接下来的讨论，我们不妨用 a,b 分别替换 $a-p$ 和 $b-p$，则约束条件等价于 $ab|(a+p)(b+p)$，即 $ab|p(a+b+p)$。此时，由于 p 为质数，因此我们以 p 是否能够整除 a,b 为标准分类进行讨论。

a 和 b 同时被 p 整除

在此种情形下，若记 $a=pa'$ 以及 $b=pb'$，则有 $p^2a'b'|p(pa'+pb'+p)$，即 $a'b'|(a'+b'+1)$。于是，由于必须满足 $a'b'\le a'+b'+1$，则 $(a'-1)(b'-1)\le 2$。因此不难验证，无论 p 的取值，此时可行的有序二元组 (a',b') 有且仅有 $(1,1)$、$(1,2)$、$(2,1)$、$(2,3)$、$(3,2)$ 这五组。

a 和 b 中恰有一个被 p 整除

不失一般性，我们不妨假设 $b=pb'$，则 $pab'|p(a+pb'+p)$ 等价于 $ab'|(a+pb'+p)$。于是，若记 $a+pb'+p=kab'$，则有 $b'=\frac{a+p}{d}$，其中 $d=ka-p$，即 $k=\frac{d+p}{a}$。此时，为使得 b' 和 k 为整数，则必须满足 $d|(a+p)$ 以及 $a|(d+p)$。与此同时，由于 p 不能整除 a，则必有 p 不能整除 d。因此，此种情形下的任意有序二元组 (a,d) 与下种情形的所有解一一对应，即该部分解的总数恰为下部分解总数的两倍。

a 和 b 均不被 p 整除

首先考虑 $a=b$ 的情形，则 $a^2|(a+p)^2$ 等价于 $a|(a+p)$，即 $a=gcd(a,a+p)=gcd(a,p)=1$，因此 $(1,1)$ 必然为一组合法解。

接下来我们不失一般性假设 $a<b$，由于 p 为质数，因此 a,b 和 p 均互质，则 $ab|p(a+b+p)$ 等价于 $ab|(a+b+p)$。若记 $a+b+p=kab$，则有 $b=\frac{a+p}{d}$，其中 $d=ka-1$，即 $k=\frac{d+1}{a}$，于是必须满足 $d|(a+p)$ 以及 $a|(d+1)$。与此同时，由 $ab\le a+b+p$ 可以推知 $(a-1)(b-1)\le p+1$，而 $a<b$ 则进一步表明 $a\le 1+\sqrt{p+1}$。若令 $\omega=1+\sqrt{p+1}$，则有 $a+p\le p+\omega$。此时，由于 $bd=a+p$，因此 $b\le\sqrt{p+\omega}$ 和 $d\le\sqrt{p+\omega}$ 必然至少成立一个，并且可知 $a\equiv -p\pmod d$ 以及 $a\equiv -p\pmod b$，于是我们只需直接枚举 b 或 d 并进行验证即可。

时间复杂度：$\Theta(\sqrt{p})$

空间复杂度：$\Theta(1)$

82 SHORTCIR

题目大意

给定一个仅由括号和逻辑运算构成的布尔表达式 s，并且保证其中出现的所有 n 个变量互不相同。给定其中每个变量值为真的概率，要求根据短路求值原理自行决定计算顺序，最小化计算次数的期望。

时间限制：3 s

数据规模

- $1 \le t \le 50$，$1 \le |s| \le 3 \times 10^4$，$1 \le n \le 10^3$。

算法分析

首先不难发现，由于所有布尔运算的优先级为严格偏序关系，因此实际可能影响最终期望的只有同层的连续与（或）运算的计算顺序。于是，对于任意布尔表达式 s，不妨记 $P(s)$ 为该表达式最终值为真的概率，记 $E(s)$ 为该表达式最小计算次数的期望，则我们可以对每种运算分别进行处理：

- 非运算：若记 s' 为 s 去除此非运算后剩余的表达式，则易知 $P(s) = 1 - P(s')$ 以及 $E(s) = E(s')$；
- 与运算：若 $s = s_1\ and\ s_2\ and\ \cdots\ and\ s_k$，则显然 $P(s) = \prod_{i=1}^{k} P(s_i)$，而对于 $1 \sim k$ 的任意排列 q 而言，$E(s) = \sum_{i=1}^{k} E(s_{q_i}) \prod_{j=1}^{i-1} P(s_{q_j})$，于是不难证明，将 s_i 按 $\frac{E(s_i)}{1-P(s_i)}$ 从小到大的顺序计算时最优；
- 或运算：若 $s = s_1\ or\ s_2\ or\ \cdots\ or\ s_k$，则显然 $P(s) = 1 - \prod_{i=1}^{k}(1 - P(s_i))$，而对于 $1 \sim k$ 的任意排列 q 而言，$E(s) = \sum_{i=1}^{k} E(s_{q_i}) \prod_{j=1}^{i-1}(1 - P(s_{q_j}))$，于是不难证明，将 s_i 按 $\frac{E(s_i)}{P(s_i)}$ 从小到大的顺序计算时最优。

利用上述结论，则本题只需直接对表达式求值即可。

时间复杂度：$\Theta(|s| + n)$

空间复杂度：$\Theta(|s| + n)$

83 SIGFIB

题目大意

给定 $Fibonacci$ 数列的初值 $f_0 = 0$ 和 $f_1 = 1$，要求计算：

$$\sum_{x+y+z=n} 6xyzf_xf_yf_z \pmod m$$

时间限制：6 s

数据规模

- $0 \le n \le 10^{18}$，$1 \le m \le 10^5$，$\sum m_i \le 10^6$。

算法分析

由于本题中给定的求和式为与 f_i 有关的卷积形式，因此我们不妨考虑利用生成函数来解决。此时，若令 $g_i = i \cdot f_i$，则有：

$$G(x) = x \cdot F'(x) = x \cdot (\frac{x}{1-x-x^2})' = \frac{x(1+x^2)}{(1-x-x^2)^2}$$

于是，本题所求即为 $Ans = [x^n]6G(x)^3$。在经过一系列较为复杂的数学推导后可知：

$$Ans = \frac{(5n^5 + 35n^3 - 4n) \cdot f_n - 36n^2 \cdot l_n}{500}$$

其中，l_i 是初值为 $l_0 = 2$ 和 $l_1 = 1$ 的 $Lucas$ 数列 $l_i = l_{i-1} + l_{i-2}$（$i \ge 2$）。在具体实现时，f_n 和 l_n 的计算均可以利用矩阵乘法加速，而为处理模 m 意义下 500 不存在逆元的情形，我们不妨先在模 $500m$ 意义下进行计算，最后只需再将结果除以 500 即可。

时间复杂度：$\Theta(\log n)$

空间复杂度：$\Theta(1)$

84 *SIMGRAPH

题目大意

给定两张 n 个点的简单无向图，要求分别对其进行重标号，最大化重标号后共有边的条数。

时间限制：4 s

数据规模

- $1 \leq t \leq 5$，$1 \leq n \leq 75$。

算法分析

由于本题中给定的两张图近似于完全随机生成，因此我们可以考虑利用模拟退火算法。在具体实现时，每次随机调整后只需重新统计可能发生改变的贡献值，而无需全部重新统计。

上述算法在 $Codechef$ 上可得 $0.932pt$。

时间复杂度：$\Theta(CountRandom \cdot n)$

空间复杂度：$\Theta(n^2)$

85 *SIMNIM

题目大意

给定 n 个正整数 a_i，保证初始时 $\bigoplus_i a_i = 0$。要求给出一种 n 个数的合法划分方案，使得每组内所有正整数的异或和均为 0，要求在此前提下最大化划分方案中的划分组数。

时间限制：4 s

数据规模

- $t = 10$，$10 \leq n \leq 10^3$，$1 \leq a_i < 2^{60}$。

算法分析

为加快逼近最优解的速度，我们不妨首先贪心地将所有二元、三元线性相关组去除，再对剩余的所有正整数进行搜索。此时，由于 a_i 至多只有 60 个二进制位，因此不难证明最优解中划分出的每组元素个数均不会超过 60。于是，若我们模拟高斯消元的过程，则易知至多只需选取 60 次主元即可找到新的线性相关组，然后对划分出的两个新部分继续分治处理。

上述算法在 $Codechef$ 上可得 $0.549pt$。

时间复杂度：$\Theta(n^2)$

空间复杂度：$\Theta(n^2)$

86 SPMATRIX

题目大意

给定 t 组相互独立的询问 n，要求统计 $n \times n$ 的特殊整数矩阵的总数。其中，定义 $n \times n$ 的整数矩阵是特殊的，当且仅当满足以下所有约束条件：

- $\forall\ 1 \le x \le n$，$a_{x,x} = 0$；
- $\forall\ 1 \le x < y \le n$，$1 \le a_{x,y} = a_{y,x} \le n-2$；
- $\forall\ 1 \le x, y, z \le n$，$a_{x,y} \le \max\{a_{x,z}, a_{z,y}\}$；
- $\forall\ 1 \le i \le n-2$，存在 $1 \le x, y \le n$ 使得 $a_{x,y} = i$。

时间限制：0.666 s

数据规模

- $1 \le t \le 10^5$，$3 \le n \le 10^7$。

算法分析

首先由第三个约束条件不难联想到三角不等式，又由于矩阵元素均为正整数，因此考虑建立一张 n 个点的无向完全图，则对于矩阵元素的所有约束分别等价于：

- 图中不存在自环；
- 任意两点 x 和 y 之间恰好存在一条边权在 $[1, n-2]$ 内的无向边；
- 对于任意三元环 (x, y, z)，环上三条边 (x, y)、(y, z)、(z, x) 的边权最大值不唯一；
- 对任意 $i \in [1, n-2]$，存在边权为 i 的边。

此时，若定义 $G_d = (V, E_d)$ 是由边权不超过 d 的所有边和原图的所有点构成的子图，则易证任意 G_d 必然由若干张不相交的完全子图构成，证明如下：

- 定义点集 V 上的二元关系 $R_d(x, y)$ 为 $a_{x,y} < d$，则易知 R_d 满足自反性、对称性和传递性，因此 R_d 均为等价关系；
- 由 R_d 的传递性可知，G_d 中的任意极大连通分量均为完全图，因此 G_d 必然由若干张不相交的完全子图构成。

于是，我们不妨从大到小依次考虑每种边权 i，则每次去除图中所有边权为 i 的边，表现为将图中的某些完全子图分裂成若干个更小的完全子图，这也等价于不断对点集进行划分。因此，我们尝试建立一张等效图 T，其中所有点构成一条链，链上边权均在 $[1, n-2]$ 内，并且使得 $a_{x,y}$ 恰好为 T 中链 (x, y) 上边权的最大值。易知上述的等效图 T 必然存在，并且满足上述的所有约束条件，因此本题可以再次转化为统计本质不同的等效图 T 的方案数。

由于第四个约束条件的存在，则 T 中必然存在长度为 $[1, n-2]$ 内任意值的边，因此恰有一种边权 k 出现两次，接下来我们分三种情形进行计数。

$k = n - 2$

此时，易知 T 共有 $\frac{n!(n-1)!}{2}$ 种可能的情形。同时，对于 R_{n-2} 而言，由于分裂出的三部分完全等价，因此每种本质不同的方案均有 3! 的重复计数。对其余的任意 R_d 而言，由于每次仅分裂出两个等价部分，因此每次均会产生 2! 的重复计数。于是，此种情形下的每个等价类大小均为 $3! \times 2^{n-3} = 3 \times 2^{n-2}$，于是本质不同的该类方案总数为 $\frac{n!(n-1)!}{3\times 2^{n-1}}$。

$k < n - 2$ 且分居 $n - 2$ 两侧

不难求得，此种情形的总方案数为 $\frac{n!(n-1)!}{3!}(n-3)$。同时，由于 k 分居 $n-2$ 的两侧，因此每次划分均分裂出两个等价部分，于是类似的，本质不同的该类方案总数为 $\frac{n!(n-1)!}{3\times 2^n}(n-3)$。

$k < n - 2$ 且位于 $n - 2$ 同侧

在此种情形下，由于无法判断重复计数的具体情形，因此我们无法像上述两种情形一样直接得出结果。不过，若记 f_n 表示点集大小为 n 时所有本质不同的 T 的方案数，则利用递推关系可知：

$$f_n = \frac{n!(n-1)!}{3\times 2^{n-1}} + \frac{n!(n-1)!}{3\times 2^n}(n-3) + \sum_{i=3}^{n-1} \binom{n}{i}\binom{n-3}{i-2} \cdot f_i \cdot \frac{(n-i)!(n-i-1)!}{2^{n-i-1}}$$

其中，右式中前两项分别为前两种情形的方案数，第三项中的 i 表示与 k 位于 $n-2$ 同侧的点数，而 $\frac{(n-i)!(n-i-1)!}{2^{n-i-1}}$ 则为另一侧的方案数。通过一些简单的化简可得：

$$f_n = \frac{n!(n-1)!}{3\times 2^n}(n-1) + \frac{n!(n-3)!}{2^{n-1}} \sum_{i=3}^{n-1} \frac{2^i}{i!(i-2)!} f_i$$

此时，若令 $g_n = \frac{2^{n-1}}{n!(n-1)!} f_n$，则有：

$$g_n = \frac{n-1}{6} + \frac{2}{(n-1)(n-2)} \sum_{i=3}^{n-1} (i-1) g_i$$

通过一系列化简处理可得递推式：

$$g_n = \frac{1}{2} - \frac{1}{3(n-1)} + g_{n-1}$$

因此带入初值 $g_3 = \frac{1}{3}$ 计算可得通项为 $g_n = \frac{n}{2} - \frac{h_{n-1}+2}{3}$，其中 h_i 为调和数列的前 i 项和。于是，最后可以推知通项 $f_n = \frac{n!(n-1)!}{2^{n-1}}(\frac{n}{2} - \frac{h_{n-1}+2}{3})$，则通过预处理即可 $\Theta(1)$ 回答每次询问。

时间复杂度：$\Theta(n+t)$

空间复杂度：$\Theta(n)$

87 *STEPAVG

题目大意

给定 n 个正整数 a_i，每次操作要求选取其中任意两个元素 x 和 y，并用其平均数 $\frac{x+y}{2}$ 替换 x 和 y，即每次操作后将减少一个元素。要求找到一种合法的操作序列，使得最后剩余的元素与给定值 k 尽可能接近。最终评分以剩余元素与 k 之差的绝对值作为参考。

时间限制：5 s

数据规模

- $1 \leq t \leq 10$，$1 \leq n \leq 10^3$。

算法分析

本题由于保证数据完全随机，因此我们考虑按倒序进行贪心。若记当前剩余数集中的最小值和最大值分别为 min 和 max，则其平均数 $mid = \frac{min+max}{2}$。此时，假设我们选取某个适当的参考值 Δ，则有：

- 若 $k < mid - \Delta$，则我们认为 k 远小于 mid，此时显然应当增加 min 所占的权重。因此，我们考虑最后再对 min 进行操作，则剩余所有数的操作结果应尽可能接近 $k' = 2k - min$；
- 若 $k > mid + \Delta$，则我们认为 k 远大于 mid，此时显然应当增加 max 所占的权重。因此，我们考虑最后再对 max 进行操作，则剩余所有数的操作结果应尽可能接近 $k' = 2k - max$；
- 否则，由于 mid 已经较为接近 k，因此我们不妨直接合并 min 和 max。

通过适当调整 Δ 值可以发现，上述方法当 Δ 的取值在 $\frac{max-min}{10}$ 附近时所得解较优。

上述算法在 $Codechef$ 上可得 $1.000pt$。

时间复杂度：$\Theta(n \log n)$

空间复杂度：$\Theta(n)$

88 STREETTA

题目大意

给定两个长度均为 n 的序列 $\{s_i\}$ 和 $\{t_i\}$，要求处理 m 次以下三种操作：

- $1\ u\ v\ a\ b$：对于 $[u,v]$ 内的所有 s_i，对首项为 b、公差为 a 的等差数列的相应项取最大值；
- $2\ u\ v\ a\ b$：对于 $[u,v]$ 内的所有 t_i，加上首项为 b、公差为 a 的等差数列的相应项；
- $3\ i$：查询 $s_i + t_i$ 的值。

时间限制：3 s

数据规模

- $1 \le n \le 10^9$，$1 \le m \le 3 \times 10^5$。

算法分析

首先不难发现，由于本题中的两个序列相互独立，因此显然可以对两个序列分别进行处理。此时，对于第二类修改操作而言，易知只需借助线段树即可解决，且维护的懒标记可以直接进行合并。而对于第一类修改操作而言，同样考虑维护类似的懒标记 (a,d)，分别表示首项和公差，则当合并两个标记 (a_1,d_1) 和 (a_2,d_2) 时：

- 首先易知，单个等差数列必然在平面直角坐标系中呈直线分布，则两个有限项的等差数列等价于两段线段；
- 若 (a_1,d_1) 和 (a_2,d_2) 所对应的线段相离，则此时必有其中一个等差数列恒大于另一个等差数列的对应项，于是仅需保留其中较大的一个等差数列作为新标记即可；
- 否则，由于两条线段仅有一个交点，因此该节点必然存在一棵子树，使得在这棵子树中其中一个等差数列恒大于另一个等差数列的对应项。于是，我们不妨将该等差数列作为该节点的新标记，而另一个等差数列显然只会对另一棵子树产生影响，则此部分只需递归处理即可。

利用上述维护方式，则维护 $\{t_i\}$ 的单次代价显然为 $\Theta(\log n)$。对于维护 $\{s_i\}$ 而言，由于一段连续区间对应于线段树上 $\Theta(\log n)$ 个节点，同时标记的合并至多只会递归执行 $\Theta(\log n)$ 层，于是可知单次代价为 $\Theta(\log^2 n)$。

时间复杂度：$\Theta(m \log^2 n)$

空间复杂度：$\Theta(m \log n)$

89 STRQUERY

题目大意

给定初始字符串 s_{init}，要求处理 q 次下列操作：

- $INSERT\_LEFT\ c$：在当前字符串开头插入字符 c；
- $INSERT\_RIGHT\ c$：在当前字符串结尾插入字符 c；
- $INSERT\_MIDDLE\ c$：在当前字符串中点插入字符 c；
- $DELETE\_LEFT$：删除当前字符串开头字符；
- $DELETE\_RIGHT$：删除当前字符串结尾字符；
- $DELETE\_MIDDLE$：删除当前字符串中点字符；
- $QUERY\ t$：询问字符串 t 在当前字符串中的出现次数。

时间限制：1 s

数据规模

- $|s_{init}| = 10$，$1 \le q \le 1.5 \times 10^5$，$\sum |t_i| \le 1.5 \times 10^6$。

算法分析

本题是一道经典的后缀平衡树问题。首先易知，单棵后缀平衡树可以支持在串头动态插入/删除字符，并且维护其所有后缀字典序的相对大小关系，则查询操作只需在后缀平衡树中二分即可。

接下来，若要求同时支持串头和串尾的修改操作，则我们考虑将整个字符串从某处分割为左右两部分。于是，我们对分割出的两部分字符串分别借助一棵后缀平衡树进行维护，并且所有涉及串头的修改操作仅在左侧的后缀平衡树中进行，而所有涉及串尾的修改操作相应的仅在右侧的后缀平衡树中进行。同时，若某个时刻其中某侧字符串为空串，则不妨重新均分整个字符串并暴力重构两侧的后缀平衡树，易证重构操作涉及的总串长为线性。此时，对于询问操作而言，整体位于某侧字符串中的方案数显然可以通过在后缀平衡树中二分求得，而横跨分割处的方案数则只需借助 Kmp 即可解决。

最后，考虑对中点处修改操作的支持，则我们不妨始终将整个字符串均分为 s_l 和 s_r 两部分，使得 s_l 和 s_r 的分割处即为整个字符串的中点。于是，所有的修改操作仅可能位于 s_l 或 s_r 的两端，因此只需分别借助两棵后缀平衡树维护 s_l 和 s_r 即可。同时，由于单次修改操作对串长的影响至多为 1，即 s_l 和 s_r 的分割处至多移动距离为 1，因此移动分割处的代价也是可以承受的。对于询问操作而言，则我们只需采用类似于单个字符串中的处理方式即可。

时间复杂度：$\Theta((q + \sum |t_i|) \log q)$

空间复杂度：$\Theta(q + \sum |t_i|)$

90 TAPAIR

题目大意

给定一张 n 个点 m 条边的简单无向连通图，求有多少对无序二元组 (i,j)，使得在原图中删去编号为 i 和 j 的边后，整张图不再连通。

时间限制：3 s

数据规模

- $1 \le n \le 10^5$，$1 \le m \le 3 \times 10^5$。

算法分析

首先，所有无向边均可以分为割边与非割边两类，易知当删除的两条边中至少包含一条割边时，新图必然不连通。因此，下面我们着重讨论删除两条非割边的情形。

接下来，假设 T 是原图的一棵生成树，则所有非割边被分为树边与非树边两类，此时存在以下三种情形：

- 若 i 和 j 均为非树边，则删除 i 和 j 后新图必然依旧连通；
- 若 i 和 j 均为树边，易知此时 t 被分为三部分。由于 i 和 j 均为非割边，则要使得新图不连通，必须满足覆盖 i 和 j 的非树边集合完全一样。此处，定义非树边 (u,v) 覆盖树边 i，当且仅当树边 i 位于树上从 u 到 v 的路径上；
- 若 i 和 j 一条为树边另一条为非树边，不妨假设 i 为树边，则要使得新图不连通，必有 j 是覆盖 i 的唯一非树边。

综合上述分析，若定义覆盖集合 s_i 为覆盖 i 的所有非树边所构成的集合，其中非树边 j 的覆盖集合 $s_j = \{j\}$，则易证删边方案 (i,j) 可行当且仅当 $s_i = s_j$。同时，为统一删除割边的情形，可知此时必有 $s_i = \varnothing$ 或 $s_j = \varnothing$。

最后，此处有多种途径来解决这一问题，其中较为简便的实现方法是利用哈希。考虑为每条非树边生成一个 $64bit$ 的随机数，然后定义某个覆盖集合的权值为集合中所有边权值的异或和。此时，若利用 Dfs 树作为原图的生成树，则整个过程只需一次遍历即可解决。

时间复杂度：$\Theta(n+m)$

空间复杂度：$\Theta(n+m)$

91 TICKETS

题目大意

给定 n 道互不相同的菜肴，以及 m 位顾客的需求 (a_i, b_i)，即表示第 i 位顾客仅接受第 a_i 或 b_i 道菜肴。要求最大化接待的顾客数 k，但必须满足任意 k 位顾客都可以同时被接待。

时间限制：8 s

数据规模

- $1 \leq t \leq 15$，$2 \leq n \leq 200$，$0 \leq m \leq 500$。

算法分析

由于每位顾客的需求仅为两道菜肴，因此不妨将菜肴视作点而将顾客视作无向边。于是不难证明，最大接待顾客数至少为 k 的充要条件为：对于任意满足 $|E| \leq k$ 的边集 E 而言，必有 $|E| \leq |V|$，其中 V 为与其邻接的点集。

此时，易知我们需要找到某个最小的边集 E 使得 $|E| = |V| + 1$，则有 $k = |E| - 1$。同时，由于此情形下的 V 中必然不可能存在度为 1 的点，于是所有可能的情形仅有以下三种：

- 相异的两点 s 和 t 被恰好三条不相交路径连接；
- 两个不相交的简单环被一条简单路径连接；
- 两个边不相交的简单环恰好交于同一点。

可以发现，在上述任意一种情形中，重复经过同一点其实并不影响最终答案。因此，上述三种情形均可以在 Bfs 树的基础之上较为轻易地解决。

时间复杂度：$\Theta(n^2 m)$

空间复杂度：$\Theta(n + m)$

92 TKCONVEX

题目大意

给定 n 根长度为 a_i 的木棍，要求给出任意一组合法方案，选取 $2k$ 根木棍，使得这 $2k$ 根木棍恰能拼成两个严格凸 k 边形。

时间限制：3 s

数据规模

- $2k \le n \le 10^3$，$3 \le k \le 10$，$1 \le a_i \le 10^9$。

算法分析

首先，不妨将木棍按长度从小到大排列，则易知若存在一组合法方案，则必然存在一组合法方案属于以下两种情形之一：

- 情形一：拼成两个凸 k 边形的木棍分别为序列 $\{a_i\}$ 中的连续 k 根；
- 情形二：选取的所有 $2k$ 根木棍为序列 $\{a_i\}$ 中的连续 $2k$ 根。

与此同时，长度为单调不减序列 $\{a_i\}$ 的木棍无法构成一个严格凸 k 边形，当且仅当对任意 $i \ge k$ 均满足，$a_i \ge \sum_{j=1}^{k-1} a_{i-j}$。由于单根木棍长度存在上限，因此当 n 不小于某个阈值 n_0 时，任意长度为 n 的序列 $\{a_i\}$ 必能构成一个严格凸 k 边形。于是，当 $n \ge n_0 + k$ 时，长度为 n 的序列 $\{a_i\}$ 必能构成两个严格凸 k 边形。

由于阈值 n_0 较小，因此当 $n < n_0 + k$ 时，我们只需暴力处理满足情形二的所有方案即可，该部分代价为 $\Theta(\binom{2k}{k} n_0)$。

时间复杂度：$\Theta(n + \binom{2k}{k} n_0)$

空间复杂度：$\Theta(n)$

93 TMP01

题目大意

给定一个初始时为空的字符串 s，要求处理 q 次以下两种操作：

- 在当前 s 的串尾插入一个给定字符；
- 删除当前 s 串头的字符。

要求在每次操作后统计当前 s 本质不同的子串总数。

时间限制：3 s

数据规模

- $1 \leq q \leq 10^6$。

算法分析

首先不难发现，本题显然可以借助后缀数组来离线解决。但是，由于本题中操作数较多，因此借助后缀数组的算法需要大量的常数优化方可通过本题。同时，我们也可以考虑直接利用Ukkonen 算法在线解决本题。

时间复杂度：$\Theta(q)$

空间复杂度：$\Theta(q)$

94 TREECNT2

题目大意

给定一棵 n 个点的带权无根树，要求处理 q 组询问，其中第 i 组询问将第 a_i 条边的边权修改为 c_i。要求在每次修改操作后计算当前树上权值为 1 的简单路径条数，其中路径的权值定义为路径上所有边权的最大公约数。

时间限制：2 s

数据规模

- $1 \leq n \leq 10^5$，$0 \leq q \leq 100$，$1 \leq c_i \leq 10^6$。

算法分析

由于本题涉及到最大公约数，因此我们考虑利用莫比乌斯反演，则 $Ans = \sum \mu_i f_i$，其中 f_i 为树上权值为 i 的整倍数的路径条数。于是，接下来我们考虑分别计算所有 f_i。此时，若记 T_i 为仅由边权为 i 的整倍数的边构成的森林，则 f_i 的初值显然可以借助并查集维护连通块大小求得。为处理询问操作，则我们不妨将所有不涉及修改操作的边预先加入到并查集中，然后对于剩余的所有边则根据当前边权决定是否加入并查集。其中，由于并查集无法高效地支持删除操作，因此我们必须在每次询问操作后暴力复原并查集。

时间复杂度：$\Theta(c + (n + q^2)\tau(c) \log n)$

空间复杂度：$\Theta(c + \tau(n)n)$

95 TRIPS

题目大意

给定一棵 n 个点的无根树，并且保证边权仅可能为 1 或 2。给定 m 组询问 (s_i, f_i, p_i)，其中 p_i 表示单次移动的距离上限，要求计算从 s_i 移动到 f_i 所需的最少移动次数。

时间限制：8 s

数据规模

- $1 \leq n, m \leq 10^5$。

算法分析

首先考虑单次移动，则显然只需利用倍增即可解决。此时，由于本题中边权仅可能为 1 或 2，因此易知移动次数与 p_i 关系较大。于是，我们不妨考虑对 p_i 进行平衡规划：

- 若 $p_i > \sqrt{n}$，则移动次数显然是 $\Theta(\sqrt{n})$ 级别的，因此我们仅需利用倍增暴力进行每次移动即可；
- 若 $p_i \leq \sqrt{n}$，则同样考虑利用倍增预处理出每个点向上进行 2^k 次距离上限为 p_i 的移动后所处位置，于是此种情形下的询问即可 $\Theta(\log n)$ 处理。

时间复杂度：$\Theta((n+q)\sqrt{n}\log n)$

空间复杂度：$\Theta(n \log n)$

96 TSUBSTR

题目大意

给定一棵 n 个节点的字典树，对于其中任意一对节点 (x, y)，若 x 是 y 的祖先，则树上 x 到 y 路径上的所有字符构成该棵字典树的一个子串。要求统计给定的字典树不同子串的总数，并给出 q 组询问，每组询问给定所有字符间的全序关系 p_i 以及 k_i，要求给定字典树中的第 k_i 小子串。

时间限制：4 s

数据规模

- $1 \leq n \leq 2.5 \times 10^5$，$1 \leq q \leq 5 \times 10^4$，$1 \leq k_i < 2^{63}$。

算法分析

本题显然可以直接对给定字典树建立后缀自动机，然后记 f_i 为从后缀自动机中第 i 个节点出发的不同子串数，则给定字典树的不同子串总数即为 f_{root}，其中 $root$ 为后缀自动机中 $Parent$ 树的根。对于每组询问而言，则只需按给定的全序关系沿后缀自动机中的转移边进行 Dfs 即可。

时间复杂度：$\Theta(|\Sigma|(n + \sum |Ans_i|))$

空间复杂度：$\Theta(|\Sigma| \cdot n)$

97 TWOCOMP

题目大意

给定一棵 n 个节点的无根树，以及两个非空的链集合 S_1 和 S_2，其中的每条链均有一定的收益。要求分别从 S_1 和 S_2 中选择若干条链（允许不选择其中某个集合中的任意一条链），使得从 S_1 中选择的任意一条链 a 与从 S_2 中选择的任意一条链 b 在给定的树上均不相交（包括点交和边交）。最大化选择的所有链的收益之和。

时间限制：2 s

数据规模

- 对于 40% 的数据，$n \leq 10^3$，$|S| \leq 100$；
- 对于 100% 的数据，$1 \leq n \leq 10^5$，$1 \leq |S| \leq 700$。

算法分析

首先通过初步分析可知，由于选择链时的限制条件只存在于两个集合之间，因此结合数据范围容易想到二分图模型。于是，不妨将 S_1 中的链视为左点集，S_2 中的链视为右点集，则题中的限制条件表现为某些特定的左点与右点不能同时被选择，收益即可视作相应的点权，那么题目所求即为最大化点权之和。不难发现，上述转化其实就是经典的二分图最大点权独立集模型，其中点数为 $\Theta(|S_1|+|S_2|)$，边数为 $\Theta(|S_1||S_2|)$。此问题只需借助相关网络流算法即可解决，在此便不再赘述。

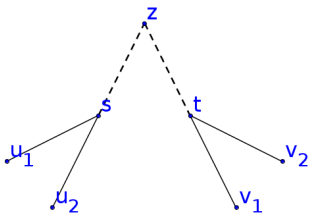
于是，遗留的问题便是如何求得左右点集之间的连边状况。对于 40% 的数据而言，每次简单地暴力扫描的代价也是可以承受的。但当数据规模扩大后，这种方法显然过于低效。下面，我们尝试证明：树上的两条链 (u_1, v_1) 与 (u_2, v_2) 相交，当且仅当节点 $lca(u_1, v_1)$ 在链 (u_2, v_2) 上，或者节点 $lca(u_2, v_2)$ 在链 (u_1, v_1) 上。

充分性

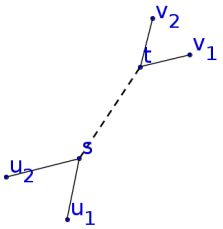
上述结论的充分性应该是显而易见的：当节点 $lca(u_1, v_1)$ 在链 (u_2, v_2) 上时，$lca(u_1, v_1)$ 便是两条链的一个公共点，因此两条链在树上必然相交，另一种情形同理。

必要性

如图，由于树的特性，两条链的所有公共部分必然是像图中 (s, t) 这样连续一段。不妨记 $z = lca(s, t)$，下面分两种情形进行讨论：



图一



图二

- （图一）此时 z 既是 $lca(u_1, v_1)$ 也是 $lca(u_2, v_2)$，于是必要性得证；
- （图二）此时 $z = t$，假设 z 既不是 $lca(u_1, v_1)$ 也不是 $lca(u_2, v_2)$，则链 (z, v_1) 和链 (z, v_2) 都必须向上延伸经过 $father_z$，但这表明 $father_z$ 也是两条链的公共部分，矛盾。

借助上述结论，则每次询问只需分别判断 $lca(u_1, v_1)$ 与 (u_2, v_2)，以及 $lca(u_2, v_2)$ 与 (u_1, v_1) 的相交情况即可，此处可以通过 Dfs 序、倍增等多种途径完成，实现均较为简单。

时间复杂度：$\Theta(MaxFlow(|S_1| + |S_2|, |S_1||S_2|))$

空间复杂度：$\Theta(n \log n + |S_1||S_2|)$

98 TWOROADS

题目大意

给定平面直角坐标系中的 n 个点 (x_i, y_i)，要求任选两条直线，最小化所有点的权值和。其中，定义点 a 的权值为 a 到两条直线上最近点距离的平方。

时间限制：1 s

数据规模

- $3 \leq n \leq 100$，$0 \leq x_i, y_i \leq 10^3$。

算法分析

考虑最优方案中的两条直线，易知其控制区域以夹角的两条角平分线为界，则每条直线各控制相对的两个象限。此时，在不改变点集的划分方案的前提下，通过对两条角平分线进行适当的平移和旋转必然可以使得：

- 其中一条角平分线至少经过两个给定点；
- 另一条角平分线至少经过一个给定点。

因此，可能的划分方案仅有 $\Theta(n^3)$ 种情形。于是，我们不妨在枚举第一条角平分线之后，按照剩余所有给定点在其上投影的位置顺次枚举第二条角平分线，则只需同时维护出两部分点集内的信息即可直接 $\Theta(1)$ 计算出最优的直线拟合方案。

时间复杂度：$\Theta(n^3 \log n)$

空间复杂度：$\Theta(n)$

99 XRQRS

题目大意

给定一个初始时为空的序列，要求处理 m 个下列操作：

- $0\ x$：在序列尾部插入一个数 x；
- $1\ l\ r\ x$：在 $[l,r]$ 内查询与 x 异或值最大的数；
- $2\ k$：删除序列尾部的连续 k 个数；
- $3\ l\ r\ x$：在 $[l,r]$ 内查询有多少个数不超过 x；
- $4\ l\ r\ k$：在 $[l,r]$ 内查询第 k 大的数。

时间限制：1.5 s

数据规模

- $1 \le m \le 5 \times 10^5$，$1 \le x \le 5 \times 10^5$。

数据规模

由于本题中所有插入/删除操作均只会对序列尾部产生影响，因此显然只需将权值线段树可持久化即可。对于查询异或值最大的数而言，由于需要按位进行贪心，因此不妨以 2 的整次幂作为线段树中节点所表示的区间长度，则左右子树即为按该位的 $0/1$ 状态进行划分。

时间复杂度：$\Theta(m \log x)$

空间复杂度：$\Theta(m \log x)$

100 YALOP

题目大意

给定一个 $n \times m$ 的 01 矩阵，其中恰有 k 个给定网格中元素为 1，要求判定是否存在一条起点为 $(1,1)$、终点为 (n,m) 的八连通路径，使得矩阵中元素全部变为 0。其中，对于该条路径中的每一步 $(i,j) \to (i',j')$，网格 (i',j') 以及与其相邻的四个网格中的元素均被取反。

时间限制：5 s

数据规模

- $1 \le t \le 50$，$1 \le n,m \le 10^9$，$\min\{n,m\} < 40$，$0 \le k \le \min\{nm, 10^4\}$。

算法分析

由于本题与 $Lights\ Out$ 游戏十分相似，因此我们尝试利用八连通路径进行移动，以实现与原游戏中类似的作用。接下来，我们分两种情形对本题进行讨论，并且不失一般性假设 $n \le m$。

$n, m \ge 2$

在此种情形下可以推知，无论当前所处网格位置 (i,j)，总是存在一条八连通路径可以到达与其相邻的任意网格，并且所有网格的状态均不改变。下面以证明向右移动一步为例：

$$\begin{bmatrix} (i,j) & (i,j+1) \\ (i+1,j) & (i+1,j+1) \end{bmatrix}$$

此时，易知八连通路径 $(i,j) \to (i+1,j) \to (i+1,j+1) \to (i,j+1) \to (i+1,j) \to (i+1,j+1) \to (i,j+1)$ 符合要求。同时，由于当 $n < 40$ 时循环节长度较短，因此我们只需直接利用 $Lights\ Out$ 游戏中的方法消元即可解决。

$n = 1$

当 $n = 1$ 时，则问题简化为一维情形，但不再可以随意移动。此时，易知八连通路径 $i \to i+1 \to i+2 \to i+1 \to i+2$ 实现了向右移动两步，并且所有网格的状态均未改变。因此，借助该操作显然可以将矩阵中所有元素 1 都转移到最后两个网格上，于是只需再结合 m 即可判定是否存在可行方案。

时间复杂度：$\Theta(k + n^3)$

空间复杂度：$\Theta(k)$