# IOI2016中国国家集训队作业
# 解题报告

绍兴一中 任之洲

# Contents

# Chapter 1

# 非challenge型试题

## 1.1 Future of draughts

### 【试题来源】

Codechef AUG 15 CLOWAY

### 【试题大意】

给出$T$张无向图，第$k$张无向图的点数为$N_k$，边数为$M_k$。

有$Q$组询问，每次询问取出$[L_i, R_i]$区间内的图进行游戏，游戏的回合数要求在$[1, K_i]$范围内，游戏的规则如下：

- 游戏开始时，在每张图上都选取一个起点。
- 每一回合，选取一个图的非空子集，将那些图中的位置沿着任意一条边移动一次。
- 游戏结束时，所有图中都必须回到起点。

对于每组询问，求游戏有多少种进行方案，答案对$10^9 + 7$取模。

数据范围：$1 \leq T, N_k \leq 50$，$0 \leq M_k \leq \frac{N_k(N_k-1)}{2}$，$1 \leq Q \leq 2 * 10^5$，$1 \leq K_i \leq 10^4$

### 【算法介绍】

先考虑一张的情况，设$H$为这张图的邻接矩阵，那么一次长度为$K$回合的游戏的方案数为矩阵$H^K$主对角线上的权值之和，设

$$tr(H^K) = \sum_{i=1}^{N} H^K[i][i]$$

为了后续的计算，需要求出$tr(H) \sim tr(H^K)$。

设矩阵$H$的特征多项式为$P(x)=\sum_{i=0}^{N} c_i x^i$，由Cayley-Hamilton定理得

$$
\begin{aligned}
c_0 I + c_1 H + c_2 H^2 + ... + c_N H^N &= 0 \\
c_0 H^{k-N} + c_1 H^{k-N+1} + c_2 H^{K-N+2} + ... + c_N H^k &= 0 \\
tr\left(c_0 H^{k-N} + c_1 H^{k-N+1} + c_2 H^{K-N+2} + ... + c_N H^k\right) &= 0 \\
c_0 tr(H^{k-N}) + c_1 tr(H^{k-N+1}) + c_2 tr(H^{K-N+2}) + ... + c_N tr(H^k) &= 0
\end{aligned}
$$

由此可以得到一个$tr(H^K)$的递推关系式，矩阵$H$的特征多项式可以根据其定义用插值求出，计算前$K$项的时间复杂度为$O(N^4+NK)$。

考虑多张图同时游戏的情况，设$G(K)$为可以选空集合的进行恰好$K$回合游戏的方案数，那么有

$$G(K)=\prod_{k=L_i}^{R_i} tr((H_k+I)^K)$$

由二项式反演容斥可得不能选空集合的方案数$F(K)$

$$
\begin{aligned}
F(K) &= \sum_{k=0}^{K} (-1)^{K-k} \binom{K}{k} G(k) \\
&= K! \sum_{k=0}^{K} \left(\frac{(-1)^{K-k}}{(K-k)!}\right) \left(\frac{G(k)}{k!}\right)
\end{aligned}
$$

这个式子为卷积形式，但由于题中的模数不是传统NTT模数，需要取三个传统NTT模数分别计算后用中国剩余定理合并。

时间复杂度$O(TN^4+TNK+T^2K\log K+Q)$，空间复杂度$O(TN^2+TK+Q)$。

## 1.2 Simple Queries

### 【试题来源】

Codechef AUG 15 DISTNUM

### 【试题大意】

给定一个含$n$个正整数的数组$A$。现有关于它的$Q$个询问，询问有以下五种类型：

- 1 $l$ $r$：令$S$为区间$[l, r]$中不同元素构成的有序集合，求
$$\left(\sum_{1\le i<j<k\le |S|} S_iS_jS_k\right) \ mod\ 10^9+7$$
- 2 $x$ $y$：将下标为$x$的元素赋值为$y$。
- 3 $x$：将下标为$x$的元素从数组中删除。
- 4 $z$ $y$：在下标为$z$的元素之后插入元素$y$，若$z$等于0，则在数组最前端插入。
- 5 $l$ $r$：输出区间$[l, r]$中不同元素个数。

数据范围：$n, Q \le 10^5$，$1 \le A_i < 10^9+7$

### 【算法介绍】

为了简化问题，可以先离线用平衡树把所有元素的位置确定下来，操作3、4就转化为对固定数列的修改。

考虑维护区间内有多少不同的数，可以为每个位置的元素再多维护一个关键字——这个数前一次出现的位置，区间$[l, r]$中的不同元素个数就等价于区间$[l, r]$中前一次出现位置$< l$的数的个数，可以用树状数组套线段树维护。

考虑计算

$$\sum_{1\le i<j<k\le|S|} S_iS_jS_k$$

设$Sum_i=\sum_{j=1}^{|S|}{S_j}^i$，由容斥可得

$$\sum_{1\le i<j<k\le|S|} S_iS_jS_k=\frac{{Sum_1}^3-3Sum_1Sum_2+2Sum_3}{3!}$$

对每种数维护一棵平衡树用来查找每个元素的前驱后继，树状数组套线段树维护不同元素个数，以及不同元素的$Sum_1, Sum_2, Sum_3$，可以$O(\log^2(n+Q))$完成修改和询问。

时间复杂度$O((n+Q)\log^2(n+Q))$，空间复杂度$O((n+Q)\log^2(n+Q))$。

## 1.3 Easy exam

### 【试题来源】

Codechef JULY 15 EASYEX

### 【试题大意】

有一个$K$面的骰子，上面分别写着$1 \sim K$。

给出两个整数$L$和$F$，掷了$N$次骰子，记掷出数字$i$的次数为$a_i$，求$\prod_{i=1}^{L} {a_i}^F$的期望值。

输出在模域2003下的答案。

数据范围：$0 < N, K \leq 10^9$，$0 < L \leq K$，$0 < L * F \leq 50000$，$0 < F \leq 1000$

### 【算法介绍】

设$x_{i,j}$为第$i$次是否掷出了$j$，若掷出则为1，否则为0，那么

$$\prod_{i=1}^{L} {a_i}^F = \prod_{j=1}^{L} \left( \sum_{i=1}^{N} x_{i,j} \right)^F$$

考虑将$\left( \sum_{i=1}^{N} x_{i,j} \right)^F$展开，将会转化为一些由$F$项$x_{i,j}$相乘的项。再考虑将$L$块这样的项相乘，每两块间不能有相同的$x_{i,j}$，假设这$L$中共有$m$种不同的$x_{i,j}$，那么对答案的贡献即为$\frac{1}{K^m}$。

设$g(F, m)$为$F$项$x_{i,j}$相乘中有$m$项不同的$x_{i,j}$的方案数，这个值可以由递推得到。

$$g(F, m) = mg(F-1, m) + g(F-1, m-1)$$

这样计算出方案中的$m$项元素是无序的。

设$f(L,m)$为$L$块相乘后有$m$项不同的$x_{i,j}$的方案数，这个值也可以由递推得到。

$$f(L,m)=\sum_{i=1}^{F}f(L-1,m-i)g(F,i)$$

这样计算出方案中的$m$项元素也是无序的，所以每项对答案的贡献为$f(L,m)*m!*\binom{N}{m}*\frac{1}{K^m}$。

这个式子为卷积形式，可以用倍增FFT计算，卷积长度随倍增次数调整。

考虑当$m\geq 2003$时，该项对答案实际不会产生贡献，所以只要保留前2002项做卷积就行了。

时间复杂度$O(FL\log FL)$，空间复杂度$O(FL)$。

## 1.4 A game on a graph

### 【试题来源】

Codechef JULY 15 HAMILG

### 【试题大意】

两个玩家在用一个在无向图$G$上进行游戏，这个游戏如下进行：

- 先手选择一个起始点，起点放在这个点上。
- 接下来，两个玩家轮流操作，后手先进行操作。
- 轮到每个玩家操作时，他需要沿着一条边移至另一点。
- 不能重复到达同一点。
- 无法操作的玩家将输掉这个游戏。

求先手有多少种选点方式能保证必胜。

数据范围：$1 \le N \le 2000$，$1 \le M \le 10^6$

### 【算法介绍】

对于一个点$v$，在选了这个点后先手有必胜策略必须保证存在一种原图的最大匹配使得点$v$不在这个最大匹配中，那么不管后手怎么操作，先手都可以沿着最大匹配的边移动。

先用带花树算法求出一组最大匹配的可行解，对于那些没有被匹配的点，再尝试用带花树算法寻找增广路。增广路是不可能被找到的，所以将会遍历到所有可以被替换的匹配边，这些被经过点是可以不存在于最大匹配中的。

时间复杂度$O(NM)$，空间复杂度$O(N+M)$。

## 1.5 Chefbook

### 【试题来源】

Codechef JUNE 15 CHEFBOOK

### 【试题大意】

要求构造$N$个非负整数$P_i$以及$N$个非负整数$Q_i$，使其满足$M$个约束

$$S_i \le P_{x_i} - Q_{y_i} + L_i \le T_i$$

并使得$\sum_{i=1}^{M} P_{x_i} - Q_{y_i} + L_i$尽量大，且$P_i, Q_i \in [1, 10^6]$。

保证每一组$(x_i, y_i)$都不同。

数据范围：$1 \le x_i, y_i \le N \le 100$，$1 \le M \le N^2$，$-600 \le L_i \le 600$，$-1000 \le S_i \le T_i \le 1000$

### 【算法介绍】

这是一个线性规划问题，约束可以转化为

$$S_i - L_i \le P_{x_i} - Q_{y_i} \le T_i - L_i$$

考虑费用流建模

- 由$y_i$向$x_i$连流量为$\infty$，费用为$T_i - L_i$的边。
- 由$x_i$向$y_i$连流量为$\infty$，费用为$L_i - S_i$的边。
- $x_i$向汇连边，源向$y_i$连边，流量为它们在所有约束中的出现次数，费用为0。

费用流中有流量的边表示这条约束被卡着限制条件满足，可以列出差分约束的方程用最短路求出一组可行解，而这组可行解对应着一组满足这组残量网络的解。

点规模$O(N)$，边规模$O(N + M)$。

## 1.6 Chef and Balanced Strings

### 【试题来源】

Codechef MAY 15 CBAL

### 【试题大意】

一个串是平衡的，当且仅当这个串中的字符每种都出现偶数次。

对于一个长度为$N$，由小写字母组成的串$P$，有$Q$组询问$L\ R\ type$，求

$$\sum_{\substack{L \le s < e \le R \\ P[s,e]\text{是平衡串}}} (e-s+1)^{type}$$

询问通过某种方式加密，即强制在线。

数据范围：$1 \le N, Q \le 10^5$，$type \in \{0,1,2\}$

### 【算法介绍】

串$P[l,r]$中每种字符出现次数都为偶数，等价于串$P[1,l-1]$和串$P[1,r]$中每种字符的出现次数的奇偶性相同。

考虑用二进制表示串$P$每个前缀中26种字符出现次数的奇偶性，离散化后只有$O(n)$中状态。

考虑对串$P$按位置分块，块大小为$S=\sqrt{N}$，预处理$ans[L'][R'][type]$为第$L'$块到第$R'$块的答案，询问时将块外元素暴力计算贡献。

当$type=0$时，相当于计算平衡子串的数量。

当$type=1$时，可以把$\sum s$和$\sum e$分开统计。

当$type=2$时，可以通过计算$\sum s^2$，$\sum e^2$以及$\sum s \sum e$来完成。

时间复杂度$O((N+Q)\sqrt{N})$，空间复杂度$O(N\sqrt{N})$。

## 1.7 Counting on a directed graph

### 【试题来源】

Codechef MAY 15 GRAPHCNT

### 【试题大意】

对于一张$N$个点$M$条边的有向图，统计无序对$(X, Y)$的对数，要求满足存在一条从1到$X$的路径，以及一条从1到$Y$的路径，使得这两条路径除了点1以外没有公共点。

数据范围：$1 \le N \le 10^5$，$0 \le M \le 5 * 10^5$

### 【算法介绍】

假如两个点$X, Y$没有公共必经点，那么$(X, Y)$就是合法的。

单源必经点关系可以描述为树状结构，即建出dominator tree，问题转化为计算点对$(X, Y)$的数量，满足$X$和$Y$在树上的LCA为1。

计算出dominator tree上，1的所有儿子的子树大小即可完成计算。

时间复杂度$O(M + \alpha N)$，空间复杂度$O(N + M)$。

## 1.8 Black-white Board Game

### 【试题来源】

Codechef APRIL 15 BWGAME

### 【试题大意】

有一个$N \times N$的矩阵，第$i$行区间$[L_i, R_i]$的格子是黑的，剩下的格子是白的。

Fedya和Alex在这个矩阵上玩游戏，Alex需要找到排列$P$，使得$(i, P_i)$都是黑色的，且$P$的逆序对为偶数，而Fedya需要逆序对是奇数。

求出他们谁能找到的排列数较多，或平局。

数据范围：$N \leq 10^5$

### 【算法介绍】

考虑计算两个人排列数的差值，相当于计算一个行列式，矩阵第$i$行$L_i \sim R_i$位置为1，其余位置为0。

考虑用消元法来计算这个行列式，假设现在消到了第$i$行，找出所有$j$满足$L_j = i$，并用最小的$R_j$去消其他行。

设最小的$R_j$为$R_{min}$，$[i, R_j]$消元后的结果为$[R_{min}+1, R_j]$，仍然是一个区间。

用线段树维护所有$i$所对应的$R_j$，消元时只需要删除最小元素后进行线段树合并。可以发现行列式的绝对值$\leq 1$，当绝对值等于1时，只需要考虑交换两行时所乘的-1即可。

时间复杂度$O(N \log N)$，空间复杂度$O(N \log N)$。

## 1.9 Little Party

### 【试题来源】

Codechef APRIL 15 LPARTY

### 【试题大意】

给出$M$个长度为$N$的串，每个串包含最小的$N$个英文字符描述一个集合，每个字符的大小写表示状态。

假如一个集合$S$满足所有满足$S \subseteq T$且$|T| = N$的$T$都在给出的$M$个集合中，那么$S$称为一个基子集。

要求选出最少数量的基子集，使得给出的$M$个集合都至少包含一个被选中的基子集。

数据范围：$1 \leq N \leq 5$，$0 \leq M \leq 1000$

### 【算法介绍】

等价于最小点权覆盖问题，考虑用搜索计算。

先暴力求出所有可以使用的基子集，然后搜索枚举每个基子集是否使用来得到最小值。

- 最优性剪枝，枚举大于等于已知最优解时就不再继续搜索。
- 新选择一个基子集后，这个基子集必须能够覆盖到至少一个还没有被覆盖到的集合。
- 丢弃一个基子集后（即确定不选这个基子集），不能存在一个还没有被覆盖的集合当前只能被这个基子集覆盖。

- 假如存在两个基子集$A$、$B$满足$A \subseteq B$，那么$B$一定不会存在于最优解中，因为$A$一定可以代替$B$且代价更小。
- 二进制压缩状态后，hash记忆化搜索。

时间复杂度$O(2^{2^n})$，空间复杂度$O(2^{2^n})$。

## 1.10 Counting on a Tree

### 【试题来源】

Codechef MARCH 15 TREECNT2

### 【试题大意】

给出一棵$N$个点的有标号带边权无根树，边权范围为$[1, Z]$。

要求计算无序对$(S, T)(S \neq T)$的数量，满足$S, T$两点路径上所有边的边权的最大公约数为1。

给定$Q$次修改$A_i, C_i$，将第$A_i$条边的边权设为$C_i$，每一次修改后也输出无序对的数量。

数据范围：$1 \le N \le 10^5$，$0 \le Q \le 100$，$1 \le Z, C_i \le 10^6$

### 【算法介绍】

设$f_i$为路径边权的最大公约数是$i$的倍数的$(S, T)$数量，由莫比乌斯反演得，需要求$\sum_{i=1}^{Z} \mu(i) f_i$。

先不考虑修改的影响，对于计算$f_i$，只需要保留所有边权是$i$的倍数的边，求出所有连通的$(S, T)$对数即可。可以用并查集实现，每加一条边，将刚连通的两个连通块的大小的乘积累加入答案中。

考虑修改的影响，只要套上CDQ图重构就行了，而且每次需要重构的部分只有和修改有关的那些边，即只有$O(Q)$条。

由于并查集涉及到还原操作，选用按秩合并的版本。

时间复杂度$O(Z \ln Z + 2^k N \log N + 2^k Q \log Q \log N)$[1]，空间复杂度$O(Z + N)$。

[1]这里的$k$为边权的质因子个数，在$[1, 10^6]$范围中最大为7。

## 1.11 Random Number Generator

### 【试题来源】

Codechef MARCH 15 RNG

### 【试题大意】

设

$$A_i = \sum_{j=1}^{k} C_j A_{i-j}\ mod\ 104857601$$

给出数列$A$的前$k$项以及递推系数$C_j$，求$A_N$。

数据范围：$1 \le k \le 30000$，$1 \le N \le 10^{18}$

### 【算法介绍】

假设已经得出了$A_i$与$A_{i-1} \sim A_{i-k}$的递推关系，将这个关系倍增一下，可以得到$A_i$与$A_{i-1} \sim A_{i-2k}$的递推关系。

为了使得相关的项的长度在倍增中不改变，可以用$A_i$与$A_{i-1} \sim A_{i-k}$的递推关系，得到$A_{i-1} \sim A_{i-k}$与$A_{i-k-1} \sim A_{i-2k}$ 的关系，这样就能得到$A_i$与$A_{i-k-1} \sim A_{i-2k}$的关系。

这样不断倍增可以得到一个$O(d^2 \log N)$的算法。

倍增的第一步是一个卷积，可以直接用NTT做到$O(d \log d)$，对于第二部分。

设需要被调整的多项式为$A(x)$，递推系数多项式为$C(x)$，前$d$项对后$d$项的贡献为多项式$B(x)$，即$B(x)$中的系数为该项被消去时的系数，则有在模$x^{d+1}$域下

$$\begin{aligned} B(x) &= A(x) + B(x)C(x) \\ &= \frac{A(x)}{1 - C(x)} \end{aligned}$$

多项式求逆可以$O(d\log d)$完成，而后$d$项可以由$A(x)+B(x)C(x)$得到，这样就能$O(d\log d\log N)$完成递推。

时间复杂度$O(k\log k\log N)$，空间复杂度$O(k)$。

## 1.12 Devu and Locks

### 【试题来源】

Codechef FEB 15 DEVLOCK

### 【试题大意】

给出$N, P, MM$，对于所有$0 \le M \le MM$，求出有多少被$P$整数的$N$位整数（允许前导0），满足各位数字之和不超过$MM$。

答案对998244353取模。

数据范围1：$1 \le N \le 10^9$，$1 \le P \le 50$，$1 \le MM \le 500$

数据范围2：$1 \le N \le 10^9$，$1 \le P \le 16$，$1 \le MM \le 15000$

### 【算法介绍】

设$f[i][j][k]$为长度为$i$，模$P$等于$j$，各位数字之和为$k$的整数数量。

考虑对$i$倍增计算，设$a+b=i$，那么

$$f[i][j][k] = \sum_{j_a=0}^{j} \sum_{k_a=0}^{k} f[a][j_a][k_a] * f[b][j - 10^b j_a][k - k_a]$$

这是一个二维卷积的形式，把$f[i][j]$看作多项式，那么

$$f[i][j] = \sum_{j_a=0}^{j} f[a][j_a] * f[b][j - 10^b j_a]$$

对其每一维DFT后，暴力对这些点值做卷积，再逆DFT运算后即可得到$f[i][j]$所对应的多项式。

时间复杂度$O(P^2 MM \log N + PMM \log MM \log N)$，空间复杂度$O(PMM)$。

## 1.13 Payton numbers

### 【试题来源】

Codechef FEB 15 CUSTPRIM

### 【试题大意】

定义三元组$(a,b,c)$的乘法运算：

```
def multiply((a1, b1, c1), (a2, b2, c2)):

    s = (a1a2 + b1b2 + c1c2) + (a1b2 + b1a2) + (c1 + c2)
    t = floor[s/2] + 16(c1 + c2) - c1c2
    A = (t - 2(a1b2 + b1a2) - (a1c2 + c1a2) + 33(a1 + a2)
            + (b1b2 - a1a2))
    B = (t - 5(a1b2 + b1a2) - (c1b2 + b1c2) + 33(b1 + b2)
            + (2b1b2 + 4a1a2))

    if s is even:
        return (A-540, B-540, 24)
    else:
        return (A-533, B-533, 11)
```

给出一个三元组$(a,b,c)$，判断这个三元组是否是质数。

数据范围：$|a|,|b| \le 10^7$，$c \in \{11,24\}$

### 【算法介绍】

这是一道由结论反构造而得到的题。

设$\omega$为$\omega^3 = \omega - 3$的解，即$\omega = \frac{1+\sqrt{-11}}{2}$。

一个三元组$(a, b, c)$可以映射到$(33 - 2a - c) + (b - a)\omega$，问题转化为在这个新的域下判断这个数是否为质数。

定义$(a + b\omega)' = (a + b - b\omega)$共轭，对于质数的判断有以下结论：

- 若$b = 0$，那么$|a|$必须是质数，且满足以下两个条件之一：
  - $|a| = 2$。
  - $|a| \neq 11$且$-11$在模$|a|$域下没有二次剩余。
- 若$b \neq 0$，那么$(a + b\omega)(a + b - b\omega)$必须是质数。

是否存在二次剩余可以用勒让德负号判断。

时间复杂度$O(\log(a + b))$，空间复杂度$O(1)$。

## 1.14 Ranka

### 【试题来源】

Codechef JAN 15 RANKA

### 【试题大意】

在一个$9 \times 9$的棋盘上下围棋，玩家允许跳过自己的那轮，但必须保证整个棋盘不出现相同的局面。

要求构造一种下法使得棋局进行至少$N$轮。

数据范围：$N \leq 10000$

### 【算法介绍】

由于可以弃权，考虑每次整个棋盘上都只有一方的棋子，在这一方将整个棋盘都放满只剩最后一个位置时，再由对方放上最后一个棋子，使得整个棋盘被对方清空，开始新的一轮。

把$9 \times 9$的棋盘看作一个长度为81的数组，那么假设每次放的棋子的位置都是连续的，就可以得到81种初始位置，而完成一轮需要$2 * 81 - 1$步，这样已经可以满足题中的步数要求。

时间复杂度$O(N)$，空间复杂度$O(1)$。

# 1.15 Xor Queries

## 【试题来源】

Codechef JAN 15 XRQRS

## 【试题大意】

给定一个初始为空的整数序列，以及$M$个操作：

- 类型0：在序列最后加入元素$x$。
- 类型1：在区间$[L, R]$中找到数字$y$使得$x\ xor\ y$最大。
- 类型2：删除序列的最后$k$个元素。
- 类型3：在区间$[L, R]$中统计$\le x$的元素个数。
- 类型4：在区间$[L, R]$中找到第$k$小的元素。

数据范围：$1 \le M, x \le 5 * 10^5$

## 【算法介绍】

对这个序列建一棵可持久化字典树，5种类型的询问都是可持久化字典树的基本操作。

对于类型0、2，每次只会在末端的历史版本添加和删除。

对于类型1，在字典树上贪心遍历计算。

对于类型3、4，在字典树上类似线段树二分遍历计算。

时间复杂度$O(M \log M)$，空间复杂度$O(M \log M)$。

# 1.16 Course Selection

## 【试题来源】

Codechef DEC 14 RIN

## 【试题大意】

有$N$门课，$M$个学期，第$i$门在第$j$个学期学的价值为$X_{i,j}$，$X_{i,j} = -1$代表不能在这个学期学。

有$K$个限制，$A_i$是$B_i$的前置课程，即必须先学$A_i$再学$B_i$。

求出学完所有课程的最大平均价值。

数据范围：$1 \le N, M \le 100$，$0 \le K \le 100$，$-1 \le X_{i,j} \le 100$

## 【算法介绍】

考虑建立最小割模型，对每门课拆出$M+1$个点，设为$p_{i,0} \sim p_{i,M}$。

由于转化为最小割，对于在第$j$个学期学第$i$门课的情况，用一条$p_{i,j-1} \to p_{i,j}$的边表示，流量为$100 - X_{i,j}$，如果割了这条边则表示选择在这个学期学这门课。

考虑解决$K$条限制，比如一条限制$A, B$，对所有$p_{A,j} \to p_{B,j}$连一条流量为$\infty$的边，这样割边集就一定满足要求。

点规模$O(NM)$，边规模$O(NM + KM)$。

# 1.17 Chef and Churu

## 【试题来源】

Codechef NOV 14 FNCS

## 【试题大意】

给定一个长度为$N$的数组$A_i$，以及$N$个函数，第$i$个函数返回数组$[L_i, R_i]$中元素之和。

有$Q$个操作，操作有两种：

- $1\ x\ y$，将数组的第$x$个元素修改为$y$。
- $2\ l\ r$，询问区间$[l, r]$中函数值之和。

数据范围：$1 \le N, Q \le 10^5$，$1 \le A_i, y \le 10^9$

## 【算法介绍】

对数组按$S = \sqrt{N}$分块，预处理出每一块中每个数贡献了几次，对于修改操作可以$O(\sqrt{N})$维护每个块的函数值之和。

对于询问，整块的信息可以直接得到，块外的每个函数如果要$O(1)$计算，就需要维护数组$A_i$的前缀和。

考虑用块状数组维护，修改操作可以看成在被修改位置加上差值，对于前缀和而言就是对一个后缀加上一个数，可以维护整块标记和块外的单个元素标记$O(\sqrt{N})$实现，对于这个块状数组询问可以做到$O(1)$。

时间复杂度$O((N+Q)\sqrt{N})$，空间复杂度$O(N\sqrt{N})$。

## 1.18 Sereja and Order

### 【试题来源】

Codechef NOV 14 SEAORD

### 【试题大意】

有$N$个程序，每个程序要分别在两台电脑上运行，运行的时间分别为$A_i$和$B_i$。

一台电脑不能同时运行两个程序，一个程序也不能同时在两台电脑上运行。

求最少需要多少时间能运行完所有程序，并输出一组方案。

数据范围：$1 \le N \le 10000$，$1 \le A_i, B_i \le 10^5$

### 【算法介绍】

容易发现几个答案的下界：$\sum A_i$，$\sum B_i$，$Max\{A_i + B_i\}$。

这个下界是一定可以取到的，当取到$Max\{A_i + B_i\}$时可以直接构造。

对于另外两种情况，一定是有一台电脑是一直在运行的，并且可以发现最优值取到的方案非常多。

考虑随机两个操作序列，两台电脑都按照对应的操作序列运行程序，强制保证取到极值的那台电脑是连续运行的，相当于判断另一台电脑是否能够在限定时间内完成，用贪心判断。

时间复杂度$O(N)$，空间复杂度$O(N)$。

## 1.19 Children Trips

### 【试题来源】

Codechef OCT 14 TRIPS

### 【试题大意】

给出一棵$N$个点的树，树边的边权为1或2。

给出$M$组询问，从点$S$走向点$F$，每一步最长能走的距离为$P$，所需要的最少步数。

数据范围：$1 \le N, M \le 10^5$，$2 \le P \le 2N$

### 【算法介绍】

对于$P > \sqrt{N}$的情况，答案保证在$O(\sqrt{N})$范围内，可以利用倍增暴力模拟，每次询问的复杂度为$O(\sqrt{N} \log N)$。

对于$P \le \sqrt{N}$的情况，把这些询问单独取出来，对每种$P$预处理出从每个点往祖先方向走$2^k$步后能达到的点，这个可以通过倍增来$O(N \log N)$计算，这样对于每次询问都可以$O(\log N)$完成。这一部分的总耗时为$O(N\sqrt{N} \log N)$。

时间复杂度$O(N\sqrt{N} \log N)$，空间复杂度$O(N \log N)$。

## 1.20 Union on Tree

### 【试题来源】

Codechef OCT 14 BTREE

### 【试题大意】

给出一棵$N$个点的树，每条边的长度都是1。

有$Q$组询问，每组询问给出$K_i$个数对$(A_i, R_i)$，表示与$A_i$距离不超过$R_i$的点都被覆盖，求一共有多少点被覆盖。

数据范围：$N, Q \le 50000$，$\sum K_i \le 500000$

### 【算法介绍】

先考虑$K_i = 1$的情况，可以用点分治解决，维护与每个重心距离不超过$d$的点数，每次询问只需要在静态点分树上查询即可$O(\log N)$完成。

对于$K_i > 1$的情况，可以先对这$K_i$个点求一棵虚树，对于虚树上的每个点，将它的覆盖半径修改为所有能覆盖到它的点中，能够继续延伸的最大值，这样操作不会影响答案，最长延伸长度可以用最短路算法求解。

先将每个点覆盖到的点数累加到答案中，然后考虑计算每条边上多计算的部分。

在被重复计算的边上，一定能找到一个点使得边的两个端点覆盖到这个点后能继续延伸的长度相同，这个点可能是某条边的中点。设延伸长度为$d$，那么减去离这个点距离不超过$d$的点数即可。

时间复杂度$O(N \log N + \sum K_i \log N)$，空间复杂度$O(N \log N)$。

## 1.21 Rectangle Query

### 【试题来源】

Codechef SEPT 14 QRECT

### 【试题大意】

给定一个二维笛卡尔平面，需要支持$Q$次操作：

- $I\ x_1\ y_1\ x_2\ y_2$，插入一个左下角为$(x_1, y_1)$，右上角为$(x_2, y_2)$的矩形。
- $D\ index$，删除第$index$个插入的矩形。
- $Q\ x_1\ y_1\ x_2\ y_2$，询问有多少之前加入的矩形，与左下角为$(x_1, y_1)$，右上角为$(x_2, y_2)$的矩形至少有一个公共点。

数据范围：$1 \le Q \le 10^5$，$1 \le x_1 \le x_2 \le 10^9$，$1 \le y_1 \le y_2 \le 10^9$

### 【算法介绍】

考虑用CDQ分治离线处理，问题转化为计算$A$个矩形对$B$个询问矩形的贡献。

将二维的矩形看作两个一维的区间$[x_1, x_2]$和$[y_1, y_2]$，两个矩形有交的条件为，这两个矩形的两个区间分别有交。

由容斥可得，与矩阵$[x_1, x_2][y_1, y_2]$相交的矩阵数量为：总矩阵数量-与$[x_1, x_2]$不交的数量-与$[y_1, y_2]$不交的数量+与$[x_1, x_2][y_1, y_2]$都不交的数量。

与区间$[l, r]$不交的条件可以描述为$r' < l$或$l' > r$，并且这两个条件是不可能同时达成的，可以简单地分开统计。

对于计算与$[x_1, x_2][y_1, y_2]$都不交的矩阵数量，可以将其中一维排序，另一维树状数组维护处理。

时间复杂度$O(Q \log^2 Q)$，空间复杂度$O(Q)$。

## 1.22 Team Sigma and Fibonacci

### 【试题来源】

Codechef AUG 14 SIGFIB

### 【试题大意】

设$Fib_i$为斐波那契数列第$i$项，即

$$\begin{cases} Fib_0 &= 0 \\ Fib_1 &= 1 \\ Fib_i &= Fib_{i-1} + Fib_{i-2} \quad \forall i \geq 2 \end{cases}$$

对于给定的$N$，求

$$\sum_{x+y+z=N} 6xyzFib_xFib_yFib_z$$

答案对给定的$M$取模。

数据范围：$0 \leq N \leq 10^{18}$，$1 \leq M \leq 10^5$

### 【算法介绍】

设$G(x)$为斐波那契数列的生成函数，$F(x)$为数列$kFib_k$的生成函数。

$$\begin{aligned} G(x) &= \frac{x}{1-x-x^2} \\ F(x) &= x(F(x)+G(x)) + 2x^2(F(x)+G(x)) + x \\ (1-x-x^2)F(x) &= \frac{x^2+2x^3}{1-x-x^2} + x \\ (1-x-x^2)F(x) &= \frac{x+x^3}{1-x-x^2} \\ F(x) &= \frac{x+x^3}{(1-x-x^2)^2} \end{aligned}$$

需要求解的是$6F(x)^3$中$x^N$前的系数。

$$F(x)^3 = \frac{(x+x^3)^3}{(1-x-x^2)^6}$$

这是一个递推数列的生成函数，递推系数即为分母$(1-x-x^2)^6$中$x^i$前系数的相反数。

时间复杂度$O(\log N)$，空间复杂度$O(1)$。

# 1.23 Push the Flow!

## 【试题来源】

Codechef AUG 14 PUSHFLOW

## 【试题大意】

给出一张由$N$个点$M$条边组成的无向图，每条边有一个流量限制，并保证每个点最多属于一个简单环。

需要维护$Q$个操作，操作有两种：

- $0\ S\ T$，求$(S,T)$间的最大流。
- $1\ k\ w$，将第$k$条边的流量限制修改为$w$。

数据范围：$1 \le N \le 100000$，$1 \le M,Q \le 200000$

## 【算法介绍】

对于树的情况，只要求出路径$(S,T)$上流量限制最紧的边即可。

对于仙人掌上的环，可以找出两条简单路径，一条经过了环上最紧边，另一条没经过。可以把每个环的最紧边断掉，将那条边的流量加到环上剩下的边上，问题就转化到树的情况。

对于边的流量修改，可以对每个环维护一个堆，用LCT解决加删边。

时间复杂度$O((M+Q)\log M)$，空间复杂度$O(N+M)$。

## 1.24 Game of Numbers

### 【试题来源】

Codechef JULY 14 GNUM

### 【试题大意】

给出一个长度为$N$的数组$A_i$，以及一个长度为$N$的数组$B_j$。

维护两个二元组的集合$S_1, S_2$，一开始$S_1 = S_2 = \emptyset$。

每次选择两个数对$(i, j)(p, q)$，满足$(i, j)$不在$S_1$中，$(p, q)$不在$S_2$中，且$B_j > A_i$，$B_p < A_q$，$gcd(B_j, A_i, B_p, A_q) > 1$，然后将$(i, j)$加入$S_1$，$(p, q)$加入$S_2$。

求最多能进行多少次这样的操作。

数据范围：$1 \leq N \leq 400$，$1 \leq A_i, B_j \leq 10^9$

### 【算法介绍】

由于$B_j > A_i$，$B_p < A_q$的限制，$S_1 \cap S_2 = \emptyset$，所以可以把这些数对看成一个$N^2$个点的二分图，两边$gcd > 1$的点对之间有边相连，需要对这张二分图求出最大匹配。

朴素连边方式的边数为$O(N^2 \log A)$，考虑对所有出现过的质因子拆点，这一部分的点数为$O(N \log A)$，每个数对都与其$gcd$的质因子连边，这样边数缩减为$O(N^2 \log A)$。

点规模$O(N^2 + N \log A)$，边规模$O(N^2 \log A)$

## 1.25 Sereja and Equality

### 【试题来源】

Codechef JULY 14 SEAEQ

### 【试题大意】

定义两个长度为$n$的数组$A, B$相似，需要满足对于所有$i(1 \le i \le n)$，$C(A, A_i) = C(B, B_i)$，其中$C(X, y)$等于满足$X_i < y$且$i \in [1, n]$的$i$的数量。

对于两个排列$P_1, P_2$，定义函数$F(P_1, P_2)$等于满足$P_1[l, r]$与$P_2[l, r]$相似，且$P_1[l, r]$中逆序对数不超过$E$的数对$(l, r)$的对数量。

计算对于所有排列$P_1, P_2$的$F(P_1, P_2)$之和，答案对$10^9 + 7$取模。

数据范围：$n \le 500$，$E \le 10^6$

### 【算法介绍】

设$g[i][j]$为长度为$i$，逆序对数为$j$的排列数，$f[i][j]$为长度为$i$，逆序对数最多为$j$的排列数，那么

$$\begin{aligned} g[i][j] &= f[i-1][j] - f[i-1][j-i] \\ f[i][j] &= \sum_{k=0}^{j} g[i][k] \end{aligned}$$

枚举相似的区间长度$L$，共有$n - L + 1$种位置，在单个排列中，区间内的数有$\binom{n}{L} f[L][E]$种放法，剩下位置的数有$(n - L)!$种放法。

长度为$L$的区间对答案的贡献为$(n - L + 1)\left((n - L)!\binom{n}{L}\right)^2 f[L][E]$。

时间复杂度$O(n^3)$，空间复杂度$O(n^3)$。

## 1.26 Two Companies

### 【试题来源】

Codechef JUNE 14 TWOCOMP

### 【试题大意】

给出一棵$N$个点的树，以及两个集合，分别有$M_1$和$M_2$个元素，每个元素都为树上的一条路径，并有一个权值。

对这两个集合都求出一个子集，使得在被选出的子集中不存在两条路径相交且属于不同集合。

要求选出的两个子集的权值和尽量大。

数据范围：$1 \le N \le 10^5$，$1 \le M_1, M_2 \le 700$

### 【算法介绍】

问题相当于求二分图最大权独立集，可以用最小割解决。

$O(M_1M_2)$枚举来判断是否需要连边，对于两条路径$(u_i, v_i)(u_j, v_j)$，设$c_i = LCA(u_i, v_i), c_j = LCA(u_j, v_j)$，判断这两条路径是否相交，只要判断$(u_i, c_i)(v_i, c_i)$是否和$(u_j, c_j)(v_j, c_j)$相交即可，而对于这样的两端路径，只要用DFS括号序中的嵌套关系就能完成判断。

只要从源汇向每个元素连流量为权值的边，相交的元素之间连流量为$\infty$的边，最后用权值和减去最小割就是最大权独立集。

点规模$O(M_1 + M_2)$，边规模$O(M_1M_2)$。

## 1.27 Sereja and Arcs

### 【试题来源】

Codechef JUNE 14 SEAARC

### 【试题大意】

有$N$个点，坐标为$(1,0) \sim (N,0)$，每个点有一个颜色，坐标为$(i,0)$的点的颜色为$A_i$。

在所有颜色相同的点对中都画上了圆弧，即对于$A_i = A_j (i \neq j)$，那么会画一条圆弧连接$(i,0)$和$(j,0)$，圆心在$x$轴上，且这条圆弧的颜色为$A_i$，所有圆弧都在第一象限内。

求有多少对不同颜色的圆弧相交了。

数据范围：$1 \leq N, A_i \leq 10^5$

### 【算法介绍】

考虑两对圆弧，不相交的可能为()()和(())。

对于()()这种情况，容易计算$f_i$为右端点位置在$i$的圆弧数，$g_i$为左端点$\geq i$的圆弧数，这一部分的答案即为$\sum_{i=1}^{N-1} f_i g_{i+1}$，可以$O(N)$完成。

对于(())这种情况，考虑将颜色按出现次数分为两类，设阈值为$S$。

对于出现次数$\geq S$的颜色，只可能有$\frac{N}{S}$种，对每种单独计算一次贡献。

- 当这种颜色出现在外层时，处理出$L_i, R_i$分别为左侧和右侧该种颜色的点数，那么对于剩下颜色的一对括号$(u,v)$，贡献为$L_u * R_v$，可以$O(N)$计算。
- 当这种颜色出现在内层时，同样处理出$L_i$，对于括号$(u,v)$，贡献为$\frac{(L_v - L_u)(L_v - L_u - 1)}{2}$，把式子展开后也可以$O(N)$计算。

这一部分的复杂度为$O(\frac{N^2}{S})$。

对于出现次数$< S$的颜色，能形成的圆弧总数为$O(NS)$级别，可以转化为二维数点问题，用树状数组维护，复杂度为$O(NS\log N)$。

总复杂度为$O(\frac{N^2}{S} + NS\log N)$，于是$S = \sqrt{\frac{N}{\log N}}$较优。

时间复杂度$O(N\sqrt{N\log N})$，空间复杂度$O(N)$。

## 1.28 Dynamic Trees and Queries

### 【试题来源】

Codechef MAY 14 ANUDTQ

### 【试题大意】

给定一棵$N$个点的有根树，以及$M$个操作，强制在线：

- 给一个节点增加一个儿子节点。
- 将一棵子树中的所有点权值加上一个值。
- 将一棵子树中的点删除。
- 询问一棵子树的权值和。

数据范围：$1 \leq N, M \leq 10^5$

### 【算法介绍】

经典问题，用平衡树维护DFS序，可以用括号方便地确定子树，用平衡树维护标记和子树移动和删除。

时间复杂度$O(N + M\log(N + M))$，空间复杂度$O(N + M)$。

## 1.29 Sereja and Subsegment Increasings

### 【试题来源】

Codechef MAY 14 SEINC

### 【试题大意】

给出一个长度为$n$的数列$A_i$，每次可以选择一段区间把所有$A_i$加一后对4取模，求最少需要多少次操作可以把数列$A$转化为数列$B$。

数据范围：$1 \le n \le 10^5$，$0 \le A_i, B_i \le 3$

### 【算法介绍】

先把所有$A_i$在模域下设为$B_i - A_i$，再做差分求出$C_i = A_i - A_{i-1}$，易得$C_i \in [-3, 3]$。

做差分之后，问题转化为每次可以选一对$u, v(u < v)$，将$C_u$加一，$C_v$减一，最后要使得数列$C$全为0。

假如不考虑模域，答案显然为所有正的$C_i$的和。在模域下，可以将$C_i$的值随意加4或减4，但必须保证所有$C_i$的和为0。

考虑$1 \sim n$遍历$C_i$，假如$C_i = 1$，不做调整，当$C_i \ge 2$时，可以将前面的一个-2或-3加4，将这个$C_i$减4。

经过这样的调整后最后的数列即为答案。

时间复杂度$O(n)$，空间复杂度$O(n)$。

## 1.30 Chef and Graph Queries

### 【试题来源】

Codechef MARCH 14 GERALD07

### 【试题大意】

给定一张$N$个点$M$条边的无向图$G$，$Q$组询问，当仅保留编号在$[L_i, R_i]$范围内的边时，该图会有多少连通块。

数据范围：$1 \le N, M, Q \le 200000$

### 【算法介绍】

考虑将询问按右端点排序，按照右端点扫描过去，维护一棵生成森林，连通块个数即为$N$-生成森林中的边数。

维护这个生成森林时，每新加入一条边，就找出那两个点间编号最小的边删去，再连上新的边。

对于询问操作，假设$R_i$固定，只需要求有多少森林中的边编号$\ge L_i$，可以用线段树或树状数组维护。

时间复杂度$O(M \log N + Q \log M + Q \log Q)$，空间复杂度$O(N + M + Q)$。

# 1.31 The Street

## 【试题来源】

Codechef MARCH 14 STREETTA

## 【试题大意】

有$N$家店铺，以及$M$个操作：

- $1\ u\ v\ a\ b$，编号在$[u,v]$范围内的店铺，对于店铺$u+i$，增加一个价格为$b+ai$的商品。
- $2\ u\ v\ a\ b$，编号在$[u,v]$范围内的店铺，对于店铺$u+i$，增加$b+ai$的税收。
- $3\ i$，在店铺$i$购买最贵的物品后加上税收需要支付多少钱。

数据范围：$1 \le N \le 10^9$，$1 \le M \le 3*10^5$

## 【算法介绍】

对于操作1，考虑每个点维护一个等差数列标记，考虑合并两个标记时，一定可以把当前区间划分为两段，使得这两个等差数列各在其中一段为较优值。

为了保证每个点上只有一个标记，就把较短的那一段下传到对应的子树，这样每次递归只有一个分支，时间复杂度为$O(\log^2 N)$。

对于操作2，由于两个等差数列相加还是等差数列，所以可以用简单的线段树标记维护。

时间复杂度$O(M\log^2 N)$，空间复杂度$O(M\log^2 N)$。

# 1.32 Graph Challenge

## 【试题来源】

Codechef FEB 14 DAGCH

## 【试题大意】

给出一张$N$个点$M$条边的有向图，保证每个点的标号为它被从1开始DFS到的序号。

点$x$是点$y$的$supreme$点，需要满足存在一条有向路满足其中经过的所有点$v$保证$x \leq y \leq v$。

点$x$是点$y$的$superior$点，需要满足$x$是$y$的所有$supreme$点中序号最小的。

$Q$组询问，求有多少点是点$v$的$superior$点。

数据范围：$1 \leq N, Q \leq 10^5$，$N - 1 \leq M \leq 200000$

## 【算法介绍】

对读入的边按照目标点排序后连边遍历，DFS后就可以得到题中给定的DFS序号所对应的DFS树。

$superior$点等价于半必经点，可以直接用dominator tree的做法来算。

时间复杂度$O(M \log M + \alpha N)$，空间复杂度$O(N + M)$。

## 1.33 Count on a Treap

### 【试题来源】

Codechef FEB 14 COT5

### 【试题大意】

要求维护一个大根堆Treap，支持$N$次操作：

- $0\ k\ w$，插入一个关键字为$k$，权值为$w$的点。
- $1\ k$，删除一个关键字为$k$的点。
- $2\ k_u\ k_v$，求关键字为$k_u, k_v$的两点在树上的距离。

保证任意时刻树上不存在关键字或权值相同的点。

数据范围：$1 \le N \le 200000$，$0 < k, w, k_u, k_v < 2^{32}$

### 【算法介绍】

两个点$k_u, k_v$在Treap上的LCA为关键字在$[k_u, k_v]$范围内的权值最大的点，它们之间的距离为$deep_{k_u} + deep_{k_v} - 2deep_{LCA(k_u,k_v)}$。

考虑求点$k$在树上的深度，相当于求以点$k$为起点的，向左向右的权值上升子序列长度之和，可以用线段树维护。

考虑求向右的权值上升子序列长度：

- $calc(l, r, val)$，函数返回以权值$val$为初值，区间$[l, r]$中的上升序列长度。
- $Max[l, r]$，区间$[l, r]$中的最大权值。
- $len[l, r]$，$calc(l, r, 0)$的值。

通过维护$Max[l,r]$和$len[l,r]$，$calc(l,r,val)$可以$O(\log k)$完成计算：

- $val \geq Max[l,mid]$时，$calc(l,r,val) = calc(mid+1,r,val)$。
- 否则，$calc(l,r,val) = calc(l,mid,val) + len[l,r] - len[l,mid]$。

将关键字和权值离散化后可以都在$O(N)$范围内，每个操作都可以$O(\log^2 N)$完成。

时间复杂度$O(N \log^2 N)$，空间复杂度$O(N)$。

## 1.34 Counting D-sets

### 【试题来源】

Codechef JAN 14 CNTDSETS

### 【试题大意】

在一个$N$维空间中，求有多少本质不同的点集满足其直径恰好为$D$。点集的直径定义为点集中距离最远的一对点的距离，而点的距离定义为两个点差距最大的那一维坐标的差值。

两个点集被认为是本质不同的，要求它们不能通过平移坐标轴互相得到。

点坐标必须为整数，答案对$10^9+7$取模。

数据范围：$1 \le N \le 1000$，$1 \le D \le 10^9$

### 【算法介绍】

直径恰好为$D$的点集数为直径不超过$D$的点集数减去直径不超过$D-1$的点集数。

为了保证求出的点集本质不同，可以强制每一维坐标的取值范围在$[0, D]$且0一定要取到。

设$F(k)$为有$k$维坐标强制没有取到0的点集数，有容斥得答案即为

$$\sum_{k=0}^{N}(-1)^k\binom{N}{k}F(k)$$

其中

$$F(k) = 2^{D^k(D+1)^{N-k}}$$

时间复杂度$O(N\log D)$，空间复杂度$O(N)$。

## 1.35 Counting The Important Pairs

### 【试题来源】

Codechef JAN 14 TAPAIR

### 【试题大意】

给出一张$N$个点$M$条边的连通无向图，对于所有边的无序二元组$(u, v)$，有多少二元组删去后原图会不连通。

数据范围：$1 \le N \le 100000$，$1 \le M \le 300000$

### 【算法介绍】

对这张图求一棵生成树，给所有非树边随机一个权值，树边的权值为所有跨越它的非树边的权值异或和。

基于随机，所有会导致图不连通的边集合都会至少有一个异或和的子集。

对于两条边的情况，就是两条边的权值相同，或者其中有一条边权值为0。

时间复杂度$O(N + M)$，空间复杂度$O(N + M)$。

## 1.36 Query on a tree VI

### 【试题来源】

Codechef DEC 13 QTREE6

### 【试题大意】

给出一棵$N$个点的树，点有两种颜色黑或白，一开始所有点都是白色的。

给出$M$个操作，操作有两种：

- 0 $u$，询问点$u$所在的同色连通块有多少点。
- 1 $u$，将点$u$反色。

数据范围：$1 \le N, M \le 10^5$

### 【算法介绍】

对于每个同色连通块，找出这个连通块中深度最浅的点，在树上这个点是唯一的，所以可以作为这个连通块的代表。

考虑反色操作，需要维护每个点下方所有异色点的连通块大小之和，在反色时可以迅速求出它所在的同色连通块位于它子树中的点数。

对于被反色点的祖先，他们所在的同色连通块大小需要加上或减去被反色点的同色子树大小，可以用树链剖分维护，相当于对一段树链加上一个值，需要被修改的祖先范围也可以用树链剖分求出，修改的复杂度为$O(\log^2 N)$。

由于已经维护了被询问的信息，所以询问只需要在线段树上把标记下传就能完成，复杂度为$O(\log N)$。

时间复杂度$O(N + M\log^2 N)$，空间复杂度$O(N)$。

## 1.37 Petya and Sequence

### 【试题来源】

Codechef DEC 13 REALSET

### 【试题大意】

给出一个长度为$N$的数列$A_0, A_1, \ldots, A_{N-1}$，问是否存在一个长度为$N$的数列$B$满足：

- 存在至少一个$i(0 \le i \le N-1)$满足$B_i \ne 0$。
- 对于任意$0 \le j \le N-1$满足

$$\sum_{i=0}^{N-1} A_i B_{(i+j) mod\ N} = 0$$

数据范围：$1 \le N \le 30000$

### 【算法介绍】

可以列出一个方程组，判读是否有解，所以问题可以转化为求矩阵$A$的行列式的值是否为0。

$$|A| = \begin{vmatrix} A_0 & A_1 & A_2 & \ldots & A_{N-1} \\ A_{N-1} & A_0 & A_1 & \ldots & A_{N-2} \\ A_{N-2} & A_{N-1} & A_0 & \ldots & A_{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_1 & A_2 & A_3 & \ldots & A_0 \end{vmatrix}$$

设

$$f(x) = \sum_{k=0}^{N-1} A_k x^k$$

那么$|A| = \prod_{k=0}^{N-1} f(\epsilon_k)$，其中$\epsilon_k = \omega_n^k = e^{\frac{2\pi ki}{n}}$，证明如下：

设

$$B = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ \epsilon_0 & \epsilon_1 & \epsilon_2 & \dots & \epsilon_{N-1} \\ {\epsilon_0}^2 & {\epsilon_1}^2 & {\epsilon_2}^2 & \dots & {\epsilon_{N-1}}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ {\epsilon_0}^{N-1} & {\epsilon_1}^{N-1} & {\epsilon_2}^{N-1} & \dots & {\epsilon_{N-1}}^{N-1} \end{pmatrix}$$

$B$是一个范德蒙矩阵，所以$|B| = \prod_{0 \le j < i \le N-1}(\epsilon_i - \epsilon_j)$。

因为$\epsilon_i \ne \epsilon_j (i \ne j)$，所以$|B| \ne 0$。

由于${\epsilon_k}^n = 1$，所以

$$\begin{aligned} f(\epsilon_k) &= A_0 + A_1\epsilon_k + A_2{\epsilon_k}^2 + \ldots + A_{N-1}{\epsilon_k}^{N-1} \\ \epsilon_k f(\epsilon_k) &= A_{N-1} + A_0\epsilon_k + A_1{\epsilon_k}^2 + \ldots + A_{N-2}{\epsilon_k}^{N-1} \\ {\epsilon_k}^2 f(\epsilon_k) &= A_{N-2} + A_{N-1}\epsilon_k + A_0{\epsilon_k}^2 + \ldots + A_{N-3}{\epsilon_k}^{N-1} \\ \dots & \\ {\epsilon_k}^{N-1} f(\epsilon_k) &= A_1 + A_2\epsilon_k + A_3{\epsilon_k}^2 + \ldots + A_0{\epsilon_k}^{N-1} \end{aligned}$$

所以

$$AB = \begin{pmatrix} f(\epsilon_0) & f(\epsilon_1) & f(\epsilon_2) & \dots & f(\epsilon_{N-1}) \\ \epsilon_0 f(\epsilon_0) & \epsilon_1 f(\epsilon_1) & \epsilon_2 f(\epsilon_2) & \dots & \epsilon_{N-1} f(\epsilon_{N-1}) \\ {\epsilon_0}^2 f(\epsilon_0) & {\epsilon_1}^2 f(\epsilon_1) & {\epsilon_2}^2 f(\epsilon_2) & \dots & {\epsilon_{N-1}}^2 f(\epsilon_{N-1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ {\epsilon_0}^{N-1} f(\epsilon_0) & {\epsilon_1}^{N-1} f(\epsilon_1) & {\epsilon_2}^{N-1} f(\epsilon_2) & \dots & {\epsilon_{N-1}}^{N-1} f(\epsilon_{N-1}) \end{pmatrix}$$

$$
\begin{aligned}
|AB| &= \left(\prod_{k=0}^{N-1} f(\epsilon_k)\right)\left(\prod_{0\le j<i\le N-1}(\epsilon_i-\epsilon_j)\right)\\
|A||B| &= \left(\prod_{k=0}^{N-1} f(\epsilon_k)\right)|B|\\
|A| &= \prod_{k=0}^{N-1} f(\epsilon_k)
\end{aligned}
$$

设

$$
\begin{aligned}
F(x) &= \sum_{i=0}^{N-1} A_i\omega_{2n}^{ii}\\
G(x) &= \sum_{i=-N+1}^{N-1} \omega_{2n}^{-ii}
\end{aligned}
$$

那么$F(x)G(x)$中第$i$项前的系数即为

$$
\sum_{j=0}^{N-1} A_j\omega_{2n}^{jj}\omega_{2n}^{-(i-j)(i-j)} = \sum_{j=0}^{N-1} A_j\omega_{2n}^{2ij-ii}
$$

而

$$
|A| = \prod_{i=0}^{N-1}\sum_{j=0}^{N-1} A_j\omega_n^{ij}
$$

所以只要乘上$\omega_{2n}^{ii}$就能得到需要的项。

时间复杂度$O(N\log N)$，空间复杂度$O(N)$。

## 1.38 Gangsters of Treeland

### 【试题来源】

Codechef NOV 13 MONOPLOY

### 【试题大意】

给出一棵$N$个点的有根树，根为1。每个点有一个权值，一开始每个点的权值都是不同的。

需要维护$Q$次操作，操作有两种：

- $O\ u$，将点$u$到根路径上的所有点的权值赋为一个没有出现过的新权值。
- $q\ u$，询问点$u$子树所有点到根路径上不同权值种数的平均值。

数据范围：$1 \le N, Q \le 10^5$

### 【算法介绍】

考虑用LCT维护那些权值相同的树链，$O$操作相当于LCT中的access操作。

对于每条权值树链，链顶节点将会对它的整颗子树产生贡献，这个贡献可以用树状数组或线段树维护，每次$O(\log N)$完成修改。

由LCT的均摊分析可以保证树链的变动次数最多为$O(N \log N)$次。

时间复杂度$O(N \log^2 N)$，空间复杂度$O(N)$。

# 1.39 Queries With Points

## 【试题来源】

Codechef NOV 13 QPOINT

## 【试题大意】

给出$N$个最多只在端点处相交的多边形，多边形上总点数为$K$。

给出$Q$个询问，求点$(x_i, y_i)$在哪个多边形内，强制在线。

数据范围：$N, Q \leq 100000$，$K \leq 300000$

## 【算法介绍】

这是经典的点定位问题，可以用线段树套vector维护，由于线段只会在端点处相交，所以每个vector中的线段可以按照$y$坐标的相对关系进行排序。

对于每个询问，找到它上方遇到的第一条线段，通过这条线段的方向判断这个点在多边形内部还是外部。

时间复杂度$O((Q+K)\log^2 K)$，空间复杂度$O(K\log K)$。

## 1.40 Fibonacci Number

### 【试题来源】

Codechef OCT 13 FN

### 【试题大意】

求一个$n$使得斐波那契数列第$n$项$Fib_n \equiv C\ (mod\ P)$。

数据范围：$11 \le P \le 2*10^9$，$P\ mod\ 10$为完全平方数

### 【算法介绍】

考虑斐波那契数列的通项式

$$Fib_n = \frac{\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n}{\sqrt{5}}$$

设$\alpha = \frac{1+\sqrt{5}}{2}$，当$n$为奇数时$Fib_n = \frac{\alpha^n+\alpha^{-n}}{\sqrt{5}}$，$n$为偶数时$Fib_n = \frac{\alpha^n-\alpha^{-n}}{\sqrt{5}}$。

考虑奇数的情况，偶数的情况也可以相应得到

$$\begin{aligned}\alpha^n + \alpha^{-n} &= \sqrt{5}C \\ \alpha^{2n} - \sqrt{5}C\alpha^n + 1 &= 0 \\ \alpha^n &= \frac{\sqrt{5}C \pm \sqrt{5C^2-4}}{2}\end{aligned}$$

开根号操作在模域下可以用原根完成，对于无法开根的情况判断无解。

离散对数问题可以用BSGS算法$O(\sqrt{P})$解决。

时间复杂度$O(\sqrt{P})$，空间复杂度$O(\sqrt{P})$。

# 1.41 Two Roads

## 【试题来源】

Codechef SEPT 13 TWOROADS

## 【试题大意】

给出一个$N$个点的点集，求两条直线，使得每个点到这两条直线的较短距离的平方的平均值最小。

求出这个最小平均值。

数据范围：$3 \le N \le 100$

## 【算法介绍】

先考虑一条直线的情况，设直线方程为$y = kx + b$，即最小化

$$\begin{aligned}
\delta &= \sum_{i=1}^{N} \frac{(kx_i - y_i + b)^2}{k^2+1} \\
&= \frac{k^2 \sum {x_i}^2 - 2k \sum x_i y_i + 2bk \sum x_i + \sum {y_i}^2 - 2b \sum y_i + Nb^2}{k^2+1} \\
&= \frac{Nb^2 + (2k \sum x_i - 2 \sum y_i)b + (k^2 \sum {x_i}^2 - 2k \sum x_i y_i + \sum {y_i}^2)}{k^2+1}
\end{aligned}$$

假如将$k$看作常量，要使$\delta$最小就需满足

$$b = \bar{y} - k\bar{x}$$

设

$$\begin{aligned}
A &= \sum {x_i}^2 - 2\bar{x} \sum x_i + N\bar{x}^2 \\
B &= 2\bar{y} \sum x_i + 2\bar{x} \sum y_i - 2 \sum x_i y_i - 2N\bar{x}\bar{y} \\
C &= \sum {y_i}^2 - 2\bar{y} \sum y_i + N\bar{y}^2
\end{aligned}$$

那么

$$\delta = \frac{Ak^2 + Bk + C}{k^2 + 1}$$
$$(A - \delta)k^2 + Bk + C - \delta = 0$$

在这个一元二次方程中

$$\Delta = B^2 - 4(A - \delta)(C - \delta)$$

当$\Delta$取到0时，$\delta$最小，可以直接解出，这样一条直线的情况就解决了。

考虑两条直线的情况，假设这两条直线已经确定，那么这个点集将被4条对角线分割，每一部分的点将会在对应的那条直线上取到较小值。

容易发现，这两条对角线一定是垂直的，并且可以在不改变点集分布的前提下，将对角线移动为其中一条经过两个点，另一条经过一个点。

可以$O(N^2)$枚举一条对角线，对于每种情况再处理出所有点在这条对角线上的垂足，并将其排序按顺序扫过来，这样就可以得到所有可能的点集分布方案，对于每种方案分别就最优直线即可。

时间复杂度$O(N^3 \log N)$，空间复杂度$O(N)$。

# 1.42 Fibonacci Numbers on Tree

## 【试题来源】

Codechef SEPT 13 FIBTREE

## 【试题大意】

给出一棵$N$个节点的树，需要维护$M$次操作，操作有四种，强制在线：

- 将$(x, y)$上的点依次加上斐波那契数列。
- 询问以$x$为根时，子树$y$的权值和。
- 询问路径$(x, y)$的权值和。
- 将整棵树的状态更改为第$k$次操作后的状态。

所有答案对$10^9 + 9$取模。

数据范围：$N, M \leq 100000$

## 【算法介绍】

斐波那契数列的通项为$\frac{1}{\sqrt{5}}\left((\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n\right)$，$\sqrt{5}$的值是可以在模$10^9 + 9$域下求出的，所以问题转化为在树链上加一个等比数列，由于公比是定值，所以可以简单地维护。

考虑使用树链剖分和DFS序解决，在生成DFS序时优先往重链方向遍历，这样一条重链在DFS序中的位置是连续的，同时一棵子树也是连续的，所以只要一棵线段树就能完成维护。

对于最后一种操作，可以用可持久化解决。

时间复杂度$O(N + M\log^2 N)$，空间复杂度$O(N + M\log^2 N)$。

## 1.43 Music & Lyrics

### 【试题来源】

Codechef AUG 13 LYRC

### 【试题大意】

给出$W$个串$P_i$以及$N$个串$S_j$，对于每个$P_i$求它在所有$S_j$中出现的总次数。

串$P_i$和串$S_j$由大小写字母、数字以及‘-’组成。

数据范围：$1 \le W \le 500$，$1 \le |P_i| \le 5000$，$1 \le N \le 100$，$1 \le |S_j| \le 50000$

### 【算法介绍】

对$P_i$建立AC自动机，将所有$S_j$在自动机上遍历，所有经过的位置权值+1。

每个$P_i$出现的总次数为fail树中它的子树的权值和。

时间复杂度$O(\sum|P_i| + \sum|S_j|)$，空间复杂度$O(\sum|P_i|)$。

## 1.44 Prime Distance On Tree

### 【试题来源】

Codechef AUG 13 PRIMEDST

### 【试题大意】

给出一棵$N$个点的树，求随机选一对点$u, v (u \neq v)$，$u$到$v$的路径长度为质数的概率。

数据范围：$2 \leq N \leq 50000$

### 【算法介绍】

考虑点分治，设当前重心为$x$，求出当前连通块中所有点到$x$的距离，用FFT计算卷积。在同一棵子树内的路径会算重，可以对每棵子树单独做一次FFT卷积，把重复部分剪掉。

FFT的长度与该子树深度相关，所以可以保证FFT的总长度为$O(N \log N)$。

时间复杂度$O(N \log^2 N)$，空间复杂度$O(N)$。

## 1.45 Across the River

### 【试题来源】

Codechef JULY 13 RIVPILE

### 【试题大意】

有一条宽度为$W$的河流，河岸为直线$y = 0$以及直线$y = W$。

在河上有$N$个木桩$(X_i, Y_i)$，木桩上可以放圆盘，圆盘有$M$种，第$i$种半径为$R_i$，价格为$C_i$。

现在要求花最小的代价，得到一条可以渡河的圆盘路径。

数据范围：$N, M \leq 250$

### 【算法介绍】

对每个木桩建$2M$的点，为每种圆盘所对应的入点和出点，在入点和出点间加上代价为$C_i$的边。

在这$O(NM)$个点中能走到的圆盘间连边，朴素的连边方式的边数为$O(N^2M^2)$，将圆盘预先按半径排序后可以将边数缩减为$O(N^2M)$。

在构造出的这张图上跑最短路即可得到答案。

点规模$O(NM)$，边规模$O(N^2M)$

## 1.46 Two k-Convex Polygons

### 【试题来源】

Codechef JUNE 13 TKCONVEX

### 【试题大意】

给出$n$根木棍，求能否在其中选出$2k$根木棍拼成两个凸多边形。

使得两个凸多边形都恰好有$k$根木棍组成，且任意相邻的边都不共线。

数据范围：$2k \le n \le 1000$，$3 \le k \le 10$

### 【算法介绍】

能构成满足题设条件的多边形的木棍集合需要满足最长的木棍的长度小于其余木棍长度之和。

将木棍按照长度排序后，合法解一定可以由其中的一段或两段连续区间组成。

对于两段连续区间的情况，一定是同一段属于同一个凸多边形，可以很简单地得到。

对于一段区间的情况，强制将这个区间的第一条线段加入第一个多边形后，可能的方案只有$\binom{2k-1}{k}$种，可以搜索枚举计算所有可能方案。

时间复杂度$O(n\log n + \binom{2k-1}{k}n)$，空间复杂度$O(n)$。

## 1.47 Count Special Matrices

### 【试题来源】

Codechef JUNE 13 SPMATRIX

### 【试题大意】

求有多少$N \times N$的矩阵，满足

- $\forall 1 \le x \le N, A_{x,x} = 0$
- $\forall 1 \le x, y \le N, A_{x,y} = A_{y,x} \in \{1, 2, ..., N-2\}$
- $\forall 1 \le x, y, z \le N, A_{x,y} \le max(A_{x,z}, A_{z,y})$
- $\forall k \in \{1, 2, ..., N-2\}$, 存在$A_{x,y} = k$

$T$组询问，答案对$10^9 + 7$取模。

数据范围：$1 \le T \le 10^5$，$3 \le N \le 10^7$

### 【算法介绍】

计算公式为

$$\frac{N!(N-1)!}{2^{N-1}} \left( \frac{N}{2} - \frac{2}{3} - \frac{\sum_{k=1}^{N-1} \frac{1}{k}}{3} \right)$$

可以$O(N)$预处理阶乘和逆元，其中

$$k^{-1} \equiv -(p \; mod \; k)^{-1} \left\lfloor \frac{p}{i} \right\rfloor \; (mod \; p)$$

时间复杂度$O(N+T)$，空间复杂度$O(N)$。

## 1.48 Queries on tree again!

### 【试题来源】

Codechef MAY 13 QTREE

### 【试题大意】

给出一张$N$个点$N$条边的简单无向连通图，即这张图仅包含一个环，并保证这个环的长度为奇数。

图中的每条边都有一个边权。

需要维护$Q$个操作，操作有两种：

- 将两点$(u, v)$的经过边数最少的路径上的所有边的边权变为相反数。
- 求两点$(u, v)$的经过边数最少的路径上的所有边的边权构成数列的最大子段和。

数据范围：$N, Q \leq 100000$

### 【算法介绍】

将环上的边断开一条，就可以转化为树的情况，在原图上的简单路径最多为两条，只要在每次操作时判断即可。

考虑用树链剖分维护，需要维护最大子段和、左端连续的最大子段和、右端连续的最大子段和，由于有取相反数操作，还需要同时维护最小值。

时间复杂度$O(N + Q\log^2 N)$，空间复杂度$O(N)$。

## 1.49 Little Elephant and Colored Coins

### 【试题来源】

Codechef MARCH 13 LECOINS

### 【试题大意】

有$N$种硬币，第$i$种硬币面值为$V_i$，颜色为$C_i$，每种硬币都有无限个。

$Q$组询问，用一些硬币组成面值恰好为$S$，最多可以使用多少种颜色。

数据范围：$1 \leq N \leq 30$，$1 \leq V_i, Q \leq 200000$，$1 \leq C_i \leq 10^9$，$1 \leq S \leq 10^{18}$

### 【算法介绍】

选出面值最小的一枚硬币作为模域，求出$f_{u,v}$为使用$u$种硬币后，在模域下构成$v$的最小面值和，剩下的部分可以直接用最小面值的那种硬币填充。

求$f_{u,v}$是一个最短路模型，可以通过$O(n^2)$次最短路得到。

不使用最小硬币的情况也可以用类似的方法得到。

时间复杂度$O(n^2 V \log V)$，空间复杂度$O(nV)$。

## 1.50 Making Change

### 【试题来源】

Codechef MATCH 13 CHANGE

### 【试题大意】

有$N$种硬币，第$i$种硬币的面值为$D_i$，保证$D_i$两两互质。

每种硬币可以用无限个，求有多少种方案可以拼出面值$C$。

数据范围：$1 \le N \le 50$，$1 \le D_i \le 500$，$1 \le C \le 10^{100}$

### 【算法介绍】

设硬币面值集合$D = \{D_i\}$

$$f(x) = \prod_{d \in D} \sum_{i=0}^{\infty} x^{id}$$

原问题相当于求$x^C$前的系数。

由等比数列求和可以得到两个重要的恒等式：

$$\begin{aligned}\frac{1}{1-x} &= \sum_{i=0}^{\infty} x^i \\ 1 - x^n &= (1-x)\sum_{i=0}^{n-1} x^i\end{aligned}$$

那么

$$\begin{aligned}f(x) &= \prod_{d \in D} \frac{1}{1-x^d} \\ &= \frac{1}{(1-x)^N} \prod_{d \in D} \frac{1}{\sum_{i=0}^{d-1} x^i}\end{aligned}$$

根据部分分式分解，可以设

$$f(x) = \frac{A(x)}{(1-x)^N} + \sum_{d\in D} \frac{B_d(x)}{\sum_{i=0}^{d-1} x^i}$$

先考虑求解$B_d(x)$，方程两边同乘$\sum_{i=0}^{d-1} x^i$得

$$\frac{1}{(1-x)\prod_{d'\in D, d'\neq d}(1-x^{d'})} = \left(\sum_{i=0}^{d-1} x^i\right)\left(\frac{A(x)}{(1-x)^N} + \sum_{d'\in D, d'\neq d} \frac{B_{d'}(x)}{\sum_{i=0}^{d'-1} x^i}\right) + B_d(x)$$

由分式分解得多项式$B_d(x)$的阶数小于分母的阶数，即$< d-1$。

考虑将$\omega_d^j (j = 1, 2, \ldots, d-1)$这$d-1$个点代入插值，由于$\omega_d^d = 1$，所以

$$\sum_{i=0}^{d-1} \omega_d^{ij} = \frac{1-\omega_d^{jd}}{1-\omega_d^j} = 0$$

由此可以得到

$$B_d(\omega_d^j) = \frac{1}{(1-\omega_d^j)\prod_{d'\in D, d'\neq d}(1-\omega_d^{jd'})}$$

有结论

$$\frac{1}{1-\omega_d^j} = -\frac{1}{d}\sum_{i=0}^{d-1} i\omega_d^{ij}$$

证明如下：

$$\begin{aligned} S &= \sum_{i=0}^{d-1}(i+1)x^i \\ xS &= \sum_{i=1}^{d-1} ix^i + dx^d \\ (1-x)S &= \sum_{i=0}^{d-1} x^i - dx^d \end{aligned}$$

将$x=\omega_d^j$代入，由于有$\sum_{i=0}^{d-1}\omega_d^{ij}=0$，$\omega_d^d=1$，易得

$$\begin{aligned}\frac{1}{1-\omega_d^j} &= -\frac{1}{d}S\\ &= -\frac{1}{d}\sum_{i=0}^{d-1}(i+1)\omega_d^{ij}\\ &= -\frac{1}{d}\sum_{i=0}^{d-1}i\omega_d^{ij}\end{aligned}$$

有了上述结论，计算$B_d(\omega_d)$就是计算若干卷积，设

$$\left(\sum_{i=0}^{d-1}a_i\omega_d^i\right)\left(\sum_{i=0}^{d-1}i\omega_d^{ij}\right) = \sum_{i=0}^{d-1}b_i\omega_d^i$$

那么

$$\begin{aligned}b_i &= \sum_{k=0}^{d-1}ka_{i-jk}\\ &= \sum_{k=0}^{d-1}(k-1)a_{(i-j)-j(k-1)}+\sum_{l=0}^{d-1}a_l-da_i\\ &= b_{i-j}+\sum_{l=0}^{d-1}a_l-da_i\end{aligned}$$

由于$D_i$两两互质，所以只需要暴力计算出$b_0$，之后可以$O(d)$递推完成卷积。

可以发现

$$\begin{aligned}B_d(\omega_d) &= \sum_{i=0}^{d-1}b_i\omega_d^i\\ B_d(\omega_d^j) &= \sum_{i=0}^{d-1}b_i\omega_d^{ij}\end{aligned}$$

所以插值后的结果必然为

$$B_d(x) = \sum_{i=0}^{d-1}b_ix^i$$

至此，可以$O(N^2D)$完成所有$B_d(x)$的计算，现在考虑计算$B_d(x)$对最终答案的贡献。

$$\frac{B_d(x)}{\sum_{i=0}^{d-1} x^i} = B_d(x)(1-x)\sum_{i=0}^{\infty} x^{id}$$

由于多项式$B_d(x)(1-x)$的阶数不超过$d-1$，所以$x^{id+j}$前的系数为$b_j - b_{j-1}$。

考虑计算$A(x)$部分对答案的贡献，将$A(x)$的系数看作常量，由于

$$\frac{1}{(1-x)^N} = (1-x)^{-N} = \sum_{i=0}^{N} \binom{-N}{i} (-x)^i$$

可知$x^i$前的系数是一个关于$i$的$N$阶多项式，可以暴力求出前$N+1$项的答案，去掉$B_d(x)$部分的贡献，就得到了这个多项式在$1 \sim N+1$处的点值，然后用拉格朗日插值$O(N)$求出$x^C$前的系数。

时间复杂度$O(N^2D + N\lg C)$，空间复杂度$O(ND + \lg C)$。

# 1.51 Room Corner

## 【试题来源】

Codechef FEB 13 ROC

## 【试题大意】

给出一个房间长度为$N$的房间，这个房间中的每个90°角落处都有一个小朋友。

小朋友可以和相邻角落的小朋友交换位置，每秒可以移动一个格子，每个时刻每个小朋友最多只能和另外的一个小朋友交换位置。

给出$T$组询问，求两个小朋友最少要多少时间才能在某处相遇。

这个房间满足，所有水平方向的空格子都是连续的。

数据范围：$5 \le N \le 2500$，$1 \le T \le 10000$

## 【算法介绍】

假设房间的长宽都为$O(N)$，由房间性质可得小朋友的数量也是$O(N)$的。

把小朋友构成的一个环求出，两个小朋友$A, B$相遇一定是在某一条线段$CD$的中点处，耗时为$Max\{AC, DB\} + \frac{CD}{2}$。

设$M$为$CD$的中点，于是$Max\{AC, DB\} + \frac{CD}{2} = MaxAM, MB$，最优值取到的位置可以二分得出。

时间复杂度$O(N^2 + T\log N)$，空间复杂度$O(N^2)$。

# 1.52 Observing the Tree

## 【试题来源】

Codechef FEB 13 QUERY

## 【试题大意】

给出一棵$N$个点的树，以及$M$个操作，强制在线：

- 给一条树上路径上的点加上一个首项为$A$，公差为$B$的等差数列。
- 求一条路径的权值和。
- 把权值还原到某一次修改操作后的状态。

数据范围：$1 \le N, M \le 10^5$，$0 \le A, B \le 1000$

## 【算法介绍】

经典问题，两个等差数列相加仍是等差数列，可以用两个标记维护，路径操作用树链剖分完成。

由于有还原操作，需要可持久化线段树。

时间复杂度$O(N + M\log^2 N)$，空间复杂度$O(N + M\log^2 N)$。

# 1.53 A New Door

## 【试题来源】

Codechef JAN 13 ANDOOR

## 【试题大意】

在一个宽为$W$高为$H$的木板上，有$N$个圆，圆心在$(X_i, Y_i)$，半径为$R_i$。

现在将这些圆全部涂为黑色，求黑色部分圆弧的总长度。

数据范围：$1 \le N, W, H \le 1000$，$0 \le X_i, Y_i \le 1000$，$0 < R_i \le 1000$

## 【算法介绍】

对于每个圆，用余弦定理求出和其他圆相交处的圆弧的极角区间，将这些区间排序后可以统计出有多少极角范围没有和其他圆相交。

时间复杂度$O(N^2 \log N)$，空间复杂度$O(N)$。

## 1.54 Cucumber Boy and Cucumber Girl

### 【试题来源】

Codechef JAN 13 CUCUMBER

### 【试题大意】

给出$B$个$N \times N$的矩阵$Q_i$。

对于每对$(a, b), 1 \leq a < b \leq B$，构造一个新矩阵$C_{a,b}$满足：

$$C_{a,b}[i][j] = \sum_{k=1}^{N} Q_a[i][k]Q_b[j][k]$$

对于所有$1 \sim N$的排列$P$，定义一个新序列：

$$T[i] = C_{a,b}[i][P_i]$$

设$Cnt(a, b)$为$T$中至少有一个奇数的排列$P$的个数，定义数对$(a, b)$是好的，当且仅当$Cnt(a, b)$是奇数。

计算有多少个好的数对。

数据范围：$1 \leq N \leq 60$，$2 \leq B \leq 8000$

### 【算法介绍】

由于只需要关注奇偶性，所以可以放在模2域下计算，设

$$D_{a,b}[i][j] = (C_{a,b}[i][j] + 1) mod\, 2$$

那么需要计算的就是$N! - |D_{a,b}|$，考虑优化计算行列式$|D_{a,b}|$的效率。

$$D_{a,b}[i][j] = \left( \sum_{k=1}^{N} Q_a[i][k]Q_b[j][k] + 1 \right) mod\, 2$$

考虑将$Q$扩展为$N \times N+1$的矩阵，新增一列都为1，那么

$$D_{a,b}[i][j] = \sum_{k=1}^{N+1} Q_a[i][k]Q_b[j][k] \; mod \; 2$$

可以发现$D_{a,b} = Q_a{Q_b}^T$，那么根据Binet-Cauchy定理

$$|D_{a,b}| = \sum_{c=1}^{N+1} |Q_{a,c}||Q_{b,c}|$$

其中$Q_{a,c}$为矩阵$Q_a$去掉第$c$列后的矩阵。

假设已经计算完成了所有$Q_{a,c}$，可以利用位运算$O(B^2)$计算出所有$|D_{a,b}|$模2后的值。

对于一个$N \times N+1$的矩阵$Q_a$，先用高斯消元将其消为最简形式，假如这个矩阵的秩不足$N$，那么所有$|Q_{a,c}|$都为0。

对于秩为$N$的情况，消元后的最简形式一定是一个$N$阶单位矩阵加上一列构成的。

设这一列为$k$，这一列上第$i$行的值为$G[i]$，有如下结论：

- 当$c < k$时，$|Q_{a,c}| = G[c]$
- 当$c = k$时，$|Q_{a,c}| = 1$
- 当$c > k$时，$|Q_{a,c}| = 0$

这样就可以在$O(N^2B)$的时间内计算所有$|Q_{a,c}|$的值。

时间复杂度$O(N^2B + B^2)$，空间复杂度$O(N^2B)$。

## 1.55 Different Trips

### 【试题来源】

Codechef DEC 12 DIFTRIP

### 【试题大意】

给出一棵$N$个点的有根树，根为1，每个节点的权值定义为它的度数。

对于所有从某个节点往祖先方向走的路径，将路径上的点的权值看作一个数字串。

求有多少不同的串。

数据范围：$1 \leq N \leq 10^5$

### 【算法介绍】

从根开始遍历这棵树可以建出一棵Trie，对这棵Trie建出SAM能够直接统计不同的子串数。

可以用可持久化数据结构维护SAM的边表。

时间复杂度$O(N \log N)$，空间复杂度$O(N \log N)$。

# 1.56 Quasi-Polynomial Sum

## 【试题来源】

Codechef DEC 12 QPOLYSUM

## 【试题大意】

对于一个多项式$P(x)=\sum_{k=0}^{D} C_k x^k$，给出$P(0)\sim P(D)$以及一个数$Q$，求

$$\sum_{k=0}^{N-1} Q^k P(k)$$

答案对$M$取模，$M$与$2\sim d+14$互质。

数据范围：$1\le N\le 10^{100000}$，$0\le D<20000$，$1<M<10^{18}$

## 【算法介绍】

设

$$G(N)=\sum_{k=0}^{N-1} Q^k P(k)$$

对于$Q=0$的情况，答案就是$P(0)$。

对于$Q=1$的情况，$G(n)=\sum_{k=0}^{N-1} P(k)$是一个$D+1$阶多项式，可以采用拉格朗日插值

$$G(N)=\sum_{k=0}^{D+1} G(k)\prod_{i=0,i\ne k}^{D+1}\frac{N-k}{i-k}$$

这个式子$O(D)$完成计算，接下来考虑$Q>1$的情况。

可以得到一个$D$阶多项式$F(N)$，使得

$$G(N)=Q^N F(N)-F(0)$$

考虑用数学归纳证明，当$D=0$时，$G(N)=\frac{Q^N-1}{Q-1}P(0)$，$F(N)=\frac{P(0)}{Q-1}$，成立。

假设当$D=d$时成立，考虑证明$D=d+1$的情况。

$$\begin{aligned}(Q-1)G(N) &= (Q-1)\sum_{k=0}^{N-1}Q^kP(k)\\ &= \sum_{k=1}^{N}Q^kP(k-1)-\sum_{k=0}^{N-1}Q^kP(k)\\ &= Q^NP(N-1)+\sum_{k=0}^{N-1}(P(k-1)-P(k))-P(-1)\end{aligned}$$

$P(k-1)-P(k)$是一个$d$阶多项式，所以$\sum_{k=0}^{N-1}(P(k-1)-P(k))$可以表示为$Q^Nf(N)-f(0)$。

那么$(Q-1)G(N)$可以表示为$Q^N(P(N-1)+f(N))-f(0)-P(-1)$，所以

$$F(N)=\frac{P(N-1)+f(N)}{Q-1}$$

$F(N)$是一个$d+1$阶多项式，$D=d+1$时成立。

由于

$$G(N+1)-G(N)=Q^NP(N)=Q^{N+1}F(N+1)-Q^NF(N)$$

得

$$\begin{aligned}P(N) &= QF(N+1)-F(N)\\ F(N+1) &= \frac{F(N)+Q(N)}{Q}\end{aligned}$$

根据这个递推式，可以把$F(1)\sim F(D+1)$表示为关于$F(0)$的线性方程。

由拉格朗日插值公式可以列出方程

$$
\begin{aligned}
F(D+1) &= \sum_{k=0}^{D} F(k) \prod_{i=0,i\neq k}^{D} \frac{D+1-i}{k-i} \\
&= \sum_{k=0}^{D} F(k) \frac{D+1-k}{D+1-k} \prod_{i=0,i\neq k}^{D} \frac{D+1-i}{k-i} \\
&= \sum_{k=0}^{D} F(k)(D+1)! \prod_{i=0}^{D+1} \frac{1}{k-i}
\end{aligned}
$$

即

$$
\sum_{k=0}^{D+1} (-1)^k \binom{D+1}{k} F(k) = 0
$$

通过解这个线性方程可以得到$F(0)$，进而得到$F(0) \sim F(D)$，插值计算出$F(N)$。

在递推中涉及到求$Q$的逆元，并且在最终方程中，$F(0)$前的系数为

$$
\begin{aligned}
\sum_{k=0}^{D+1} (-1)^k \binom{D+1}{k} Q^{-k} &= (1-\frac{1}{Q})^{D+1} \\
&= \frac{(Q-1)^{D+1}}{Q^{D+1}}
\end{aligned}
$$

因此$Q-1$也需要存在逆元。

设$M = m_1m_2m_3$，$gcd(m_2m_3, Q) = 1$，$gcd(m_1m_3, Q-1) = 1$，$u$满足$m_1|Q^u$的最小值，$v$为满足$m_2|(Q-1)^v$的最小值，分别计算出模$m_1, m_2, m_3$域下的答案后，用中国剩余定理合并。

由于$gcd(m_3, Q(Q-1)) = 1$，在模$m_3$域下可以找到$Q$和$Q-1$的逆元，可以直接计算。

考虑模$m_1$域下的情况，$m_1|Q^u$，在计算$\sum_{k=0}^{N-1} Q^k P(k)$时，$k \geq u$的项可以忽略。

考虑模$m_2$域下的情况，$m_2|(Q-1)^v$，且

$$\begin{aligned} Q^kP(k) &= P(k)((Q-1)+1)^k \\ &= P(k)\sum_{i=0}^{k}\binom{k}{i}(Q-1)^i \end{aligned}$$

其中$i \geq v$的项可以忽略，所以

$$\begin{aligned} \sum_{k=0}^{N-1} Q^kP(k) &= \sum_{k=0}^{N-1} P(k)\sum_{i=0}^{k}\binom{k}{i}(Q-1)^i \\ &= \sum_{i=0}^{v-1}(Q-1)^i\sum_{k=0}^{N-1}\binom{k}{i}P(k) \end{aligned}$$

$\binom{k}{i}$是一个关于$k$的$i$阶多项式，$\binom{k}{i}P(k)$是一个$D+i-1$阶多项式，所以在模$m_2$域下，$G(N)$是一个$D+v$阶多项式，可以直接插值计算。

因为$M$与$2 \sim d+14$互质，所以$M$中的最大质因子为17，而$17^{15} > 10^{18}$，所以$u, v \leq 14$。

时间复杂度$O(D+\lg N)$，空间复杂度$O(D)$。

## 1.57 Arithmetic Progressions

### 【试题来源】

Codechef NOV 12 COUNTARI

### 【试题大意】

给出一个长度为$N$的序列$A_i$，求有多少$(i, j, k)$，满足$1 \le i < j < k \le N$且$A_j - A_i = A_k - A_j$。

数据范围：$1 \le N \le 10^5$，$1 \le A_i \le 30000$

### 【算法介绍】

$A_j - A_i = A_k - A_j$即$2A_j = A_i + A_k$，相当于对所有$j$，在其左侧和右侧各选一个数，使得这两个数的和为$2A_j$。

考虑对序列分块，块大小为$S$，将每块左侧和右侧的数用FFT计算卷积，便可以得到块外对块内的贡献，块内的答案可以暴力枚举计算。

复杂度为$O(\frac{N}{S} A \log A + NS)$，故$S$取$\sqrt{A \log A}$最优。

时间复杂度$O(N\sqrt{A \log A})$，空间复杂度$O(N + A)$。

## 1.58 Martial Arts

### 【试题来源】

Codechef NOV 12 MARTARTS

### 【试题大意】

一个完全二分图，边有两个权值$A_{i,j}$和$B_{i,j}$，要进行匹配。

设匹配边的$A$值总和为$H$，$B$值总和为$G$。对手的目的是最大化$G-H$，其次最大化$G$，他会在知道了匹配之后选择是否去掉一条匹配边，使得该边的权值不算入$H$和$G$。

求一个完全匹配，最大化在对手操作后的$H-G$，其次最大化$H$。

数据范围：$1 \le N \le 100$，$0 \le A_{i,j}, B_{i,j} \le 10^{12}$

### 【算法介绍】

设边权为$w(i,j) = A_{i,j} - B_{i,j}$，那么

- 我方的目标是最大化$\sum w(i,j)$，其次最大化$\sum A_{i,j}$。
- 对方的目标是最小化$\sum w(i,j)$，其次最大化$\sum B_{i,j}$，而在$\sum w(i,j)$相同的时候，最大化$\sum B_{i,j}$也可以看作最大化$\sum A_{i,j}$。

对手的决策可以视为，求得的最大匹配中，选取最大$w(i,j)$删去，有多个时选取$A_{i,j}$最小的。

考虑将边按这两个关键字排序，逐条从原图中删去，每次重新计算最大匹配，得到的答案减去图中最大的边就是题设中要求的值。二分图最大权匹配选用KM算法，每次删去相关的匹配边，更新顶标后重新找一次增广路。

时间复杂度$O(N^4)$，空间复杂度$O(N^2)$。

## 1.59 Max Circumference

### 【试题来源】

Codechef OCT 12 MAXCIR

### 【试题大意】

给出一个三角形$ABC$，以及$N$个操作。第$i$个操作有两个参数$x_i, y_i$，使用这个操作可以使得点$A$的$x$坐标增加$x_i$，并且$y$坐标增加$y_i$。

你可以使用最多$K$个操作，这些操作的影响叠加，同一个操作不能重复使用，$ABC$三个点允许共线或重合。

最大化三角形$ABC$的周长，答案的**绝对误差**必须小于$10^{-12}$。

数据范围：$K \le N \le 500$，$|A_x|, |A_y|, |B_x|, |B_y|, |C_x|, |C_y| \le 10^9$，$|x_i|, |y_i| \le 10^6$

### 【算法介绍】

由于$|BC|$是固定的，所以只需要考虑最大化$|AB| + |AC|$，但是这将涉及到平方根的运算，所以考虑从其他方向求解最大值。

设点$A$的最终坐标为$(X, Y)$，可以找到两个实数$u, v$，使得在题设条件下，最大化函数$f(X, Y) = uX + vY$的同时，$|AB| + |AC|$ 也最大化。

设可以达到的$|AB| + |AC|$的最大值为$D$，那么所有合法的操作方案都满足$|AB| + |AC| \le D$，这个图像可以表示为一个焦点为$B, C$的椭圆，最优解$A'$在这个椭圆的圆周上。那么，只需要通过求解$A'$的切线方程就可以找到一组合适的$u, v$了。

分析至此，问题就转化为找到合适的$u, v$，并找到最大化线性函数$f(X, Y) = uX + vY$的方案。

假设已经确定了$u, v$，那么每个操作对函数$f(X, Y)$的贡献是独立的，其值为$f(x_i, y_i)$，排序后取前$K$个正权操作即可。

考虑$f(x_i, y_i) = f(x_j, y_j)$的情况，可以解出两组$(u, v)$，把这两组$(u, v)$看作向量后，它们关于原点对称，分割出的两个半平面将确定$f(x_i, y_i)$和$f(x_j, y_j)$的大小关系。

那么，可以尝试对于每一对操作都求出其相等时的向量，把这些向量按极角排序后，在相邻两个向量夹角中随意选取一个向量作为$(u, v)$，排序计算，这样可以得到一个$O(N^3 \log N)$的算法。

容易发现，重新排序这一步实际上是不需要的，每转过一个向量，只需要将涉及到的操作进行修改，再二分出前$K$个正权操作，前缀和可以在交换相邻操作是$O(1)$更新。

由于是找前$K$个正权操作，所以$f(x_i, y_i)$的正负也会影响操作的选取。考虑$f(x_i, y_i) = 0$的情况，求出的两组向量也需要加入排序。

整个算法的瓶颈在于对$O(N^2)$个向量的排序，以及$O(N^2)$次二分。

由于坐标范围比较大，答案要求绝对误差小于$10^{-12}$，精度误差可能会出现在以下的几个子问题中。

在相邻两个向量夹角中随意选取一个向量：这个问题比较简单，将两个向量相加即能得到一个这样的向量，只要注意这两个向量共线的情况。

把向量按极角排序：$atan2$函数的精度并不能满足需求，可以将象限分开，用叉积排序。

平方根运算：答案要求绝对误差小于$10^{-12}$，而点的坐标范围为$10^9$级别，需要特殊的开平方根算法来保证精度。

设$sqrt(S) = I + D$，其中$I$为整数部分，$D$为小数部分，将$I$和$D$分开计算，其

中$D$的计算较为复杂：

$$\begin{aligned} I^2 + 2ID + D^2 &= S \\ D &= \frac{S - I^2}{2I + D} \\ D &= \frac{S - I^2}{I + \sqrt{S}} \end{aligned}$$

上式中的$\sqrt{S}$可以直接使用内置函数计算。

时间复杂度$O(N^2 \log N)$，空间复杂度$O(N^2)$。

## 1.60 Ciel and password cracking

### 【试题来源】

Codechef OCT 12 CIELHACK

### 【试题大意】

Ciel有一个保险箱，这个保险箱有$K$个密码，然而她只记得第$i$个密码是一个不超过$N_i$的正整数，她打算进行暴力破解。

由于Ciel的电脑太慢了，所以她打算借用市里的电脑中心。市里共有$C$个电脑中心，Ciel打算借用其中恰好$K$个来分别破解$K$个密码。

这$C$个电脑中心和Ciel的餐馆在同一条街上，以Ciel的餐馆为坐标轴零点，第$i$个电脑中心的坐标为$X_i$，坐标的正负表示所在的方向。

Ciel将会选择$K$个不同的电脑中心去破解她的$K$个密码，她会以任意顺序去遍历它们，从她的餐馆出发，最后回到她的餐馆。

Ciel的行走速度为$V$，即从坐标$A$走到坐标$B$需要消耗$\frac{|A-B|}{V}$单位时间。

在第$j$个电脑中心，Ciel可以使用最多$P_j$台电脑，每台电脑需要消耗$T_j$单位时间去枚举验证一种密码。这些电脑被一种奇怪的网络连接着，于是Ciel的计划将以下面的步骤进行：

- 选择一部分电脑（不超过$P_j$台），这一步不消耗时间。
- 在一台电脑中安装破解程序，这一步不消耗时间。
- 把破解程序通过网络传输到其他被选择的电脑中，这一步的消耗时间在后文会详述。
- 在所有被传输的电脑上同时运行破解程序，这一步的消耗时间在后文会详述。

- 一旦其中一台电脑成功破解了密码，那么这里的工作将会全部停止，然后她将会奔赴下一个电脑中心。

接下来将描述这个问题的更多细节。

**破解程序细节**

破解程序每次会随机验证一个没有被验证过的密码，对于第$j$个电脑中心的电脑，验证一次将消耗$T_j$单位的时间。

假如某一次验证的密码成功了，程序将通知Ciel并终止。

电脑和电脑之间并没有共享信息，即可能会验证在其他电脑上已经验证过的密码。

**网络连接细节**

通过网络连接，已经装有破解程序的电脑，可以将破解程序备份传输到没有安装的电脑上。

连接所需要的时间是随机的，这个随机值呈指数分布，对于第$j$ 个电脑中心的电脑，均值为$S_j$。

只有已经装有破解程序的电脑才可以与其他电脑连接，一旦连接完成，备份传输可视为不消耗时间。

在每个时刻，一台电脑只能尝试向一台电脑连接，但一台电脑可以同时接受多台电脑的连接，一旦其中一个连接成功确立，那么其他连接将会中断停止，那些电脑将会去尝试与其他电脑连接。

**连接方案细节**

可以为每台电脑预先制定一个连接列表，电脑每次会选择连接列表上的第一个没有装有破解程序的电脑进行连接。一旦连接确立或中断，它将立刻选择下一个连接目标。

在所有被选择的电脑都装有破解程序后，所有程序将同时开始运行。

Ciel想要通过分配电脑中心，选择电脑，调整连接方案，来最小化期望所需时间。

答案的相对误差要求小于$10^{-6}$。

数据范围：$1 \le C \le 1000$，$1 \le K \le min(5, C)$，$1 \le V, S_j, T_j \le 10^{20}$，$1 \le N_i, P_i \le 10^{18}$，$|X_i| \le 10^{20}$

## 【算法介绍】

设$Time[i][j]$为第$i$个密码在第$j$个电脑中心破解的最小期望所需时间，假设这个已经完成了计算，那么只需要一个简单的状压DP就可以解决选择电脑中心的问题，所以主要问题是计算$Time[i][j]$。

计算$Time[i][j]$涉及到密码范围$N_i$，电脑数量$P_j$，连接时间$S_j$，以及运算速度$T_j$，设$Time[i][j] = F(N_i, P_j, S_j, T_j)$，现在需要快速计算函数$F(N, P, S, T)$。

设$A(k)$为假设$S = 1$的前提下，连接$k$台电脑的最小期望时间，$B(N, k)$为假设$T = 1$的前提下，用$k$台电脑破解范围为$N$的密码的期望时间。那么

$$F(N, P, S, T) = \min_{k=1}^{P} \left( \, S * A(k) + T * B(N, k) \, \right)$$

现在问题转化为计算$A(k)$和$B(N, k)$，再求得最小值来计算$F(N, P, S, T)$。

**计算A(k)**

由于连接所需的时间是指数分布的，所以具有无记忆性，即设函数返回值为$R$，对于两个数$a, b \ge 0$，$P(R > a + b \mid R > b) = P(R > a)$。

根据无记忆性，连接列表中的顺序并不会影响连接完成所需的期望时间，所以可以假定所有电脑的连接列表都为$1 \sim k$。

根据指数分布函数的性质，$m$个参数为$\lambda_1, \lambda_2, ..., \lambda_m$的指数分布函数取$min$后是一个参数为$\lambda = \sum_{i=1}^{m} \lambda_i$的指数分布函数。所以$m$台电脑同时连接一台电脑的期望连接时间为原先的$\frac{1}{m}$倍，这个结论通过无记忆性也可以直观地得到。

由此可以得到

$$A(k) = \sum_{i=1}^{k-1} \frac{1}{i}$$

可以预处理$k \le 10^6$的情况，对于$k > 10^6$的情况

$$\sum_{i=1}^{k-1} \frac{1}{i} \approx \ln(k-1) + \gamma$$

其中$\gamma$为欧拉常数。

**计算B(N,k)**

考虑需要进行第$i$次验证的概率为$\left(\frac{N-i+1}{N}\right)^k$，根据期望的线性贡献易得

$$B(N,k) = \sum_{i=1}^{N} \left(\frac{i}{N}\right)^k$$

考虑设定阈值，设$t = \frac{k+1}{N}$，当$t < 2$时，使用欧拉-麦克劳林公式。

$$\sum_{i=m+1}^{n} f(i) - \int_m^n f(x)dx \ \approx \ \sum_{i=1} \frac{B_i}{i!} \left(f^{(i-1)}(n) - f^{(i-1)}(m)\right)$$

其中$B_i$为第$i$项伯努利数，由于伯努利数除第1项外的奇数项都为0，所以这个式子可以写为

$$\sum_{i=m+1}^{n} f(i) - \int_m^n f(x)dx \ \approx \ B_1(f(n) - f(m)) + \sum_{i=1} \frac{B_{2i}}{(2i)!} \left(f^{(2i-1)}(n) - f^{(2i-1)}(m)\right)$$

考虑对于函数$f(x) = \left(\frac{x}{N}\right)^k$套用这个公式，并取阈值$p = min(10, \frac{k}{2})$，计算

$$B(N,k) \approx \frac{N}{k+1} + \frac{1}{2} + \sum_{i=1}^{p} \frac{B_{2i}}{(2i)!} \prod_{j=0}^{2i-2} \frac{k-j}{N}$$

这样计算的相对误差不超过$3\left(\frac{t}{2\pi}\right)^{2p}$，所以可以取$p = 10$。

当$t \geq 2$时，大多数项都比较小，取$p = min(n-1, 10)$。

$$B(N,k) \approx \sum_{i=0}^{p} \left(\frac{N-i}{N}\right)^k$$

这样计算的相对误差不超过$\frac{e^{-pt}}{t}$，所以可以取$p = 10$。

**计算F(N,P,S,T)**

设$G(k) = S * A(k) + T * B(N,k)$，可以发现这是一个单峰函数，考虑计算

$$G(k+1) - G(k) = \frac{S}{k} - T(B(N,k) - B(N,k+1))$$

通过二分并判断差值可以快速找到极值点。

**误差处理**

虽然只要求相对误差小于$10^{-6}$，但是还是存在精度误差的问题。

考虑在计算$B(N,k)$的过程中，$t \geq 2$时，需要计算$\left(\frac{N-p}{N}\right)^k$。由于$k$的取值范围会达到$10^{18}$，通常情况下可以计算$\exp(k \ln \frac{N-p}{N})$来完成，但当$\frac{p}{N}$非常小时会有较大误差。

考虑将$ln(1-x)$泰勒展开

$$ln(1-x) \approx -x - \frac{x^2}{2} - \frac{x^3}{3} - ...$$

可以在$\frac{p}{N} < 10^{-9}$时，计算$\exp\left(-k(\frac{p}{N} + \frac{\left(\frac{p}{N}\right)^2}{2})\right)$来保证精度。

在计算$F(N,P,S,T)$过程中，二分时需要计算$G(k+1) - G(k)$，也就是要计算$B(N,k) - B(N,k+1)$，这时分开计算$B(N,k)$和$B(N,k+1)$是不行的。

同样设阈值$t = \frac{k+1}{N}$，当$t < 2$时

$$B(N,k) - B(N,k+1) \approx \frac{N}{(k+1)(k+2)} + \sum_{i=1}^{p} \frac{B_{2i}}{(2i)!} \left(\prod_{j=0}^{2i-2} \frac{k-j}{N} - \prod_{j=0}^{2i-2} \frac{k+1-j}{N}\right)$$

其中，对于$i > 1$的情况

$$\begin{aligned}\prod_{j=0}^{2i-2}\frac{k-j}{N}-\prod_{j=0}^{2i-2}\frac{k+1-j}{N} &= \prod_{j=0}^{2i-2}\frac{k-j}{N}-\prod_{j=-1}^{2i-3}\frac{k-j}{N}\\ &= \frac{(k-2i+2)-(k+1))}{N}\prod_{j=0}^{2i-3}\frac{k-j}{N}\\ &= -\frac{2i-1}{N}\prod_{j=0}^{2i-3}\frac{k-j}{N}\end{aligned}$$

当$t \geq 2$时

$$B(N,k)-B(N,k+1)\approx\sum_{i=1}^{p}\frac{i}{N}\left(\frac{N-i}{N}\right)^k$$

这样计算即可保证在二分时所需的精度。

时间复杂度$O(CK\log P+2^KCK)$，空间复杂度$O(CK+2^KC)$。

## 1.61 Knight Moving

### 【试题来源】

Codechef SEPT 12 KNGHTMOV

### 【试题大意】

在一个无限大小的棋盘上，在$(0,0)$位置有一个骑士，现在需要将其移动到$(X,Y)$。

骑士每次可以从$(u,v)$移动到$(u+A_X, v+A_Y)$或$(u+B_X, v+B_Y)$。

棋盘上还有$K$个障碍$(x_i, y_i)$，骑士不能经过这些位置，保证$(0,0)$和$(X,Y)$不会是障碍。

求这个骑士有多少种移动方法，骑士到达$(X,Y)$后也可以继续移动，但最后必须停在$(X,Y)$。

两种移动方法不同必须满足，至少存在一个$i$使得，第$i$步所在的位置不同，答案为$\infty$时输出-1。

数据范围：$0 \le K \le 15$，$|X|,|Y|,|A_X|,|A_Y|,|B_X|,|B_Y|,|x_i|,|y_i| \le 500$

### 【算法介绍】

考虑两个移动向量$A(A_X,A_Y),B(B_X,B_Y)$，当向量$A$和向量$B$线性无关时，所有由这两个向量可达的位置都可以唯一表示为$uA+vB$。

可以通过求解方程得出每个障碍和终点所对应的$(u,v)$，接下来转化为一个经典问题。

将点按坐标排序后，设$f_k$为不经过其他障碍到达这个位置的方案数，易得

$$f_k = ways(0,k) - \sum_{i=1}^{k-1} ways(i,k) f_i$$

所以当向量$A$和向量$B$线性无关时，可以$O(K^2)$完成，考虑当向量$A$和向量$B$线性相关的情况，即这两个向量平行。

问题可以从二维转化为一维，设坐标的范围为$O(d)$，有效的坐标范围仅为$O(d^2)$，只要处理这些坐标就能求出方案数或者判断无限解。

对于这些坐标，可以根据移动规则建出一张图，假如方案数的有限的，那么和终点相关的子图一定是拓扑图，可以直接DP求解，复杂度为$O(d^2)$。

时间复杂度$O(K^2 + d^2)$，空间复杂度$O(K + d^2)$。

## 1.62 Annual Parade

### 【试题来源】

Codechef SEPT 12 PARADE

### 【试题大意】

给出一张$N$个点$M$条边的有向图，在这张有向图上求一些路径，使得以下费用之和最小：

- 路径中的所有边将造成费用$V_i$，如果一条边被多条路径覆盖到就贡献多次。
- 没有成环的路径将消耗$C$的费用。
- 没有被如何路径覆盖到的点将贡献$C$的费用。

对于这张图有$K$组询问，每组询问的$C$不同。

数据范围：$2 \le N \le 250$，$1 \le M \le 30000$，$1 \le K \le 10000$，$1 \le V_i, C \le 10000$

### 【算法介绍】

问题可以转化为，在原图的任意两点间加上一条边权为$C$的边（包括自环），然后在图上求出一些环覆盖所有点，并使环长之和尽量小。

这是一个经典问题，可以先用Floyd求出任意两点间的最短路，每个点$u$拆成$u$和$u'$，在每一对$u, v'$间连上权值为$u, v$间最短路长度的边，最小环长和即为二分图的最小权重匹配。

由于费用流每增加1的流量，费用的增量是单调的，所以对于多组询问，把每次增加流量时的费用都记下来，对于没有满流的部分用费用为$C$ 的边，每组询问可以$O(N)$解决，费用流部分只需要跑一次。

点规模$O(N)$，边规模$O(N^2)$。

## 1.63 Two Magicians

### 【试题来源】

Codechef AUG 12 MAGIC

### 【试题大意】

有两个人在一张$N$个点$M$条边的无向图上博弈，一开始先手在1号点，后手在2号点。

每个人每次需要执行以下流程：

- 沿着已经存在的无向边走任意条，如果走到了对方所在的点则取得胜利。
- 添加一条不存在的无向边，如果不能添加对方取得胜利。
- 每人共有$P$次机会，能够传送到任意一个点上。

求在最优策略下，是先手必胜还是后手必胜。

数据范围：$2 \le N \le 7777$，$0 \le M, P \le 10000$

### 【算法介绍】

如果两个人一开始就在同一个连通块，显然先手必胜。

设$E$为一开始图中还没连的边数。

当$N$为奇数时，当图中只剩两个连通块时，这两个连通块之间的边数一定为偶数。所以当$E$为奇数时，先手可以保证连通最后两个连通块的那次加边一定是后手操作的，即先手必胜。反之，后手必胜。

接下来只考虑$N$为偶数的情况，此时大小为奇数的连通块个数一定是偶数。

先考虑$P = 0$的情况，即没有传送操作。

假如$E$为奇数，那么为了使得连通最后两个连通块的加边为后手操作，必须使得最后两个连通块大小都为偶数。当一开始1或2所在连通块大小为偶数时，先手可以利用奇数连通块为偶数个的性质保证它的大小一直为偶数，此时先手必胜。否则后手可以使得这两个连通块的大小一直为奇数，后手必胜。

$E$为偶数的情况也可以类似得到，这时先手必胜的条件为一开始1或2所在连通块大小为奇数，接下来考虑$P \geq 1$的情况。

假如$E$为奇数，那么先手的目标为自己所在的连通块大小为偶数。假如一开始就存在偶数连通块，那么先手可以通过1次传送来构造出和$P = 0$ 时相同的必胜策略。假如一开始没有偶数连通块，那么只要连通块个数$> 2$，先手就可以通过一次传送避免留在奇数连通块内。所以在这种情况下，只有一开始只有2个奇数连通块时才是后手必胜。

假如$E$为偶数，那么先手的目标为自己所在的连通块大小为奇数。唯一先手必胜的情况为，一开始图中只有2个奇数连通块以及最多1个偶数连通块，先手可以移动到奇数连通块上，并将另一个奇数连通块与唯一的偶数连通块合并。

时间复杂度$O(M + \alpha N)$，空间复杂度$O(N)$。

## 1.64 A Game of Thrones

### 【试题来源】

Codechef AUG 12 GTHRONES

### 【试题大意】

有两个人在进行一种博弈游戏：

- 一开始，有$N$种数字$u_i$，每种有$c_i$个。
- 在第一回合中，先手选择一个数字，把它作为游戏的当前数字。
- 之后从第二回合开始执行当前回合的人都按如下操作：现在的当前数字为$u$，将这个$u$除去。然后选择另一个数字$v$作为当前数字，$v$要满足与$u$刚好相差一个质因子。
- 无法完成操作的人输。

求先手是否可以必胜，如果先手必胜再输出在第一回合可以选择的最小数字。

数据范围：$1 \le N \le 500$，$1 \le u_i \le 10^{18}$，$1 \le c_i \le 10^9$

### 【算法介绍】

可以将这些数建成二分图，在恰好差一个质因子的数间连边，边数最多为$O(N \log u)$条，连边的判断可以$O(N^2 \log u)$用MillerRabin素性测试完成。

这是一类经典的博弈游戏，后手必胜的条件为这张图存在一个完美匹配。

先手在第一回合可以选择的数为那些可以不存在于最大匹配中的数，这个可以用退流的方式判断，即判断是否可以找到一条等效增广路使得最大匹配不变。

点规模$O(N)$，边规模$O(N \log u)$。

## 1.65 Dynamic GCD

### 【试题来源】

Codechef JULY 12 DGCD

### 【试题大意】

给出一棵$N$个点的树，第$i$个点的权值为$A_i$，有$M$个操作，操作有两种：

- 将$(u, v)$路径上的所有点的点权都加上$X$。
- 询问$(u, v)$路径上的所有点的点权的$gcd$。

数据范围：$N \le 50000$，$A_i, X \le 10000$

### 【算法介绍】

由于$gcd(A, B, C) = gcd(A, B - A, C - B)$，所以对于一条树链的加权操作可以用差分完成，也就是维护每个点与其父亲的权值差。

考虑使用树链剖分，每条重链上用线段树维护差分，每个询问可以分解为若干条重链上的区间询问。

由于$gcd$的变化次数为$O(\log X)$次，所以在一次线段树操作中$gcd$对复杂度的贡献最多也只有$O(\log X)$。

时间复杂度$O(N + M \log N(\log N + \log X))$，空间复杂度$O(N)$。

## 1.66 Cool Numbers

### 【试题来源】

Codechef JUNE 12 COOLNUM

### 【试题大意】

对于一个数$X$，假设它共有$K$位，各位数字为$X_1, X_2, \ldots, X_K$，假如可以选出最多三位使得这几位数字之和为$S$，满足$(\sum_{i=1}^{K} X_i - S)^S$是$X$的倍数，则称$X$是一个cool number。

对于给定是数$N$，求不大于$N$的最大的cool number以及大于$N$的最小的cool number。

数据范围：$1 \le N \le 10^{1000}$

### 【算法介绍】

cool number可以分为两种，第一种是只含有不超过三个非零位的数，这一部分可以用贪心直接得到。

对于剩下的情况，考虑满足$(9K)^{27} > 10^K$的$K$，可以发现$K$不超过77，所以这一类cool number不超过$10^{77}$，而且实际上数量也非常少，可以通过枚举$(\sum_{i=1}^{K} X_i - S)$的值，并对每个不超过$10^{77}$的$(\sum_{i=1}^{K} X_i - S)^{27}$的约数进行验证来预处理。

假设已经预处理出了第二类cool number，那么询问只要在排序后的序列上二分即可。

不考虑预处理的时空复杂度，设$S$为第二类cool number的数量，时间复杂度$O(\lg N + \log S \lg N)$，空间复杂度$O(\lg N)$。

# 1.67 Expected Maximum Matching

## 【试题来源】

Codechef JUNE 12 MATCH

## 【试题大意】

一张左侧$N$个点右侧$M$个点的二分图，点$i$和点$j$间有边相连的概率为$P_{i,j}$。

求这张二分图的期望最大匹配数。

数据范围：$1 \le N \le 5$，$1 \le M \le 100$

## 【算法介绍】

对于每个右侧点，它与左侧的连边情况有$2^N$种。

在最终的匹配中，有用的右侧节点数量最多为$N$个，可以搜出这$2^{N^2}$种匹配状态，可以证明所有可能的转移都包含在这些状态中。

经去重后还剩$28w+$种状态和$150w+$种转移，可以用简单的DP计算期望值。

时间复杂度$O(2^{N^2}M)$，空间复杂度$O(2^N+NM)$。

# 1.68 Little Elephant and Boxes

## 【试题来源】

Codechef MAY 12 LEBOXES

## 【试题大意】

有$n$个盒子，第$i$个盒子里有$P_i$的概率是$V_i$块钱，$1-P_i$的概率为一块钻石。

打开所有盒子之后用那些钱和钻石去买东西，一共有$m$件物品，第$j$件需要$C_j$块钱以及$D_j$个钻石。

求在最优策略下，期望能买到多少物品。

数据范围：$2 \le n \le 30$，$1 \le m \le 30$，$1 \le V_i, C_j \le 10^7$，$0 \le D_j \le 30$

## 【算法介绍】

可以$O(nm^2)$预处理出$f_{i,j}$为在有$j$颗钻石的前提下买$i$个物品所需的最少钱数。

考虑折半搜索，设置阈值$k$。

对于前$k$个盒子，可以$O(2^k)$搜出所有状态并将其按钻石数量分组后再按钱数排序。

对于后$n-k$个盒子，$O(2^{n-k})$搜出所有状态，对于每个状态枚举对应前$k$个盒子的钻石数和目标要买的物品数，二分出合法的概率累加入答案。

时间复杂度$O(nm^2 + k2^k + mk^2 2^{n-k})$，空间复杂度$O(nm + 2^k)$。

## 1.69 Selling Tickets

### 【试题来源】

Codechef MAY 12 TICKETS

### 【试题大意】

有$N$盘菜，$M$个人，每人喜欢其中的两盘菜$A_i$和$B_i$。

求一个最大的$K$使得，不管怎么从这$M$人中选出$K$人，都能分给他们每人一盘自己喜欢的菜。

数据范围：$2 \le N \le 200$，$0 \le M \le 500$

### 【算法介绍】

将菜看作点，人看做边，问题转化为求一个最大的$K$使得，所有$K$条边的子图中的点数都$\ge K$。

更进一步地，可以通过找出一个最小的子图使得它含有$K$个点$K+1$条边，这种子图有两种：

- 两个点间的三条不相交简单路径。
- 被一条链连接的两个环。

对于第一种情况，可以$O(N^2)$枚举那两个点，再$O(M)$BFS求出这两个点间最短的三条简单路径，容易发现简单路径相交的情况不会影响答案。

对于第二种情况，枚举中间链上的一点，由BFS建出一棵树，选择两条非树边，这两条非树边分别与它所对应的树上路径构成两个环，LCA可以$O(N)$暴力完成。

时间复杂度$O(N^2M)$，空间复杂度$O(N+M)$。

## 1.70 Substrings on a Tree

### 【试题来源】

Codechef APRIL 12 TSUBSTR

### 【试题大意】

给出一棵$N$个点的有根树，每个点上有一个小写字母，对于所有从祖先往儿子方向的路径所构成的字符串中完成以下询问：

- 求出有多少种不同的串。
- $Q$次询问，对于给定的字典序表求出第$K_i$小的串。

数据范围：$1 \le N \le 250000$，$1 \le Q \le 50000$，$1 \le K_i \le 2^{63} - 1$

### 【算法介绍】

对这棵树建出字母树后建立SAM。

SAM中每一条路径都对应这一个不同的子串，可以利用这个性质$O(N)$预处理出自动机中每个点开始的路径数。

对于第二种询问，只要根据给定的字典序表在SAM上遍历，逐位确定，复杂度只与输出量有关。

时间复杂度$O(N)$，空间复杂度$O(N)$。

## 1.71 Find a special connected block

### 【试题来源】

Codechef APRIL 12 CONNECT

### 【试题大意】

给出一个$N \times M$的矩阵，每个格子上填着从$-1 \sim N \times M$中的一个数。

你的任务是找到一个四连通的连通块，在这个连通块里，至少要包含$K$个不同的正数，且不能有-1。

给出选取每一个格子所需要的代价，要求选出的连通块所包含的格子的代价和最小。

数据范围：$1 \le N, M \le 15$，$1 \le K \le 7$

### 【算法介绍】

随机将原矩阵中的权值分配到$1 \sim K$中，这样有$\frac{K!}{K^K}$的概率会使得分配符合最优解。

对于仅剩$K$种权值的图，考虑利用状态压缩计算，类似最小斯坦纳树问题，在每个点上$O(3^K)$合并子集，并在状压的每一层用最短路模型计算最优值，单次计算的复杂度为$O(3^K NM)$。

经过多次随机后就能得到最优解。

时间复杂度$O(\frac{K^K}{K!}3^K NM)$，空间复杂度$O(2^K NM)$。

# 1.72 Evil Book

## 【试题来源】

Codechef MARCH 12 EVILBOOK

## 【试题大意】

有$N$个人，打败第$i$个人需要付出$C_i$的代价，能够获得$M_i$的魔力。

可以通过消耗$X$单位的魔力使一个人的$C_i$和$M_i$都缩减为原来的$\frac{1}{3}$。

魔力必须时时刻刻是非负的，并且最终魔力必须达到666以上，问最少需要付出多少代价。

数据范围：$1 \le N \le 10$，$10 \le X \le 666$，$0 \le C_i, M_i \le 10^9$

## 【算法介绍】

考虑用搜索枚举计算。

对于一个人$i$，如果$M_i \ge 666$，那么打败这个人后这个搜索状态将直接结束，所以在计算复杂度时只考虑$M_i < 666$的情况。

假设在这之后一共进行了$k$次消减操作，那么为了使得解最优，一定有$kX < \frac{M_i}{3^k}$，容易发现$k$的取值在$[0, 2]$范围内，所以总状态数最多为$O(3^N)$。

时间复杂度$O(3^N N)$，空间复杂度$O(N)$。

# 1.73 Ciel and Earthquake

## 【试题来源】

Codechef MARCH 12 CIELQUAK

## 【试题大意】

在一个$R \times C$的方格图上，相邻两个格子间有一个墙壁，这个墙壁有$p$的概率会崩塌。

求$(1,1)$和$(R,C)$连通的概率。

数据范围：$1 \leq R \leq 8$，$1 \leq C \leq 10^{18}$，$0.1 \leq p \leq 1$

## 【算法介绍】

考虑使用轮廓线状压DP，当计算到$(r,c)$时，记录$(1,c) \sim (r,c)$和$(r+1,c-1) \sim (R,c-1)$的连通状态，并记录哪个连通块与$(1,1)$连通，状态数为$R$乘第$R$项贝尔数$B_R$。

根据这个问题的性质，可以进一步缩减这个状态数：

- 在方格图的轮廓线DP中不可能出现ABAB的连通情况。
- 设$X$为与$(1,1)$连通的连通块，不可能出现AXA的情况。

对于$R=8$的情况，状态数最后为3432种，在$C$较小的时候可以通过预处理转移（转移有$O(R^2B_R)$种），直接$O(R^2CB_R)$DP计算解决。

对于$C$较大的情况，设$F_C$为在固定的$R$和$p$下对应$C$的答案，$\frac{F_C}{F_{C-1}}$和$\frac{F_{C-1}}{F_{C-2}}$是近似相等的，可以设定阈值$k=40$，对于$C>k$的情况利用比值从$F_k$推导。

时间复杂度$O(R^2CB_R)$，空间复杂度$O(R^2B_R)$。

## 1.74 Find a Subsequence

### 【试题来源】

Codechef FEB 12 FINDSEQ

### 【试题大意】

给出一个长度为$N$的数组$A$，以及一个字符串$B$，$B$是一个$1 \sim 5$的排列。

找出一个$A$的子序列，使其相对大小关系符合串$B$的描述。

数据范围：$5 \le N \le 1000$，$|A_i| \le 10^9$

### 【算法介绍】

由于只比较相对大小，所以可以通过离散化将$A_i$的范围缩减到$[1, n]$。

考虑枚举第2个和第4个数的位置，第1个和第5个数根据它们和第3个数的大小关系贪心选取，这样就确定了其中4个数的位置，剩下的第3个数的合法范围也已知了，暴力实现是$O(N^3)$的。

可以发现在$O(N^2)$的枚举后，每次需要$O(N)$求解的是在前缀或后缀中，在一定范围内最大或最小的数，可以$O(N^2)$预处理来做到$O(1)$调用。

时间复杂度$O(N^2)$，空间复杂度$O(N^2)$。

## 1.75 Flight Distance

### 【试题来源】

Codechef FEB 12 FLYDIST

### 【试题大意】

给一张$N$个点$M$条边的带权无向图，第$i$条边连结$(A_i, B_i)$的权值为$D_i$。

为了让每条边成为它所连结的两点的最短路，需要对这些边的边权进行一些修改，可以将任意边的权值增加或减少任意的$d$，代价为$d$。

求最小代价，用分数表示。

数据范围：$1 \le N \le 10$，$M \le \frac{N(N-1)}{2}$，$1 \le D_i \le 20$

### 【算法介绍】

设两点间的最短路长度为$f_{i,j}$，第$i$条边增加了${d^+}_i$的权值，减少了${d^-}_i$的权值，根据要求可以列出一些线性规划约束：

$$\begin{aligned} f_{x,y} &\le f_{x,z} + f_{z,y} \\ f_{A_i,B_i} &\le D_i + {d^+}_i - {d^-}_i \\ f_{A_i,B_i} &\ge D_i + {d^+}_i - {d^-}_i \end{aligned}$$

其中，目标为最小化

$$\sum_{i=1}^{M} {d^+}_i + {d^-}_i$$

可以使用单纯形算法对其进行求解，但是所有变量都取0的情况并不是一组合法的初始解。

设$h_{i,j}$为$(i,j)$间最少经过的边数，那么可以用$20h_{x,y}-f_{x,y}$代换$f_{x,y}$，用$19-{d^+}_i$代换${d^+}_i$，用$D_i-1-{d^-}_i$代换${d^-}_i$。

经过这样的代换后，相当于把所有边的初始权值都设为了20，这样的初始解是符合条件的，同时还需要增加约束：

$$\begin{aligned} {d^+}_i &\le 19 \\ {d^-}_i &\le D_i - 1 \end{aligned}$$

变量规模$O(N^2+M)$，不等式规模$O(N^3+M)$。

## 1.76 Card Shuffle

### 【试题来源】

Codechef JAN 12 CARDSHUF

### 【试题大意】

有一个$N$张牌的牌堆，一开始从堆顶到堆底的牌的顺序为$1 \sim N$，进行$M$次操作，每次操作给出三个参数$A, B, C$：

- 从牌堆顶端拿走A张牌。
- 再从牌堆顶端拿走B张牌。
- 将第一步拿走的A张牌放回到剩下的牌堆上面。
- 从牌堆顶拿走C张牌。
- 将第二步你拿起的B张牌一张一张放到牌堆顶。
- 最后，将剩下的C张牌放回到牌堆顶。

从堆顶到堆底输出最后的牌堆。

数据范围：$1 \leq N, M \leq 100000$，$0 \leq A, B, C, A+B, B+C \leq N$

### 【算法介绍】

用平衡树模拟，需要区间翻转标记。

时间复杂度$O(M \log N + N)$，空间复杂度$O(N)$。

## 1.77 Misinterpretation 2

### 【试题来源】

Codechef JAN 12 MISINT2

### 【试题大意】

定义一种字符串的变化方式为将所有偶数位置的字符靠左放置，将所有奇数位置的字符靠右放置。

求有多少长度在$[L, R]$范围内的由小写字母构成的字符串，经过这种变换后仍与原串相同。

答案对$10^9+7$取模。

数据范围：$1 \le L \le R \le 10^{10}$，$R-L \le 50000$

### 【算法介绍】

考虑将这种变换看成一个置换，设长度为$X$的串变换时的置换环个数为$G(X)$，它对答案的贡献即为$26^{G(X)}$。

设位置$u$置换后的位置为$T(u)$，那么有

- 当$X$为奇数时$T(u) \equiv 2u \ (mod\ X)$。
- 当$X$为偶数时$T(u) \equiv 2u \ (mod\ X+1)$，且$G(X) = G(X+1) - 1$。

只需要考虑$X$为奇数的情况，偶数部分的答案可以直接由奇数部分得到。

对于所有满足$gcd(X, u) = v$的$u$，他们所在的置换圈长度是一样的，为$ord_2(\frac{X}{v})$，而满足这个条件的$u$的个数为$\varphi(\frac{X}{v})$，由此可得答案为

$$\sum_{d|X} \frac{\varphi(d)}{ord_2(d)}$$

考虑把所有$[L,R]$质因数分解后，用搜索高效枚举所有约数计算，约数个数可以粗略看作$O((R-L)\log R)$。

质因数分解部分，可以先预处理出$O(\sqrt{R})$范围内的所有质数，去筛$[L,R]$，筛完剩下的部分一定是那些数中的大质数因子，这一部分可以$O(\sqrt{R}\log R)$完成。

考虑搜索时计算$\varphi(d)$和$ord_2(d)$，$\varphi(d)$可以直接利用积性计算，接下来考虑如何计算$ord_2(d)$。

对于$a,b$满足$gcd(a,b)=1$，$ord_2(ab)=lcm(ord_2(a),ord_2(b))$，所以只需要考虑计算$d$只存在单种质因子的$ord_2(d)$。

由于$ord_2(d)|\varphi(d)$，可以将初值设为$\varphi(d)$，然后对所有$\varphi(d)$的质因子进行试除，用快速幂验证，可以$O(\log^2 d)$完成一次计算。

设$p$为一个质数，可以通过增加质因子$p$来由$ord_2(p^k)$计算$ord_2(p^{k+1})$。

对于$O(\sqrt{R})$范围内的$d$，可以$O(\sqrt{R}\log R)$完成预处理，考虑计算那部分大质因子。

设$p$为$N$的一个大质因子，高效计算$ord_2(p)$需要分解$p-1$的质因子。

考虑$(p-1)\frac{N}{p}=N-\frac{N}{p}$，而$\frac{N}{p}\le\sqrt{R}$，只需要扩大一开始的分解范围即可。

时间复杂度$O(\sqrt{R}\log R+(R-L)\log^2 R)$，空间复杂度$O(\sqrt{R}+R-L)$。

## 1.78 Short II

### 【试题来源】

Codechef DEC 11 SHORT2

### 【试题大意】

给出一个质数$p$，求有多少正整数对$(a,b)$满足$a,b>p$并且$(a-p)(b-p)\mid ab$。

数据范围：$1<p<10^{12}$

### 【算法介绍】

问题可以转化为求$ab\mid p(a+b+p)$，其中$a,b>0$，推导如下：

$$(a-p)(b-p)\mid ab \;\Rightarrow\; ab\mid (a+p)(b+p) \;\Rightarrow\; ab\mid p(a+b+p)$$

当$p|a$并且$p|b$时，设$a=pa'$，$b=pb'$，则有

$$ab\mid p(a+b+p) \;\Rightarrow\; p^2a'b'\mid p^2(a'+b'+1) \;\Rightarrow\; a'b'\mid a'+b'+1$$

满足这个条件的正整数对$(a',b')$只有5对。

当$p\nmid a$并且$p\nmid b$时，必须满足$ab\mid a+b+p$，那么

$$\begin{aligned} ab &\le a+b+p \\ ab-a-b+1 &\le p+1 \\ (a-1)(b-1) &\le p+1 \end{aligned}$$

对于$a=b$的情况，只有$(1,1)$一组合法，那么假设$a<b$，则有$a<\sqrt{p+1}+1$。

设$kab=a+b+1$，那么

$$\begin{aligned} kab &= a+b+p \\ (ka-1)b &= a+p \\ b &= \frac{a+p}{ka-1} \end{aligned}$$

设$d=ka-1$，问题转化为需要满足以下条件：

- $p \nmid a$
- $d \mid a+p$
- $a \mid d+1$
- $a<b$

可以讨论以下两种情况完成：

- $d \le \sqrt{p+\sqrt{p+1}+1}$，此时根据2、3两条约束可以把需要验证的$a$的数量缩减到2个以内，即$a+p \in [p+1,p+d+1]$。
- $b \le \sqrt{p+\sqrt{p+1}+1}$且$b \le d$，此时根据2、4两条约束可以把需要验证的$a$的数量缩减到2个以内，即$a+p \in [p+1,p+b+1]$。

这一部分的计算可以在$O(\sqrt{p})$的时间内解决。

当$p \nmid a$或$p \nmid b$时，假设$p \nmid a$，$p \mid b$，设$b=pb'$，那么

$$ab \mid p(a+b+p) \ \Rightarrow\ pab' \mid p(a+pb'+p) \ \Rightarrow\ ab' \mid a+pb'+p$$

可以用和之前相同的方法得到$b'=\frac{a+p}{ka-p}$，并设$d=ka-p$，$k=\frac{d+p}{a}$。

由于$b',d,k$都要是整数，所以$d \mid a+p$，$a \mid d+p$，而$a,d \nmid p$，所以等同于求$ad \mid p(a+d+p)$，和前一种情况一样。

时间复杂度$O(\sqrt{p})$，空间复杂度$O(1)$。

## 1.79 Hypertrees

### 【试题来源】

Codechef DEC 11 HYPER

### 【试题大意】

3-超图类似于一个普通的图，但其中的边都连接三个点。

3-超树是一张去掉任意一条边以后都不连通的3-超图。

求有多少种$N$个点带标号3-超树，保证答案不超过$2^{63}-1$。

数据范围：$3 \le N \le 17$

### 【算法介绍】

一棵3-超树可以分解为若干点双连通的3-超树，这些点双连通3-超树共用的点即为这棵3-超树的割点。

考虑一棵$N>3$的点双连通3-超树上的每条边，一定连结了恰好一个只被这条边连结的点，可以用反证法证明。

去掉这些度数为1的点后，剩下的部分是一张一般的点双连通图，可以用搜索进行计数。

在计算出点双连通3-超树后，可以继续利用搜索将这些点双连通3-超树拼成树。强制以1号点为根，每次把子树接在已知点上。

由于$N \le 17$，所以可以打表。

# 1.80 Luckdays

## 【试题来源】

Codechef NOV 11 LUCKYDAY

## 【试题大意】

定义一个在模质数$P$域下的递推数列$S$：

$$\begin{aligned} S_1 &= A \\ S_2 &= B \\ S_i &= XS_{i-1} + YS_{i-2} + Z \end{aligned}$$

给出一个数$C$，以及$Q$组询问$[L_i, R_i]$，求有多少$k \in [L_i, R_i]$满足$S_k = C$。

数据范围：$2 \le P \le 10007$，$Q \le 20000$，$0 \le A, B, X, Y, Z, C < P$，$1 \le L_i \le R_i \le 10^{18}$

## 【算法介绍】

这个递推数列在模$P$域下的循环节一定在$O(P^2)$范围内。

容易求出递推矩阵$M$，以及逆矩阵$M^{-1}$

$$M = \begin{pmatrix} X & Y & Z \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad M^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{Y} & -\frac{X}{Y} & -\frac{Z}{Y} \\ 0 & 0 & 1 \end{pmatrix}$$

可以类比整数离散对数问题的BSGS算法，得到矩阵形式的BSGS算法，设步长为$S$。

通过设初值为$(B, A, 1)$并乘上递推矩阵$M^{k_1 S}(k_1 \le \frac{P^2}{S})$，将它们加入hash表内，再枚举$k_2(< S)$，将$(B, A, 1)$乘上逆矩阵$M^{-k_2}$的结果去hash表内查询，可以在$O(\frac{P^2}{S} + S)$的时间内得到这个递推数列的循环节。

求得循环节后，需要求出在第一个循环节内所有满足$S_k = C$的位置，设最终状态为$(C, D, 1)$，那么可以通过枚举$D$用与之前类似的方法计算，可以在$O(PS)$的时间内完成。

综上所述，$S$取$\sqrt{P}$时最优。预处理完后每组询问可以在$O(\log P)$的时间内完成。

时间复杂度$O(P\sqrt{P} + Q\log P)$，空间复杂度$O(P\sqrt{P})$。

## 1.81 Colored Domino Tilings and Cuts

### 【试题来源】

Codechef NOV 11 DOMNOCUT

### 【试题大意】

对于一张$N \times M$的方格图，需要对每个格子染色，使得每个格子四连通相邻的格子中都有恰好一个格子与其颜色相同。

定义一个方格图的割为，选择一行或者一列割开后，得到的两张图的染色仍然合法。

求在割数最小的前提下的最小染色数，并输出一种染色方案。

数据范围：$1 \leq N, M \leq 500$

### 【算法介绍】

假设$N \leq M$。

当$N, M$都是奇数的时候，染色是显然不可能被完成的。

对于$N = 5, M = 6$以及$N = 6, M = 8$都可以构造出割数为0染色数为3的方案，并且可以在不改变割数和染色数的条件下将$N$或$M$各自增大2，所有对于$N \geq 5$的情况都是可以构造出的。

对于$N \leq 4$的情况，可以根据每种情况再单独构造。

时间复杂度$O(NM)$，空间复杂度$O(NM)$。

## 1.82 The Baking Business

### 【试题来源】

Codechef OCT 11 BAKE

### 【试题大意】

有$S$个操作，操作有以下两种，为增加产品和询问符合条件的产品数量：

- I product_id[.size_id] province[.city_id[.region_id]] M/F age units_sold
- Q product_id[.size_id] province[.city_id[.region_id]] M/F start_age[-end_age]

其中[ ]内的信息为可省略的，省略了则表示这一项信息无关紧要。

有10种产品，每种都有3种不同的大小。

有10个省份，每个省份可以被划分为20个城市，每个城市又可以被划分成5个地区。

年龄范围为1∼90，每次操作的产品数量不超过100。

数据范围：$S \le 100000$

### 【算法介绍】

开一个7维数组模拟。

时间复杂度$O(S)$，空间复杂度$O(10 * 3 * 10 * 20 * 5 * 2 * 90)$。

# 1.83 Sine Partition Function

## 【试题来源】

Codechef OCT 11 PARSIN

## 【试题大意】

给出三个数$M, N, X$，求

$$\sum_{k_1+k_2+...+k_M=N} \sin(k_1X)\sin(k_2X)...\sin(k_MX)$$

其中$k_i$为非负整数，保证答案不超过$10^{300}$，要求相对或绝对误差小于$10^{-1}$。

数据范围：$1 \le M \le 30$，$1 \le N \le 10^9$，$0 \le X \le 2\pi$

## 【算法介绍】

设

$$\begin{aligned} F_{N,M} &= \sum_{k_1+k_2+...+k_M=N} \sin(k_1X)\sin(k_2X)...\sin(k_MX) \\ G_{N,M} &= \sum_{k_1+k_2+...+k_M=N} \cos(k_1X)\sin(k_2X)...\sin(k_MX) \end{aligned}$$

由三角函数公式易得

$$\begin{aligned} F_{N,M} &= \sin(X)F_{N-1,M-1} + \cos(X)F_{N-1,M} + \sin(X)G_{N-1,M} \\ G_{N,M} &= \cos(X)F_{N-1,M-1} + \cos(X)G_{N-1,M} - \sin(X)F_{N-1,M} \end{aligned}$$

可以用矩阵乘法倍增计算这个递推式。

时间复杂度$O(M^3 \log N)$，空间复杂度$O(M^2)$。

# 1.84 Short

## 【试题来源】

Codechef SEPT 11 SHORT

## 【试题大意】

给出两个数$n,k$，求有多少数对$(a,b)$满足$n<a,b<k$且$(a-n)(b-n)|ab-n$。

数据范围：$0\le n\le 10^5$，$k\le 10^{18}$

## 【算法介绍】

设$c=a-n$，$d=b-n$，$k(a-n)(b-n)=ab-n$，那么

$$\begin{aligned} kcd &= (c+n)(d+n)-n \\ (kc-c-n)d &= nc+n^2-n \\ d &= \frac{(c+n-1)n}{(k-1)c-n} \end{aligned}$$

考虑强制$c\le d$，那么

$$\begin{aligned} c &\le \frac{(c+n-1)n}{(k-1)c-n} \\ (k-1)c^2-nc &\le cn+n^2-n \\ (k-1)c^2-2nc-n^2+n &\le 0 \end{aligned}$$

由这个二次函数易得

$$c \le \frac{2n+\sqrt{4n^2+4(k-1)(n^2-n)}}{2(k-1)}$$

当$k=2$时

$$c \leq n+\sqrt{2n^2-n}$$

所以可以考虑枚举$c$，再枚举$(c+n-1)n$的约数作为$d$进行判断。

当$c$比较大的时候，可以通过转为枚举$k$来提高效率。

设$c$为阈值平衡后每次的枚举量，时间复杂度$O(cN)$，空间复杂度$O(N\log N)$。

## 1.85 Counting Hexagons

### 【试题来源】

Codechef SEPT 11 CNTHEX

### 【试题大意】

有$N$种木棍，木棍的长度为$1 \sim N$，每种长度的木棍都有$K$根。

要求选出一些木棍拼成一个面积为正的六边形，并要求最长的一根木棍的长度至少为$L$，其余木棍长度不超过$X$。

答案对$10^9 + 7$取模。

数据范围：$1 \le K \le 5$，$1 \le X < L \le N \le 10^9$，$N - L \le 100$

### 【算法介绍】

六边形面积为正也就是要求较短的五条边长度之和必须大于最长边。

考虑使用数位DP解决，由于有五个数需要处理，可以强制这五个数按照递增顺序排列，可以用$O(2^5)$的状态记录这几个数的相等关系以及最大的数是否和上界$X$相等。

从高位往低位处理，需要考虑从低位到高位的进位，由于只有五个数，所以进位的取值范围为$[0, 4]$。

在每一位的DP中，依次确定每个数在当前位的数值，并同时维护递增顺序和相等关系。

时间复杂度$O((N - L) \lg N)$，空间复杂度$O(\lg N)$。

## 1.86 Shortest Circuit Evaluation

### 【试题来源】

Codechef AUG 11 SHORTCIR

### 【试题大意】

给出一个布尔表达式，由括号、and、or、not以及一些布尔变量组成，保证每个变量在表达式中只出现一次。

在计算一系列只包含and运算的表达式时，从左往右运算，一旦当前值为false就会进行剪枝，or运算也同理。

给出每个变量为true的概率$p_i$，要求将这个布尔表达式中变量的顺序重排，在不改变运算结果的前提下最小化期望运算次数。

数据范围：表达式长度$\leq 30000$，变量数量$\leq 1000$，变量长度$\leq 5$，$0 < p_i < 1$

### 【算法介绍】

括号的情况可以递归处理，只需要考虑计算一系列变量连续and或连续or运算的情况。

设变量$i$为true的概率为$p_{1i}$，期望运算次数为$f_{1i}$，为false的概率为$p_{0i}$，期望运算次数为$f_{0i}$。

当运算负号为and时，可以贪心将这些变量按照$\frac{f_{0i}+f_{1i}}{p_{0i}}$排成升序，运算符号为or时，按照$\frac{f_{0i}+f_{1i}}{p_{1i}}$排成升序。

设表达式长度为$S$，变量数量为$n$，时间复杂度$O(|S| + n\log n)$，空间复杂度$O(|S| + n)$。

## 1.87 Something About Divisors

### 【试题来源】

Codechef AUG 11 DIVISORS

### 【试题大意】

对于给定的正整数$B, X$，求有多少正整数$N$满足存在至少一个正整数$D(N < D \leq B)$使得$D|NX$。

数据范围：$B \leq 10^{12}$，$X \leq 60$

### 【算法介绍】

设$K = \frac{NX}{D}$，容易发现$K < X$，所以可以枚举$K$计算所对应的$N$，而同一个$N$可能被多个$K$统计，所以需要计算不存$J(K < J < X)$使得$J|NX$的$N$。

先不考虑$J$的影响，那么

$$K|NX \Rightarrow \frac{K}{gcd(K, X)}|N$$

设$C_k = \frac{K}{gcd(K,X)}$，$N = C_k M$，那么

$$NX = DK \Rightarrow N \leq \frac{BK}{X} \Rightarrow M \leq \frac{BK}{XC_K}$$

所以可以得到在不考虑重复影响下，$N$的数量为$\left\lfloor \frac{BK}{XC_K} \right\rfloor$。

考虑$K|NX$且$J|NX$的情况，可以得到$J|C_K MX$，即$\frac{J}{gcd(J,C_K X)}|M$。

设$D_J = \frac{J}{gcd(J,C_K X)}$，那么可以用容斥来计算不出现重复计算的$N$的数量

$$\sum_{S \subseteq \{K+1,K+2,\ldots,X-1\}} (-1)^{|S|} \left\lfloor \frac{\left\lfloor \frac{BK}{XC_K} \right\rfloor}{lcm(D_{S_1}, D_{S_2}, \ldots, D_{S_{|S|}})} \right\rfloor$$

由于$lcm$的合法取值数量不会太多，所以可以采取类似DP的方法进行转移计算，每次将相等的状态进行合并。

可以用以下两种方法来优化：

- 转移时会有大量状态合并后的权值和为0，可以随时舍弃。
- 由于需要容斥去除的是所有$D_J$倍数的$M$，所以当存在$D_J|D_L$时，$D_L$可以不进入容斥枚举。

设状态的数量为$S$，时间复杂度$O(X^2S\log S)$，空间复杂度$O(S)$。

# 1.88 Billboards

## 【试题来源】

Codechef JULY 11 BB

## 【试题大意】

在公路上有$N$个广告牌，要求任意连续的$M$个广告牌中至少有$K$个餐馆广告，并希望在满足条件的同时使用最少的广告牌。

现在大厨想知道有多少种使用广告牌的方案，答案对$10^9+7$ 取模。

数据范围：$1 \le K \le M \le 50$，$M \le N \le 10^9$

## 【算法介绍】

先考虑$M|N$的情况，把串分为$\frac{N}{M}$段，每段长度为$M$，根据限制每段中都必须有至少$K$个元素。这时将每段中的$K$个元素靠左放置就能满足要求，所以最小个数为$\frac{NK}{M}$。

定义$w_{i,j}$为第$i$段中第$j$个元素的位置，容易发现$w_{i,j} < w_{i,j+1}$且$w_{i,j} \ge w_{i+1,j}$，可以将其看作一个矩阵，这种矩阵被称为半标准杨氏矩阵，对于一个$n$行$m$列的矩阵方案数有计算公式

$$\prod_{i=1}^{n}\prod_{j=1}^{m}\frac{r+i-j}{n+m-i-j}$$

其中$r$为可填数的集合大小，将分子分母写出后会发现只需要计算前$r-m$行的分母以及后$r-m$行的分子即可，其余部分一定会消完。

对于$M \nmid N$的情况，如果$N \ mod \ M \le M-K$，那么每一段的前$N \ mod \ M$个位置都是0，否则后$M - N \ modM$个位置都是1，可以转化为整除的情况。

时间复杂度$O(KM\log p)$，空间复杂度$O(1)$。

# 1.89 Trial of Doom

## 【试题来源】

Codechef JULY 11 YALOP

## 【试题大意】

给出一个$N \times M$的方格图，有$K$个红色格子，其余格子都是蓝色。

每次可以移动到一个八连通的相邻格子，每当离开一个格子时，那个格子及与其四连通相邻的格子都会变色。

求是否存在一种移动方式使得能从$(1,1)$移动到$(N,M)$，并使最终所有格子的颜色都是蓝色。

数据范围：$1 \le N, M \le 10^9$，$min(N,M) \le 40$，$0 \le K \le min(NM, 10000)$

## 【算法介绍】

假设$N \le M$，当$N \ge 2$时，可以构造出一种不改变任何格子的状态来移动到相邻位置的方法：

$$\begin{matrix} \mathbf{0} & 0 \\ 0 & 0 \end{matrix} \Rightarrow \begin{matrix} 1 & \mathbf{0} \\ 0 & 0 \end{matrix} \Rightarrow \begin{matrix} \mathbf{1} & 1 \\ 0 & 0 \end{matrix} \Rightarrow \begin{matrix} 0 & \mathbf{1} \\ 0 & 0 \end{matrix} \Rightarrow \begin{matrix} 0 & 0 \\ \mathbf{0} & 0 \end{matrix}$$

依靠这个方法，也可以构造出只改变当前格子颜色的方法：

$$\begin{matrix} \mathbf{0} & 0 \\ 0 & 0 \end{matrix} \Rightarrow \begin{matrix} 1 & \mathbf{0} \\ 0 & 0 \end{matrix} \Rightarrow \begin{matrix} \mathbf{1} & 0 \\ 0 & 0 \end{matrix}$$

所以，对于$N \ge 2$的情况，问题可以转化为判断一个异或方程组是否有解，但是这个异或方程组有$O(NM)$个变量。

假设$(X, Y)$这个格子需要改变颜色，那么可以通过触发$(X, Y+1)$来完成，不停进行这样类似的操作，可以把所有红色的格子都移动至第$M$列。由于$N$较小，所以会影响到的格子集合将会出现与转移列数相关的循环节，可以预处理完成。

可以用类似的方法用第1列的$N$个格子来代表所有状态集合，剩下的部分可以直接用高斯消元判断。

考虑$N=1$的情况，移动方式受到了拘束，只能将任意红色格子左右移动3个格子的距离，于是可以将所有红色格子移动到最后3列，倒数第3 列的红色格子也是可以直接抹除的。

状态还受到$M$奇偶性的影响，当$M$是偶数时，最后两列会产生由移动造成的红色格子。

剩下还剩下第1个格子的状态可以随意改变，综合以上几种情况即可完成判断。

设循环节长度为$L$，时间复杂度$O(K+NL)$，空间复杂度$O(K+NL)$。

## 1.90 Attack of the Clones

### 【试题来源】

Codechef JUNE 11 CLONES

### 【试题大意】

定义一个由$A \to \{0,1\}$的映射为布尔函数，其中$A$包含$n$个01变量。

称满足一些条件的布尔函数的集合为$clone$，现在定义四种特殊的$clone$：

- Z：满足$f(0,0,\ldots,0)=0$。
- P：满足$f(1,1,\ldots,1)=1$。
- D：满足$\neg f(x_1,x_2,\ldots,x_n)=f(\neg x_1,\neg x_2,\ldots,\neg x_n)$。
- A：满足如果$f(a_1,\ldots,c,\ldots,a_n)=f(a_1,\ldots,d,\ldots,a_n)$那么$f(b_1,\ldots,c,\ldots,b_n)=f(b_1,\ldots,d,\ldots,b_n)$，其中$c,d$位于同一个变量位置。

现在给出一个表达式$S$，由$Z$、$P$、$D$、$A$、括号以及取并集、取交集、取反集、取差集四种运算构成，求这个表达式所代表的集合由多少个函数组成，答案对1000003取模。

数据范围：$n,|S|\le 100$

### 【算法介绍】

对于每个布尔函数，可以用一个4位二进制数来表示它是否属于集合ZPDA，那么就可以把布尔函数分为16种，对于每一种布尔函数可以预先计算它所包含的函数数量。那么，对于这个表达式的运算就可以转化为一个16位二进制数的运算，可以暴力递归处理括号来完成。

时间复杂度$O(|S|)$，空间复杂度$O(|S|)$。

# 1.91 Minesweeper Reversed

## 【试题来源】

Codechef JUNE 11 MINESREV

## 【试题大意】

在一张$R \times C$的网格矩阵中进行扫雷游戏，有一些方块是雷，没有雷的格子上显示八连通相邻格子中的雷的数量。

一次操作可以点开一个格子，当这个格子的八连通相邻格子中没有雷时，它的八连通相邻格子都会被打开，这个影响还会被递归进行下去，也就是说将会打开一块没有雷的区域。

现在所有格子都被打开了（包括有雷的位置），需要通过一些操作把格子都关上。

一次操作可以关上一个格子，以及那些能同时打开它的格子，求最小操作次数。

数据范围：$R, C \leq 50$

## 【算法介绍】

可以把能够一次性打开的块都求出来，这些块之间会有交，关闭那些交上的格子可以同时关闭那两个块。

可以对这些块的相互建一张图，用带花树求这张图的最大匹配。图的点数和边数均为$O(RC)$。

最后再加上那些只能通过一次操作单独关闭的格子数即可。

时间复杂度$O(R^2C^2)$，空间复杂度$O(RC)$。

# Chapter 2

# challenge型试题

# 2.1 Division of Lands

## 【试题来源】

Codechef APRIL 15 DIVLAND

## 【试题大意】

给出一张$N$个点$M$条边的无向图，$N$是偶数，要求把点均分为两份使得连结两个点集之间的边的边权和最小。

数据范围：$1 \le N, M \le 10^6$

## 【算法介绍】

可以选用类似Prim最小生成树的算法，随机选一个起点，将它的所有出边的边权加到对应的点上，每次选择一个权值最大的点和当前连通块合并。

这种算法在树上的效果不太好，所以对于树的情况采用另外一个算法。随机选一个点当根，每次在能够添加的子树中选择点数最多的，其次选择父亲边权值最小的。

在用贪心算法得到一个初始解后，可以再用随机调整算法来尝试更新答案。

## 2.2 Jigsaw Puzzle Solving

### 【试题来源】

Codechef FEB 15 CHPUZZLE

### 【试题大意】

给出一张$W \times H$的方格图以及$K$块拼图，每块拼图都是一个连通块。

要求构造一种方案将拼图尽量多地放到方格图上，并且拼图之间不重叠。

数据范围：$10 \le W, H \le 1000$，$K \le \frac{WH}{2}$

### 【算法介绍】

为了加速对可用位置的查找，考虑用一个矩形来粗略代表每个拼图，但是这样会导致实际可用的位置变少，所以先将拼图按照密度排序再逐一验证。

设第$i$块拼图的粗判矩形的边长为$X_i, Y_i$。

维护一个待验证坐标队列，一开始将$(1, 1)$入队。当$(x, y)$被拼图$i$占用时，就将$(x + X_i, y)$和$(x, y + Y_i)$入队。

每次都将整个队列都验证一遍的话运算量会过大，可以设定一个阈值$S$，当一个坐标被$S$次验证后仍没有拼图可以覆盖它，那么就将它从队列中移除。

## 2.3 Kali and Devtas

### 【试题来源】

Codechef DEC 14 KALKI

### 【试题大意】

给定二维欧氏平面内的$N$个点，求一棵生成树使得$C_i$的最大值最小。

其中$C_i$的定义为：对于每个点，设在生成树中与其相邻的点中最远的点的距离为$R$，那么以该点为圆心，半径$R$以内的的点的$C_i$全部增加1（包括自身）。

数据范围：$3 \le N \le 400$

### 【算法介绍】

设$p_{i,j}$为距离点$i$第$j$远的点，$rank_{i,j}$为点$j$距离点$i$第$rank_{i,j}$远。

构造完全图，其中点$(u,v)$中的边权为$d_{u,v} = rank_{u,v} + rank_{v,u}$，对这张图求一棵最小生成树就能得到不错的解。

考虑在此基础上对每个点加权以得到更优的解，设$C_i$为上述方案所得到的$C$值，构造新图

$$d_{u,v} = \sum_{i=1}^{rank_{u,v}} C_{p_{u,i}} + \sum_{i=1}^{rank_{v,u}} C_{p_{v,i}}$$

这个过程可以不断重复来得到更优的解。

## 2.4 Deleting numbers

### 【试题来源】

Codechef AUG 13 DELNMS

### 【试题大意】

给出一个长度为$N$的数组$A$，每次可以选择一个数对$(v,t)$满足$A[v]=A[v+t]=...=A[v+kt]$，其中$k$是满足$v+kt\le N$的最大的数，然后把$A[v],A[v+t],\ldots,A[v+kt]$删去。

要求用尽量少的操作将整个数组都删完。

数据范围：$1\le N,A_i\le 10^5$

### 【算法介绍】

从$N$到1依次输出$(i,1)$就是一组合法的解。

在此基础上可以选出最大的那个数，将其它数都删去后输出$(1,1)$。

除此之外可以在数组的末端几位暴力枚举$v$和$t$，在现有数组上进行验证。

或者求出最大值最后一次出现的位置，在之后的位置继续进行原算法。

## 2.5 To challenge or not

### 【试题来源】

Codechef JUNE 13 CHAORNOT

### 【试题大意】

给出一个长度为$M$的随机集合$B_i$，要求找出一个尽量大的子集使得子集中的数不能构成等差数列。

数据范围：$M \le 100000$，$0 \le B_i < 100000$

### 【算法介绍】

由于集合是随机生成的，所以答案不可能很大。

考虑将$B$中的数排序，从小到大贪心将每个数加入答案集合，根据等差数列$2B_{mid} = B_l + B_r$的式子，每次可以$O(1)$检查这个数是否可以选，$O(|Ans|)$计算出新增的不能选的数。

对于每个可以加入答案集合的数，可以以$\frac{1}{p}$的概率选择不取它，经过多次随机取最优值。

## 2.6 Maximum Sub-rectangle in Matrix

### 【试题来源】

Codechef OCT 12 MAXRECT

### 【试题大意】

给出一个$H \times W$的整数矩阵$A$，求一个子矩阵使其中元素之和尽可能大。

这个子矩阵不要求是连续的，即求出一些行和一些列，选取这些行列相交处的元素，输出这些行列。

数据范围：$200 \le H, W \le 300$，$|A_{i,j}| \le 10^9$

### 【算法介绍】

这个问题并没有合适的特殊做法，用类似爬山的通用算法就可以得到较优的解。

先随意生成一个权值为正的初始状态，比如可以选取整个矩阵中最大的那一个格子。

考虑调整这个局面，计算出将每行每列的选取状态改变后，子矩阵和的增值，然后选取出增值最大的那一行或一列进行修改。

在修改一行或一列时，可以直接计算这一行修改后对其他行列增值的贡献，这样可以做到修改和选取的时间复杂度都为$O(H)$或$O(W)$。

换成其他爬山类做法也能得到很优的解，选取较优的初始状态也可以使得答案更优。

## 2.7 Simultaneous Nim

### 【试题来源】

Codechef SEPT 12 SIMNIM

### 【试题大意】

有$N$堆石子，第$i$堆石子的石子个数为$A_i$。

在这些石子堆上进行Nim游戏后手有必胜策略，要求把这$N$堆石子划分为$K$个集合，使得在每个石子堆中进行Nim游戏后手都有必胜策略。

要求使得$K$尽可能大。

数据范围：$10 \le N \le 1000$，$1 \le A_i \le 2^{60} - 1$

### 【算法介绍】

问题相当于给出$N$个数$A_i$，将其划分为$K$个集合，并使得每个集合中$A_i$的异或和均为0。

考虑使用线性基维护，随机一个排列$P$，按照这个排列的顺序将$A_{P_i}$依次加入线性基中，假如新加入的$A_i$解出了一个异或和为0的集合，就将这个集合从线性基中删去。

将一个数加入线性基的复杂度为$O(\log A)$，提出解时需要将整个线性基重构，复杂度为$O(\log^2 A)$，重构的次数为$K$次，所以重构部分的复杂度不超过$O(N \log^2 A)$。

## 2.8 Closest Points

### 【试题来源】

Codechef JUNE 12 CLOSEST

### 【试题大意】

给出三维空间中的$N$个点$(x_i, y_i, z_i)$，以及$Q$组询问，求离每个询问点的最近点。

不需要保证每次询问都返回正确的解。

数据范围：$1 \le N, Q \le 50000$，$|x_i|, |y_i|, |z_i| \le 10^9$

### 【算法介绍】

最近点问题是经典问题，可以直接用kdtree解决。

在kdtree的建树中，需要每次以方差最大的那一维作为划分依据。

每次询问的期望复杂度为$O(\log N)$。

时间复杂度$O((N+Q)\log N)$，空间复杂度$O(N)$。

# 2.9 Killing Gs

## 【试题来源】

Codechef MAY 12 CKROACH

## 【试题大意】

有$N$只害虫，$M$种杀虫剂，第$i$种杀虫剂的价格为$C_i$，第$i$只害虫被第$j$种杀虫剂杀死的概率为$P_{i,j}\%$。

每种杀虫剂最多使用一次，要求用尽量少的钱购买杀虫剂使得每只害虫被杀死的概率不低于90%。

数据范围：$50 \le N, M \le 200$，$0 \le P_{i,j} \le 90$

## 【算法介绍】

对于每种杀虫剂，设定估价为使用这种杀虫剂后，所有害虫的生存几率下降的总和除以$C_i$，即每单位钱减少的生存概率。

基于这个估价每次贪心选取最优的杀虫剂使用，就可以得到一个不错的初始解。

在这个解的基础上，每次随机保留一部分杀虫剂的使用，剩下部分继续贪心，重复迭代多次。

将概率取log后计算估价效果更优。

## 2.10 Similar Graphs

### 【试题来源】

Codechef APRIL 12 SIMGRAPH

### 【试题大意】

给出两张$N$个点的无向图，要求把这两张图的点重新标号，使得这两张图的公共边尽可能多。

数据范围：$30 \le N \le 75$

### 【算法介绍】

可以使用随机爬山、模拟退火等通用算法。

初始化一个排列，然后依次枚举$u < v$，交换这两个点的标号，可以$O(N)$计算对答案的影响。

对于每组数据，将这个调整循环操作若干次即可。

# 2.11 The Great Plain

## 【试题来源】

Codechef OCT 11 LAND

## 【试题大意】

给出一个$N \times M$的方阵$A$，这个方阵中有一些位置上已经填有数$A_{i,j} \in [1, 50]$。

设两个相邻格子之差为$k$，那么这两个格子将会造成$2^k$的代价，求一种填满所有格子的方案使得代价和尽量小。

数据范围：$10 \leq N, M \leq 100$

## 【算法介绍】

可以先贪心初始化一个解：将每个空格子的权值赋值为离它最近的非空格子的权值。

在此基础上，每次对所有可修改格子进行枚举，将其权值修改为局部最优。

这样经过若干次迭代后，整个方阵的权值将会基本稳定在一个局面，这时可以随机将方阵中的一些数的值浮动一个单位，之后继续进行原先的迭代。这些随机浮动带来的影响可能会使答案变优。