A. Arithmetic Subsequence

A.1. **题目描述.** 给一个长度为N的整数序列$A = (A_i)$，问能否将该序列重排得到序列$B = (B_i)$，满足

- $1 \le i < j < k \le N$;
- (B_i, B_j, B_k)不构成等差序列。

A.2. **数据范围.** 25组数据,$N \le 5000, 0 \le A_i \le 10^9$.

A.3. **做法.** 首先如果某个数出现次数大于等于3则不存在解。然后如果所有数字均为偶数，我们可将所有数除以二;如果所有数字均为奇数，我们可将所有数减去一;否则,我们将所有奇数放在左边，所有偶数放在右边，对奇数/偶数分治解决。单组数据时间复杂度$O(n \log \max A_i)$。

B. Conquest of Masters Tour

B.1. **题目描述.** 两个人各带n套卡组比赛，规则如下：初始时所有卡组都视为“未通过”。每轮两人分别选择一个“未通过”的卡组进行战斗，胜者的卡组视为“通过”，如果一位玩家的所有卡组都“通过”则获胜。给出每对卡组之间的胜率，求Player 1的获胜概率。

B.2. **数据范围.** 5组数据,$n \le 8$.

B.3. **做法.** 用$dp(S_1, S_2)$表示第i位玩家“未通过”的卡组为$S_i (i \in \{1, 2\})$时，Player 1获胜的胜率。计算$dp(S_1, S_2)$时，考虑Player 1的$|S_1|$种卡组选择以及Player 2的$|S_2|$种卡组选择，可以得到一个$|S_1| \times |S_2|$的矩阵P，$P_{i,j}$表示Player 1选择第i种策略，Player 2选择第j种策略时，Player 1的获胜概率。该问题为零和博弈，存在纳什均衡点，可以用单纯形求解，可以参考：

https://www.cs.cmu.edu/∼ sandholm/cs15-892F13/algorithmic-game-theory.pdf (Section 1.4.2).

C. Fast Bubble Sort

C.1. **题目描述.** 给定任何一个长度为N的数组$A = (a_1, a_2, \dots, a_n)$,令$B(A)$表示对$A$进行一次bubble sort循环之后得到的数组。令$num(A)$表示从A到$B(A)$最少需要移动元素(数组区间循环移位)的次数。给定一个$1 - n$的排列P以及q组$1 \le l \le r \le n$，求$num(P[l, r])$

C.2. **数据范围.** 10组数据,$1 \le n, q \le 10^5$。

C.3. **做法.** 假设$P = n_1\lambda_1 n_2\lambda_2 \cdots n_k\lambda_k$,则$B(P) = \lambda_1 n_1 \lambda_2 n_2 \cdots \lambda_k n_k$,其中$n_1, \cdots, n_k$为从左到右的局部最大值且有

$$n_1 < n_2 < \cdots < n_k,$$

则不难证明答案为非空λ_i的个数。

将询问离线,每次从n到1倒序扫描左端点l并回答所有左端点为l的询问。对于每个固定的左端点l，$[l, n]$中从左到右的局部最大值可以通过单调栈维护，局部最大值插入/删除对于答案的影响可以用树状数组/线段树快速更新/求解。单组数据时间复杂度为$O((n + q) \log n)$

D. If you can't beat them, join them!

D.1. **题目描述.** 定义一个graph game (G, v)为:

- 给定一个有向图$G = (V, E)$和一个顶点$v \in V$。初始时有一个token在v上，两个人交替将token沿着一条有向边移动，无法移动者输。

对于$k (k \ge 0)$个graph game $(G_1, v_1), \dots, (G_k, v_k)$,定义这个$k$个graph game的join构成的game为

- 双方轮流交替在每一个game中同时将token沿着一条有向边移动，在任何一个game中无法移动token者输(Lose one,Lose all!)。

给定k个graph game $(G_1, v_1), \dots, (G_k, v_k)$, Alice想要选择这些graph game的一个非空集合,然后和Bob玩在选出的集合join后的游戏。Alice先手，问Alice有多少种选择方法使得在双方最优策略下必胜，答案对998244353取模。

D.2. **数据范围.** 5组数据, $1 \le k \le 10^6$, 单组满足$\sum |V_i| \le 2 \times 10^6, \sum |E_i| \le 2 \times 10^6$。

D.3. 做法. 不难发现, 对于$(G_1, v_1), \ldots, (G_k, v_k)$的join, 双方最优策略是:
对于第i个graph game,

- 如果自己当前处于必败态，那么要最大化自己失败的步数
- 如果自己当前处于必胜态，那么要最小化对方失败的步数
- 否则该游戏为平局

因此对于每个graph game (G_i, v_i)，可以计算出一个值$f(G_i, v_i)$含义如下:

- 如果先手必胜，$f(G_i, v_i) > 0$, 且$f(G_i, v_i)$表示双方最优策略下该游戏持续步数
- 如果先手必败，$f(G_i, v_i) < 0$, 且$-f(G_i, v_i)$表示双方最优策略下该游戏持续步数
- 否则该游戏为平局，且$f(G_i, v_i) = 0$

$f(G_i, v_i)$可由以下方法在$O(|V_i| + |E_i|)$时间内计算($f(G_i, v)$将被简写为$f(v)$):

- 初始时将对于所有$v \in G_i$，将$f(v)$置为0，对于G_i中没有出度的点v,$f(v)$置为-1。
- 持续更新可以按照以下方法更新的顶点u(每个点只会被更新至多一次，可用队列实现):
  - 如果存在$(u, v) \in E_i$且$f(v) < 0$, 那么将$f(u)$置为$\max\limits_{(u,v) \in E_i} (-f(v) + 1)$
  - 如果对于所有$(u, v) \in E_i$都有$f(v) > 0$, 那么将$f(u)$置为$\min\limits_{(u,v) \in E_i} (-f(v) - 1)$

令S为Alice选择的graph game的集合,令$S_{\text{Win}} = \{(G, v) \in S \wedge f(G, v) > 0\}$,$S_{\text{Lose}} = \{(G, v) \in S \wedge f(G, v) < 0\}$ 可以发现Alice获胜当且仅当$\min_{(G,v) \in S_{\text{Win}}} f(G, v) < \min_{(G,v) \in S_{\text{Lose}}} -f(G, v)$,对$S_{\text{Win}}$和$S_{\text{Lose}}$中的game按照$f$值排序后即可线性求出。单组数据单组数据总时间复杂度为$O(\sum |V_i| \log (\sum |V_i|) + \sum |E_i|)$, 注意到$f$值的绝对值至多为$|V_i|$，时间复杂度可以降到$O(\sum |V_i| + \sum |E_i|)$.

E. Leapfrogger

E.1. 题目描述. n个随从中有1个带有“跳蛙”亡语，亡语会重复k次，现在按照随机顺序杀死n个怪，求“跳蛙”亡语触发次数的期望。你需要对$k = 2, 3, ..., m$分别求解。

E.2. 数据范围. $1 \le n < 998244352, 2 \le m \le 100000$.

E.3. 做法. 通过题意可得递推式$f_n = \left(1 + \frac{k-1}{n}\right) f_{n-1} + \frac{1}{n}$. 则$f_n + \frac{1}{k-1} = \frac{n+k-1}{n} \left(f_{n-1} + \frac{1}{k-1}\right)$. 通过归纳法得

$$f_n = \frac{k}{(k-1)n} \binom{n+k-1}{k} - \frac{1}{k-1}$$

单组数据时间复杂度:$O(m)$.

F. Mario Party

F.1. 题目描述. 给定序列$a_1, ..., a_n$，定义函数$f(x, l, r)$如下:
for $i = l$ **to** r **do: if** $x + a_i \ge 0$ **then** $x = x + a_i$. **output:** x.
q次询问(x, l, r)，输出$f(x, l, r)$的值。

F.2. 数据范围. 4组数据,$n, q \le 5 \times 10^5, \sum |a_i| \le 10^6$。

F.3. 做法. 关键点(i, x)：做完第i次操作后当前值为x。只需要考虑满足$a_i + x < 0$的关键点，维护倍增。单组数据时间复杂度:$O((n + \sum |a_i|) \log n)$.

G. Matryoshka Doll

G.1. 题目描述. 有n个套娃，大小为$a_1 \le a_2 \le ... \le a_n$，现在要将这些套娃分成$k$组，每组套娃按照大小排序后相邻两个套娃之间的大小差距要求$\ge r$，求方案数。

G.2. 数据范围. 20组数据,$1 \le k \le n \le 5000, 1 \le a_i, r \le 10^9$，输入保证所有数据中$\sum n \le 50000$.

G.3. 做法. $dp(x, y)$表示将前x个套娃分成y组的方案数，转移为

$$dp(x, y) = dp(x-1, y-1) + dp(x-1, y) \cdot \max\{0, y - f(x)\}.$$

其中$f(x)$表示满足$1 \le z < x$且$a_x - r < a_z \le a_x$的z的个数。单组数据时间复杂度:$O(n^2)$.

H. Shortest Path in GCD Graph

H.1. 题目描述. 给定n个点的完全图，两个点之间距离为它们的gcd，q次询问两个点之间的最短路以及方案数。

H.2. 数据范围. $n \le 10^7, q \le 50000$.

H.3. 做法. 首先最短路长度不超过2，因为对任意(x, y)，沿路径$x \to 1 \to y$即可得到一条长度为2的路径。最短路长度为1当且仅当$gcd(x, y) = 1$，否则等价于询问$[1, n]$中满足$gcd(x, z) = 1$且$gcd(y, z) = 1$的z的数量，分解x, y后用容斥可以解决。注意$gcd(x, y) = 2$时，直接$x \to y$也是一条长度为2的路径。单组数据时间复杂度: $O(n + Q \cdot 2^p)$，其中$p \le 12$为x, y的不同质因子个数。

I. Simple Math 4

I.1. 题目描述. 给定N, L, R, X，判定是否存在整数序列$A_1, A_2, \ldots, A_N$，满足

- $L \le A_i \le R$;
- $\bigoplus_{i=1}^{N} A_i = X$.

如果存在，最大化$\sum_{i=1}^{N} A_i$.

I.2. 数据范围. 3000组数据，$1 \le N \le 10^9, 0 \le X, L, R \le 10^9, L \le R$.

I.3. 做法. 我们首先证明，如果最优解存在，则最优解至多有$\lceil \log V \rceil$个数满足$A_i \ne R$。若不然，则一定存在两个数A_i, A_j满足$A_i, A_j \ne R$且A_i, A_j的二进制末尾连续1的个数相等。因此，将A_i, A_j同时加1仍然满足条件，矛盾。

令$f_{i,j,k}$为前i位中仍有j个数二进制下前i位与L的前i位一致，k个数二进制下前i位与R的前i位一致时贡献的最大值。利用上述状态可在$O(\log^4 V)$时间内通过动态规划解决问题。

J. Sum Plus Product

J.1. 题目描述. n个数$a_1, a_2, \ldots, a_n$。每次随机选两个数a, b合并成为$ab + a + b$，直到剩下一个数为止。求剩下数字的期望。答案对998244353取模。

J.2. 数据范围. 20组数据,$1 \le n \le 500, 0 \le a_i < 998244353$。

J.3. 做法. 可以观察到$\prod_{i=1}^{n}(a_i + 1)$是一个不变量，因此答案是$\prod_{i=1}^{n}(a_i + 1) - 1$。时间复杂度$O(n)$。

K. The Alchemist of the Discrete Mathematics

K.1. 题目描述. 给定质数p，整数c，希望知道有多少个$\mathbb{F}_p$的子集满足存在n次多项式$f(x) \in \mathbb{F}_p[x]$及$g(x) \in \mathbb{F}_p[x]$并且$g(x) | f(x)$，使得其为$h(x) = f(x)g(cx)$在$\mathbb{F}_p$的零点集合

K.2. 数据范围. 多组数据,$1 \le c < p \le 5 \times 10^5$, 输入保证所有数据中$\sum p \le 10^7$.

K.3. 做法. 首先，可以划分出$(p-1)/o(c)$个长为$o(c)$的环和0这单独一个点，这些结构之间互不影响。n次多项式只能有n个零点，我们对每个环最少需要的零点数进行统计，多余的次数可以通过选择任一个零点来补足，对于环上一段连续长为l的$h(x)$的零点，我们至少需要并且可以用$\lceil l/2 \rceil$个$f(x)$的零点来满足。

为了描述方便，将是$h(x)$的零点的位置定为1，否则定为0。对于环而言，我们定向顺时针后，同时将每段连续的1改为01间隔的串不影响答案，并且该过程可逆，对该过程进行分析，最终对长为n的环，需要s个的零点的方案数为

$$2^s\left(\binom{n-s}{s} + \binom{n-s-1}{s-1}\right) - [n == 2s] + [n + 1 == 2s]$$

最后，合并所有环时，只需要在p的范围内进行FFT变换，点值快速幂后逆变换即可，

不过，注意到上述方案并不能构造出没有零点的多项式，我们需要单独对这部分进行讨论，当$n = 1$时，我们无法进行构造，而当$n \ge 2$时，我们取$\mathbb{F}_p$上的n次不可约多项式即可。

单组数据时间复杂度：$O(p \log p)$.