# AtCoder Beginner Contest 450 题目讲解

云浅

2026-3-23

# Outline

# A. 3,2,1,GO

# A. 3,2,1,GO

略

# B. Split Ticketing

# B. Split Ticketing

略

# C. Puddles

# C. Puddles

略

# D. Minimize Range

# D. Minimize Range

给定 $N, K$ 和长为 $N$ 的序列 $A$。你可以执行以下操作任意多次：

- 选择一个 $1 \leq i \leq N$，将 $A_i$ 变为 $A_i + K$。

你希望最小化 $\max(A) - \min(A)$ 的值。输出这个最小值。

**数据范围：**

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq A_i, K \leq 10^9$

# D. Minimize Range

虽然题目中只允许 $+K$，但我们只关注极差，因此我们也可以允许 $A_i \leftarrow A_i - K$ 的操作：这等价于把所有 $j \neq i$ 的 $A_j$ 都 $+K$。

那么现在每个 $A_i$ 我们只关注它 $\bmod K$ 的余数 $r_i$，$A_i$ 可以取到所有的 $r_i + c \times K$。

我们枚举最小值是哪个 $r_i + c \times K$，那么最优方案一定是让其他的 $A_j$ 在满足 $A_j \geq r_i + c \times K$ 的前提下尽可能小，也就是如果 $r_j < r_i$ 那么需要取 $r_j + (c+1)K$，否则只需要取 $r_j + cK$。这时候发现也不需要枚举 $c$ 了，直接令 $c = 0$ 即可。

这样只需要排序一遍就可可以求了，总复杂度是 $O(N \log N)$。

# E. Fibonacci String

# E. Fibonacci String

给定字符串 $X$ 和 $Y$。定义字符串序列 $S_1, S_2, \ldots$ 如下：

- $S_1 = X$
- $S_2 = Y$
- 对于 $i \geq 3$，$S_i$ 是 $S_{i-1}$ 与 $S_{i-2}$ 按此顺序拼接而成的字符串。

对于每个 $i = 1, 2, \ldots, Q$，回答以下问题：

**问题：** 给定整数 $L_i, R_i$ 和字符 $C_i$。求在 $S_{10^{18}}$ 的第 $L_i$ 个字符到第 $R_i$ 个字符中，字符 $C_i$ 出现的次数。

**数据范围：**

- $X$ 和 $Y$ 是由小写英文字母组成的字符串，长度在 1 到 $10^4$ 之间（包含端点）
- $1 \leq Q \leq 10^5$
- $1 \leq L_i \leq R_i \leq 10^{18}$
- $C_i$ 是小写英文字母

# E. Fibonacci String

由于 $S_i$ 是 $S_{i+1}$ 的前缀，因此如果某个 $S_k$ 的长度 $\geq 10^{18}$ 了，那么 $S_{10^{18}}[L, R]$ 实际上就等于 $S_k[L, R]$。而我们知道 $S_i$ 的长度至少是 $\text{Fib}_i$，因此 $k$ 只有 $\log 10^{18}$ 左右。

那么我们考虑怎么计算 $S_k[1, R]$ 中 $c$ 的出现次数。设这个数为 $\text{query}_c(k, R)$

可以考虑递归下去，由于 $S_k = S_{k-1}S_{k-2}$，因此：

- 如果 $R \leq |S_{k-1}|$，那么 $S_k[1, R] = S_{k-1}[1, R]$，有 $\text{query}_c(k, R) = \text{query}_c(k-1, R)$
- 如果 $R > |S_{k-1}|$，那么 $S_k[1, R] = S_{k-1}S_{k-2}[1, R - |S_{k-1}|]$，有 $\text{query}_c(k, R) = \text{cnt}_{c,k} + \text{query}_c(k-2, R - |S_{k-1}|)$。

其中 $\text{cnt}_{c,k} = \text{query}_c(k, |S_k|)$ 表示 $S_k$ 中字符 $c$ 的出现次数，可以通过 $\text{cnt}_{c,k} = \text{cnt}_{c,k-1} + \text{cnt}_{c,k-2}$ 预处理得到。

预处理复杂度为 $O(|\Sigma| \log V + |X| + |Y|)$，询问复杂度 $O(Q \log V)$，其中 $V = 10^{18}$。

# F. Strongly Connected 2

# F. Strongly Connected 2

有一个有向图，包含 $N$ 个顶点和 $N-1+M$ 条边，顶点和边都有编号。

- 对于 $1 \le i \le M$，边 $i$ 是从顶点 $X_i$ 指向顶点 $Y_i$ 的有向边。
- 对于 $1 \le i \le N-1$，边 $M+i$ 是从顶点 $i+1$ 指向顶点 $i$ 的有向边。

从边 $1, 2, \ldots, M$ 中选择若干条边（可以选零条）的方式共有 $2^M$ 种。在这些方式中，有多少种选择使得在删除被选中的边后，剩余的图是**强连通**的?

请将结果对 998244353 取模后输出。

**数据范围：**

- $2 \le N \le 2 \times 10^5$
- $1 \le M \le 2 \times 10^5$
- $1 \le X_i < Y_i \le N$

# F. Strongly Connected 2

相当于保留剩下的边之后，1 可以到达 $N$。

注意不管怎样，1 能到达的点一定总是一段前缀 $1, 2, ..., k$，因此可以设 $f_k$ 表示只考虑两端点都在 $[1, k]$ 内的这些边，保留一个子集使得 1 能够到达 $k$ 的方案数。

转移我们考虑容斥，如果 1 不能到达 $k$，那么我们枚举实际上 1 能够到达的最远点 $j$，那么边可以分为三部分：

- 两端点都在 $[1, j]$ 内，这部分的边需要满足留下来的能让 1 到达 $j$，方案数为 $f_j$。
- 左端点 $\leq j$，右端点 $> j$ 的边：必须全部删掉，否则 1 就能到达比 $j$ 更靠后的点了。
- 两端点都在 $[j+1, k]$ 内：可以任选删掉或者留下，若有 $c$ 条，方案数为 $2^c$。

# F. Strongly Connected 2

那么我们设 $w(l,r)$ 表示两端点都在 $[l,r]$ 内的边的个数，就有转移方程

$$f_k = 2^{w(1,k)} - \sum_{j=1}^{k-1} f_j \times 2^{w(j+1,k)}$$

可以用线段树维护所有的 $f_j \times 2^{w(j+1,k)}$，每次 $k \to k+1$ 的时候加入所有右端点为 $k+1$ 的边，加入一条边 $(x,k+1)$ 的影响是把 $1 \le j < x$ 的 $w(j+1,k+1)$ 加上 1，也就是把 $2^w$ 乘上 2，那么线段树维护区间乘区间求和即可。

时间复杂度：$O(N \log N)$

# G. Random Subtraction

给定一个由 $N$ 个非负整数组成的序列 $A$。

重复以下操作，直到 $A$ 中只剩下一个元素：

- 从 1 到 $|A|$ 中均匀随机地选择两个不同的整数 $i$ 和 $j$。
- 设 $a = A_i$，$b = A_j$。
- 从 $A$ 中移除第 $i$ 个和第 $j$ 个元素。
- 将 $a - b$ 追加到 $A$ 的末尾。

设最终 $A$ 中唯一的元素为 $x$。求 $x^2$ 的期望值，对 998244353 取模。

**数据范围：** $1 \leq N \leq 2 \times 10^5, 0 \leq A_i < 998244353$

最后得到的 $x$ 一定形如 $\sum_{i=1}^{N} c_i A_i$，其中 $c_i = \pm 1$。

那么 $x^2 = \sum_{i=1}^{N} A_i^2 + \sum_{1\le i,j\le N, i\ne j} c_i c_j A_i A_j$，根据期望的线性性，有

$$\mathbb{E}[x^2] = \sum_{i=1}^{N} A_i^2 + \sum_{1\le i,j\le N, i\ne j} \mathbb{E}[c_i c_j] A_i A_j$$

因此只需要考虑 $\mathbb{E}[c_i c_j]$ 即可。另一方面所有的 $\mathbb{E}[c_i c_j]$ 显然是相等的，因此只需要计算出来 $c_1 c_2$ 的期望。

考虑合并的过程，会形成一棵二叉树，每个点 $u$ 的权值等于它的左儿子减去右儿子权值，叶子节点是所有的 $A_i$。那么每个 $c_i$ 实际上等于 $(-1)^k$，其中 $k$ 是 $i$ 一路走到根作为右儿子的次数。

我们考虑 $c_1c_2$，那么考虑 $1,2$ 的 LCA，发现从这个点往上对 $c_1c_2$ 这个乘积就没有贡献了（每次一定是同时 $\times 1$ 或者同时 $\times(-1)$，不会变）；那么只需要在这个 LCA 处考虑它的期望值。

假设 LCA 是 $u$，两个儿子分别是 $v_1,v_2$，其中 $c_1 \in \operatorname{subtree}(v_1), c_2 \in \operatorname{subtree}(v_2)$，考虑直接写出来 $c_1,c_2$ 的具体分布：

- 如果 $v_1$ 不是叶子，那么由于 $v_1$ 的两个儿子都是等概率作为左右儿子，因此 $c_1$ 一定是 $\frac{1}{2}$ 概率等于 $1$，$\frac{1}{2}$ 概率等于 $-1$。$v_2$ 那一侧也是同理的。

# G. Random Subtraction

那么只要某一个 $v_i$ 不是叶子，它对期望的贡献就都是 0。只有它们两个都是叶子的情况才会产生非零的贡献，这时候贡献一定是 $-1$。

考虑它们都是叶子的概率，相当于 $1, 2$ 中的某个第一次被选的时候就要选到彼此，那么可以设 $f_n$ 表示这个概率，就有（方便起见，用 $C(n)$ 表示 $\binom{n}{2}$）

$$f_n = \frac{1}{C(n)} + \frac{C(n-2)}{C(n)} f_{n-1}$$

意思是，要么以 $\frac{1}{C(n)}$ 的概率直接选中 $1, 2$，要么先选两个剩下的点，然后递归下去。

这样可以递推出 $f_n$，做到 $O(N)$ 求解。

如果你观察力强一点可以发现这个式子导出的 $f_n$ 满足 $f_n = \frac{2}{3(n-1)}$，就不用递推了，可以直接得到答案为 $\sum_{i=1}^{N} A_i^2 - \frac{N}{3} \sum_{1 \le i,j \le N, i \ne j} A_i A_j$。