2021 浙大城市学院新生程序设计竞赛题解

ZUCC ACM Group

Zhejiang University City College

December 18, 2021

A. All in!

签到题，输出"All in!" 即可。

B. Boboge and Tall Building

签到题，输出 $\frac{k(n-1)}{m}$，注意至少输出 6 位小数。

C. Constructive Problem

- 对于 $n \geq 7$，有以下构造方法：$a_0 = n - 4, a_1 = 2, a_2 = 1, a_{n-4} = 1$，其余位置均为 0
- 对于 $n < 7$，可以直接暴力枚举求解，或手工打表等

D. Diseased String

对于每个 y，其后若有连续 x 个 b，则对答案的贡献为 $\max(0, x-1)$。因此对每个 y 处理出其后有多少个连续的 b 即可。

E. Equality

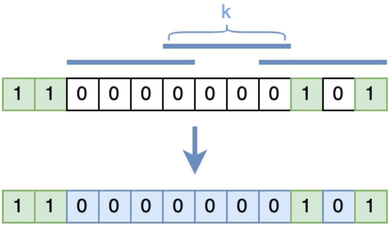
将原数组中最小值视为 1，其余值视为 0，每次操作选定长度 k 且含有 1 的区间置为全 1，问题转化为最少操作次数使得数组全 1

E. Equality

将原数组中最小值视为 1，其余值视为 0，每次操作选定长度 k 且含有 1 的区间置为全 1，问题转化为最少操作次数使得数组全 1

先考虑一种靠谱的贪心：

- 从左向右考虑，当碰到最左端的 0 时，从这个 0 开始，往右使用若干段长度 k 的区间相扣，直到某一段长度 k 的区间内含有 1 时才停止
- 如果右边完全没有 1，则直接从左侧最近的 1 开始往右以 k-1 为步长扩展



E. Equality

考虑另一种看似错误的贪心：

- 对于最左端的 0，直接使用该 0 左侧的 1 往右以 k-1 为步长扩展
- 若该 0 左侧没有 1，即该 0 位于整个数组最左端，则可以直接用数组左侧的边界当作 1，往右扩展

E. Equality

考虑另一种看似错误的贪心：

- 对于最左端的 0，直接使用该 0 左侧的 1 往右以 k-1 为步长扩展
- 若该 0 左侧没有 1，即该 0 位于整个数组最左端，则可以直接用数组左侧的边界当作 1，往右扩展

但实际上两种贪心都是正确的，甚至本质都是一样的，不理解的读者可以自己想一想

F. Future Vision

- 通过 BFS 求出从起点到每个位置的最短时间 $t_{x,y}$
- 按时间顺序考虑剑在 i 时刻出现的位置 (x_i, y_i)，找到其中最早的 $t_{x_i,y_i} \le i$ 的位置即可

Generate 7 Colors

- 可以发现当且仅当 $a_0 \geq a_1 \geq \cdots \geq a_6$ 时有解，其余时刻无解

Generate 7 Colors

- 可以发现当且仅当 $a_0 \geq a_1 \geq \cdots \geq a_6$ 时有解，其余时刻无解
- 将答案要求数量 $a_0, a_1, \ldots, a_6$ 划分为 x 段完整的 $[0, 1, 2, 3, 4, 5, 6]$ 和 y 段不完整的 $[0, 1, \ldots, i](i < 6)$，则有 $x = a_6, y = a_1 - a_6$

Generate 7 Colors

- 可以发现当且仅当 $a_0 \geq a_1 \geq \cdots \geq a_6$ 时有解，其余时刻无解
- 将答案要求数量 $a_0, a_1, \ldots, a_6$ 划分为 x 段完整的 $[0,1,2,3,4,5,6]$ 和 y 段不完整的 $[0,1,\ldots,i](i<6)$，则有 $x=a_6, y=a_1-a_6$
- 所有的完整段均可以接在不完整段的前面，因此答案就是 $\max(1,y)$

H. Hile and Subsequences' MEX

使用 f_n 表示长度 n 序列的答案，考虑从 f_n 递推到 f_{n+1}：

H. Hile and Subsequences' MEX

使用 f_n 表示长度 n 序列的答案，考虑从 f_n 递推到 f_{n+1}：

- 对于 $MEX \in [0, n-1]$ 的子序列，考虑 n 取与不取，这些子序列的方案数将变成原先的 2 倍

H. Hile and Subsequences' MEX

使用 f_n 表示长度 n 序列的答案，考虑从 f_n 递推到 f_{n+1}：

- 对于 $MEX \in [0, n-1]$ 的子序列，考虑 n 取与不取，这些子序列的方案数将变成原先的 2 倍
- 对于 $MEX = n$(i.e. $[0, 1, \ldots, n-1]$)，n 若不取，则 MEX 仍是 n，否则 $MEX = n + 1$

H. Hile and Subsequences' MEX

使用 f_n 表示长度 n 序列的答案，考虑从 f_n 递推到 f_{n+1}：

- 对于 $MEX \in [0, n-1]$ 的子序列，考虑 n 取与不取，这些子序列的方案数将变成原先的 2 倍
- 对于 $MEX = n$(i.e. $[0, 1, \dots, n-1]$)，n 若不取，则 MEX 仍是 n，否则 $MEX = n + 1$

因此有:

$$
\begin{aligned}
f_{n+1} &= (f_n - n) \times 2 + n + (n+1) \\
f_{n+1} &= f_n \times 2 + 1 \\
f_{n+1} + 1 &= (f_n + 1) \times 2 \\
f_n &= 2^n - 1
\end{aligned}
$$

H. Hile and Subsequences' MEX

使用 f_n 表示长度 n 序列的答案，考虑从 f_n 递推到 f_{n+1}：

- 对于 $MEX \in [0, n-1]$ 的子序列，考虑 n 取与不取，这些子序列的方案数将变成原先的 2 倍
- 对于 $MEX = n$(i.e. $[0, 1, \ldots, n-1]$)，n 若不取，则 MEX 仍是 n，否则 $MEX = n+1$

因此有:

$$f_{n+1} = (f_n - n) \times 2 + n + (n+1)$$
$$f_{n+1} = f_n \times 2 + 1$$
$$f_{n+1} + 1 = (f_n + 1) \times 2$$
$$f_n = 2^n - 1$$

快速幂求出 2^n 即可

H. Hile and Subsequences' MEX

使用 f_n 表示长度 n 序列的答案，考虑从 f_n 递推到 f_{n+1}：

- 对于 $MEX \in [0, n-1]$ 的子序列，考虑 n 取与不取，这些子序列的方案数将变成原先的 2 倍
- 对于 $MEX = n$(i.e. $[0, 1, \ldots, n-1]$)，n 若不取，则 MEX 仍是 n，否则 $MEX = n+1$

因此有:

$$f_{n+1} = (f_n - n) \times 2 + n + (n+1)$$
$$f_{n+1} = f_n \times 2 + 1$$
$$f_{n+1} + 1 = (f_n + 1) \times 2$$
$$f_n = 2^n - 1$$

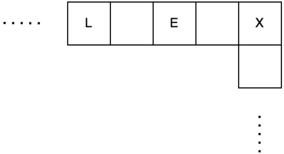
快速幂求出 2^n 即可

也可以直接列出形如 $f_n = n + \sum_{i=0}^{n-1} i \times 2^{n-1-i}$ 的公式，然后列项相消。

或者直接打表找规律

I. If I Catch You

- 可以发现，若每回合均只前进 1 步，守株待兔，即使 Liola 每回合只前进 2 步，在 $2n-4$ 回合后也一定会出现如下情况



- 因此 Eastred 一定能在至多 $2n-3$ 回合内抓到 Liola

I. If I Catch You

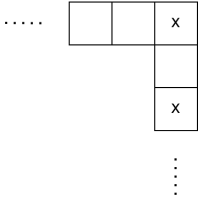
- 而如果 Eastred 采取追逐策略的话，即使 Liola 第一回合只迈出 2 步，Eastred 仍然无法在少于 $2n-3$ 回合内抓到 Liola

I. If I Catch You

- 而如果 Eastred 采取追逐策略的话，即使 Liola 第一回合只迈出 2 步，Eastred 仍然无法在少于 $2n-3$ 回合内抓到 Liola
- 因为只要 Liola 发现 Eastred 在第一回合迈出 4 步后，改用 3 步前进，Eastred 只有一直保持用 4 步前进，才有可能在少于 $2n-3$ 回合内抓到 Liola

I. If I Catch You

- 而如果 Eastred 采取追逐策略的话，即使 Liola 第一回合只迈出 2 步，Eastred 仍然无法在少于 $2n-3$ 回合内抓到 Liola
- 因为只要 Liola 发现 Eastred 在第一回合迈出 4 步后，改用 3 步前进，Eastred 只有一直保持用 4 步前进，才有可能在少于 $2n-3$ 回合内抓到 Liola
- 而只要 $n \geq 3$，通过前两回合，Liola 一定能在自身起点和向后 2 步处均置下陷阱，而 Eastred 若一直保持 4 步前进，则必定会踩中其中一个



J. Jiubei and Codeforces

小模拟，判断下每场比赛前后是否属于同一个分段即可

Klee and Bomb

- 使用并查集维护出在不改变颜色的情况下，所有同色联通块的大小
- 枚举所有点，列举出所有与其相连的联通快，将其中同色块的大小累加到一起，再带上枚举点本身既是该点染为对应颜色的贡献
- 在所有贡献中取最大值即可

L. Lexicographic Order

- 对于末尾不为 a 的字符串，将末尾字符 x 调整成 x-1，在末尾补满 z 到长度 m 即可
- 对于末尾为 a 的字符串，直接删除末尾的 a

END