

2022 Freshman Programming Contest-solution

ZUCC ACM Group

Zhejiang University City College

2022.12.10



A
oB
oC
oD
ooE
oooF
ooG
oH
ooI
ooJ
ooK
ooL
ooEND
o

难度预期

Easy: A C J

Easy - Medium: B L F G

Medium - Hard: I H K

Hard: D E



A - OP

签到题。

直接输出 fengqibisheng, yingyueerlai! (风起必胜, 应约而来)。

时间复杂度: $O(1)$



B - Steel of Heart

对于事件 1，直接获得 800 生命值，并将 5 个敌方英雄的冷却时间初始化为冷却完毕的状态。

对于事件 2，直接获得 H_2 生命值。

对于事件 3，判断一下受到攻击的英雄是否处于冷却完毕的状态，若冷却完毕则将当前生命值代入公式计算出获得的生命值，并修改其冷却状态。

注意触发心之钢的生命加成向下取整，推荐用整数模拟整除更优。

题面信息梳理清楚后其实是一道实现难度较低的签到模拟题。

时间复杂度： $O(m)$



C - Add 9 Zeros

遍历每个值，判断 $+9$ 后的值是否在原数组中存在，若不存在则答案 $+1$ 。

这个过程使用可以用 stl 容器或对原数组进行排序实现。

时间复杂度: $O(n\log(n))$



D - Cutting with Lines I

简要题意：在二维平面上有一个矩形，还有很多一端贴着矩形边界的平行坐标轴的线段将矩形分隔成很多部分，求最大部分的面积。



D - Cutting with Lines I

核心思路：注意到线段只有 2000 条，考虑将所有线段延长至矩形两端，这样矩形会被分成至多 1002001 个小矩形。

做法：直接判断相邻小矩形是否联通，来进行合并，可以采用 dfs、bfs、dsu 等做法。

时间复杂度 $O(q^2)$ ，多加个 $\log$ 也是可以通过的。



E - Cutting with Lines II

简要题意：求最大凸多边形的面积，要求这个凸多边形满足每条边都在某一条所给直线上。



E - Cutting with Lines II

结论：该面积最大的凸多边形一定为平行四边形或者三角形。

做法：枚举所有的三角形和平行四边形计算最大的面积。

std 时间复杂度 $O(n^3 + n\log(n) + n^2)$



E - Cutting with Lines II

反证法证明结论：

设：所构成的所有凸多边形中面积最大的凸多边形 A 中存在两条不相邻且不平行的边 L_1, L_2 ;

延长 L_1, L_2 并相交，记所得到图形的外轮廓为多边形 B 。

则 B 一定满足：

- 1、 B 是凸多边形
- 2、构成 B 的直线为所有构成 A 的直线的子集
- 3、 B 的面积严格大于 A

所以若凸多边形 A 中存在不相邻的非平行边，则 A 一定不是最大的凸多边形。

所以该凸多边形一定不存在不相邻的非平行边。

所以易证该凸多边形一定是平行四边形或三角形。



F - Survivor

简要题意：有 n 个人，每个人初始血量是 a_i ，每分钟会受到 b_i 的伤害，而你有一个总共能使用 k 次的技能，每次能选择一个人将他的血量提高 c_i ，你可以在任意时刻使用技能。一旦一个人的血量小于或等于 0，则他会阵亡。问最多能存活多少人。



F - Survivor

做法：

可以计算出让每个人存活所需要对他使用的技能次数，排序后从小到大贪心取即可，要注意使用 long long。

时间复杂度： $O(n\log(n))$



G - Red Black Tree

做法：

首先可以发现最终状态时一定是若干个不相交的三角形，且一个底长为 m 的三角形中叶子个数为 $\frac{m \times (m+1)}{2}$ 。也就是说，我们只要知道每个三角形的底长就可以算出最后的答案。

于是我们可以记录出每个点扩散到第 n 层的区间，然后对这些区间取并集后的每一个不相交区间分别计算答案然后相加。

求区间并集可以使用差分 + 前缀和。

时间复杂度： $O(n)$



H - Beautiful String

简要题意：给出一个字符串 s ，定义一个长度为 n 且只含有不同字符的字符串 t 是好字符串需要满足以下两个条件之一：

- 1、 t_1 出现在串 s 中且 t 串中的字符的大小是严格递增的。
- 2、 t 串中每个字符都在 s 中出现过。

求不同字符串 t 的个数。



H - Beautiful String

做法：简单的组合数学。

考虑第一种条件，我们可以记录下 s 串中出现过的字母，那么我们可以枚举每一个出现的字母，这里我们将比当前字母大的字母种类数记作为 p ，那么这部分的答案是 C_p^{n-1} 。

考虑第二种条件，我们可以记录下 s 串中出现过的字母，这里将出现过的字母种类数记作为 x ，那么这部分的答案是 A_x^n 。

但是若把这两部分的答案相加，则会多算同时满足两个条件的个数，因此可以简单的容斥一下，将这部分的答案减掉。同时满足两个条件的个数为 C_x^n 。

为了避免取模而将字符集限制在了 18，不过注意要开 long long。

时间复杂度： $O(\sum |s| + 18 \times T)$



I - Digit Problem

结论：

若 $a > 0$ 且 $b > 0$ 时，只有当 $0 \leq c \leq a + b - 1$ 时是有解的。



I - Digit Problem

下面给出一种 $a > 0$ 且 $b > 0$ 时的构造方式：

将 x 固定为 $111\dots1111000\dots000$ 。

若 $c \leq a$ 时， y 可以这么构造：

$$\underbrace{11\dots11}_{a-c \uparrow 1} \underbrace{011\dots1}_{c \uparrow 1} \underbrace{100\dots00}_{b-1 \uparrow 0}$$

若 $a < c \leq a + b - 1$ 时， y 可以这么构造：

$$\underbrace{011\dots1}_{a-1 \uparrow 1} \underbrace{100\dots00}_{c-a \uparrow 0} 100\dots00$$

最后对于 $a = 0$ 或 $b = 0$ 的情况进行特判即可。

时间复杂度： $O(a + b)$



J - Simple Game

简要题意：有 n 个数，Bob 和 Alice 两个人轮流取数，取完为止。
最后若两者取数之和的差为奇数则 Alice 获胜，否则 Bob 获胜。



J - Simple Game

ai 解决了这个问题:



Alice and Bob are playing a game on a sequence $a_1, a_2, \dots, a_n$ of length n . They move in turns, and Alice moves first.

At each player's turn, they should select an integer and remove it from the sequence. The game ends when there is no integer left in the sequence.

Assume the sum of integers selected by Alice is S_1 , and the sum of integers selected by Bob is S_2 .

If the difference between S_1 and S_2 (ie. $|S_1 - S_2|$) is odd, Alice wins; Otherwise, Bob wins.

Your task is to determine who will win the game if both players play optimally.



如果 Alice 和 Bob 都采取最优策略, 则无论他们在哪一步选择了哪个整数, 都不会改变游戏的结果。因此, 我们可以先计算出序列中所有整数的总和, 然后判断这个总和的奇偶性。如果总和是奇数, 那么 Alice 必胜; 否则 Bob 必胜。

时间复杂度: $O(n)$



K - Bit

简要题意：有 n 个固定操作，每个操作都是与、或、异或，有 q 个询问，每个询问给出一个数 r ，请你在区间 $[0, r]$ 中选一个数 x ，使它经过这 n 个操作之后得到的数是最大的。



K - Bit

做法：二进制拆位，贪心。

这其实是一个非常套路的题目，由于这 n 个操作是固定的，而且与二进制有关，所以不妨从二进制的角度去考虑。

我们可以预先处理出二进制第 i 位初始为 1 或 0 时，经过这 n 次操作之后它的变化。然后对于每一个询问，从二进制的高位到低位枚举，判断数 x 的当前位应该为 1 还是 0，如果当前位初始为 0 时能得到 1，则优先将当前位置为 0。若当前位初始为 1 时能得到 1，且将当前位置为 1 后仍不超过 r 的限制，也将当前位置为 1。这么贪心选取出的 x 一定满足它经过这 n 个操作之后得到的数是最大的。

时间复杂度： $O(30 \times (n + q))$



L - Elden Ring

简要题意：两个大小为 n 的环并行从 1 开始报数，报出的数为 m 的倍数时交换位置，问报出数字 k 时环的最终状态。



L - Elden Ring

该题的最初版本为 $k \leq 10^9$ 。

考虑只存在一个大小为 n 的环进行报数，报出的数为 m 的倍数时喊一声，在这个模式下易得存在循环节 $lcm(n, m)$ 。

现在两个环报数并加入交换操作，所以回到初始状态的循环节为 $2lcm(n, m)$ 。

可将 k 对 $2 \times lcm(n, m)$ 取模后再进行模拟，轮数上限存在于 n, m 互质时为 $2nm \leq 2 \times 10^6$ ，暴力模拟即可。

最终考虑到比赛整体难度，遂将 k 削弱至 10^6 ，所以直接暴力模拟即可通过。

时间复杂度： $O(k)$ 或 $O(lcm(n, m))$

本题也有 $O(n)$ 的做法，有兴趣的同学可以自行研究。



END

END

