

2021 Freshmen Cup Solution

Weierstras,NaVi_Awson,jjikkollp,Sinoey,salix_leaf,walk_alone,yzyyyylx

HUST ACM-ICPC Team

2021 年 10 月 17 日

A. Calculus

这是个椭圆积分显然没法直接算。考虑放缩。当 $0 \leq t \leq 1$ 时, $f(t) = (1+t)\sqrt{2-t}$ 递增, 故有 (记答案为 I)

$$I \geq \int_0^{2\pi} \frac{dx}{f(1)} = \pi \approx 3.14$$

$$I \leq \int_0^{2\pi} \frac{dx}{f(0)} = \sqrt{2}\pi \approx 4.44$$

所以只需要输出 3 或 4 就有 50% 的概率通过 (当然善良的出题人已经告诉你答案在 5 以内了, 所以直接试一遍就能过)

B. Rule 110

Rule 110 是一个著名的元胞自动机规则，它被认为是最简单的图灵完备系统。这个题要我们模拟这个系统，按照题意做即可，注意开头结尾要补充 0。

C. Make it in a Line

小学数数题。

一共有 3 种情况：与边平行，面对角线，体对角线。分类讨论即可。

答案为 $3n^2 + 6n + 4$ （小常识：正方体有 6 个面 12 条棱 8 个顶点）

D. Hard Tasks

显然每一位相互独立。 $i-1$ 的个位只能为 $0, 1, 2$ ，其他位只能为 $0, 1, 2, 3$ 。

- 30 分：直接枚举小于 n 的所有数判断是否合法，复杂度 $O(n \log n)$
- 60 分：枚举每一位，判断是否小于 n ，复杂度 $O(n^{\lg 4})$
- 100 分：数位 dp
 - $[0, 10^k)$ 中有 $3 \cdot 4^{k-1}$ 个数合法
 - 从大到小枚举每一位。假设当前枚举到了第 k 位，统计前 $k+1$ 位与 n 相同的答案。有以下两种情况：
 - 第 k 位是最低位 ($k=0$)，答案加上 $\min(n_k + 1, 3)$
 - 第 k 位不是最低位，答案加上 $\min(n_k + 1, 4) \cdot 3 \cdot 4^{k-1}$ 。当 $n_k > 3$ 时应直接退出循环，因为后面所有的数都可以选
 - 复杂度 $O(\log n)$

E. 2584

灵感来自经典的 2048 游戏，只要简单模拟即可。

需要注意最后的答案会超过 2584，所以需要多算几项斐波那契数列（std 里预处理了前 45 项）

小技巧：只需要写一个向右移的函数，别的方向可以先旋转/翻转再向右移然后再转回来。

F. Do Not Play Nim

做这题似乎只需要一点点博弈基础（下面记 Alice 为 A, Bob 为 B）
很明显在这个博弈中 A 是有优势的，推几个例子（或者脑补一下）会发现 A 有一些必胜的策略（下面的仅供参考，不一定是最简单的，推出其他结论能做也行）：

- 只有一堆，A 拿完直接赢；
- 两堆以上，如果有某堆个数大于等于 3，A 在第一次操作把这堆拿成 1 个，然后保留这堆先不动，那么显然 B 不能动这堆。
对于剩下的堆继续玩，可能：
 - 剩下的堆没拿完 B 就不能操作了，A 赢
 - 剩下的堆没拿完 A 就不能操作了？不存在，只要没拿完 A 就能操作
 - 剩下堆拿完了，轮到 A，A 拿完这堆，没了，B 不能操作了，A 赢
 - 剩下堆拿完了，轮到 B，B 不能拿这堆，A 赢

所以这种情况 A 赢；

- 三堆以上，它们每次操作只能拿一堆，那么总共至少要操作 3 次。A 在第一次操作拿走最大堆，不管 B 第一次操作怎么拿，A 在第二次操作拿 1 个石子，现在 A 拿的石子之和大于每堆石子个数，B 不能拿了，A 赢；
- 除去以上三种情况只剩 1 1, 1 2, 2 2 的情况手推一下就可以知道只有 1 1 的情况 B 能赢；

综上，除了只有 2 堆都是 1 个的情况 B 赢，其他 A 赢。

后记：这题本来想出 A 每次只能取 1 个，B 每次取的个数不少于 A 取过的个数之和的，但是没想好解法，然后 yzy 说把 A 改成每次取任意个就很可做了，然后就改了。大家可以想想这个咋做。

G. Awson Loves Chipmunks

- 30 分：前缀和把所有区间的和算出来，排序/快速选择找到第 m 小元素。复杂度 $O(n^2 \log n)$ 或 $O(n^2)$
- 100 分：二分答案。
 - 记和不超过 x 的区间个数为 $f(x)$ 。若 $f(x) \geq m$ 且 $f(x-1) < m$ ，则答案为 x
 - 考虑如何快速计算 $f(x)$ 。预处理两个数组 $R = \sum_{k < j \leq i} a_j$ ($i > k$) 与 $L = \sum_{i \leq j \leq k} a_j$ ($i \leq k$)，则每一个包含 k 的区间都可以表示成 L 和 R 中两个元素之和。将两数组分别排序，接下来有两种做法：
 - 对 R 中的每个元素 r 二分找到 L 中小于 $x - r$ 的位置，全部加起来即可。复杂度 $O(n \log n \log W)$ (W 为值域)
 - 由于 L 和 R 均单调，可以用双指针优化掉二分，复杂度 $O(n \log W)$

H. Chipmunk Sorting

当 $a = -1$ 时答案显然为 0。

当 $a = 1$ 时，考虑交换 h_i 和 h_j 。对于任意满足 $h_i > h_k > h_j$ 的 k ，都能使序列中的逆序对数减 2。因此每次交换后逆序对的数量会减少

$$1 + 2 \sum_{i \leq k \leq j} [h_i > h_k > h_j]$$

把所有交换的贡献累加，结果应当等于原序列的逆序对数 R 。假设交换次数为 S ，则有 $S + 2 \text{ans} = R \Rightarrow \text{ans} = (R - S)/2$ 。若要最大化 ans 只需最小化交换次数，其中最小交换次数为 $n - \text{置换环的数量}$ （也可以每次把 a_i 移到第 i 位统计需要移动多少次）

I. Awson is God

记 cnt_i 表示告诉 Awson 数字 i 的人数（大于等于 n 的数字都可以看成 n ，反正都是错的要改）。

以下讨论可能无欺骗者（即通过给的 n 个数能够找到一组合理解）的情况。

- 结论 1: n 个告诉 Awson 的数必须满足仅存在一种或两种数，若有两种则必相差 1。
可以考虑任意两个人 A 和 B，它们看到的除它们两人之外的“favorite number”的种类数为 x ，则 A/B 告诉 Awson 的数字要么是 x ，要么是 $x+1$ 。
- 结论 2: 若所有人告诉 Awson 的数都是 a ，只有两种合理情况。
 - 1 每个人“favorite number”都相同，则 $a = n - 1$;
 - 2 每种“favorite number”至少出现两次，则 $2a \leq n$;
 - 3 不存在第三种，否则不可能只有一种数
- 结论 3: 若存在两种告诉 Awson 的数 a 和 $a+1$ ，其中 $a+1$ 表示这个人的“favorite number”在 n 个人中至少出现了两次， a 表示这个人的“favorite number”独一无二。需要满足两个条件即可构造合理解（比赛中 65 分的好像是没考虑这两个条件）。
 - 1 $\text{cnt}_a < a+1$ ；即有 cnt_a 种独一无二的“favorite number”，那么告诉 Awson 数 $a+1$ 的人除了看到所有独一无二的“favorite number”还至少能看到和自己一样的“favorite number”；
 - 2 $2(a+1 - \text{cnt}_a) \leq n - \text{cnt}_a$ ；即可以删去独一无二的“favorite number”，剩下的满足上述结论 2 的第二种合理情况

I. Awson is God

然后下面考虑最少几个欺骗者。

根据上面的结论，要把 n 个告诉 Awson 的数全改为 a 或者全改为 a 和 $a+1$ ， $O(n)$ 枚举所有可能的 a ， $O(1)$ 计算修改为合理解的代价即可。

- 全改为 a ：
若 a 满足结论 2，则答案对 $n - \text{cnt}_a$ 取最小值
- 全改为 a 和 $a+1$ ：
根据结论 3 条件，要满足 $2(a+1) - n \leq \text{cnt}_a < a+1$

Proof

显然 $2(a+1) - n > a$ 时改为 a 和 $a+1$ 是不合法的。
即 $n \leq a+1$ 不合法，控制一下 a 的枚举范围就好。

I. Awson is God

假设不为 a 和 $a+1$ 的数字有 tot 个，那么考虑把 tot 个全改为 a 和 $a+1$ 时要如何满足上述不等式。

- 1 若 $\text{cnt}_a + \text{tot} < 2(a+1) - n$ ，说明 tot 个全改 a 都不能满足不等式，需要把一些 $a+1$ 改为 a ；
- 2 若 $a+1 \leq \text{cnt}_a$ ，说明 tot 个全改 $a+1$ 都不能满足不等式，需要把一些 a 改为 $a+1$ ；
- 3 不满足上述 1 或 2 则说明可以通过分配这 tot 个使得不等式成立

知道上面 3 点修改次数应该容易计算，然后答案对它取最小值即可。

J. Gift

注意到需要满足的第二个条件，显然可以发现需要满足选出的点集在一个匹配上。（也就是说存在一个匹配使得选出的点集是这个匹配所包含的点集的子集）。

那么我们可以利用 Hall 定理状压 DP 出满足条件的左右两侧的点集。可以证明，从左侧和右侧分别选出两个满足 Hall 定理的子集后组合出来的点集一定满足都在一个匹配上。

再考虑第一个条件，我们将两侧的合法的点集的点权异或和用 FWT 合并即可。

K. Easy Homework

考虑一种暴力：我们将所有的操作保存并不下放到实际序列中，当出现询问 S_i 时，检查所有已经保存的操作。如果某个操作的区间中包含了 i ，则这个操作对本次询问的答案产生贡献。具体的做法可以是保存 i 在序列中出现的每一个位置，将序列的左右端点在其中进行二分来判断是否产生贡献。

这种暴力毫无疑问是会超时的，故我们尝试将这个暴力优化。将输入的操作依次存储，但当输入的修改操作达到了一个阈值 T ，则我们将所有的操作以一个较优的时间复杂度整体下放到实际序列中的每一个出现的颜色，即整体进行重构。

如何重构？考虑重构的问题实际上变成了只存在修改操作，并一次计算得每一个颜色的权值。我们考虑将操作的左右端点和序列放在一起，并从左往右扫描。如果出现了一个操作的左端点，我们在一个临时的初始为 0 的权值序列上加上这个操作的权值。检查每一个位置的 S_x 时，我们记录 x 在序列中出现的上一次位置，并记录当前位置到上一次位置在临时权值序列的区间和，即能不重复地计算一个区间对 S_x 的贡献。当检查到一个操作的右端点时，我们将其左端点在临时权值序列上的权值删去。动态单点修改区间查询的权值和问题我们可以通过树状树组/线断树等数据结构进行维护。当然重构的做法并不唯一。

一次重构的复杂度为 $O(n \log n)$ ，一次询问的复杂度为 $O(T \log n)$ ，最终总复杂度为 $O(nT \log n + \frac{n}{T} n \log n)$ 。由基本不等式可以计算得当 $T = \sqrt{n}$ 时，总复杂度最优为 $O(n\sqrt{n} \log n)$ 。

L. Intervals

- 考虑每一段对答案的贡献
- 设覆盖当前段的下标为 $i_1, i_2, \dots, i_k$
- 如果选出的子区间覆盖了至少一个 i_j , 将答案加上 1
 - 注意到覆盖的下标一定是连续的
 - 用点减边的技巧, 对覆盖每个下标的区间加上贡献, 对覆盖相邻两个下标的区间减去贡献
- 用扫描线维护当前下标的集合
- 为了包政府再度, 旨在 $[i_j, i_{j+1}]$ 被删除时计算贡献
- 剩下就是二位数组, 扫描线树状数组离线做即可。

谢谢大家