NOIP 2024 模拟赛 1 总结与题解

Fly Wang

2024 年 7 月 19 日

前言

本次是 Fly Wang 第一次给学弟学妹出模拟赛，之前在本届中出过一次，出了很多锅……

个人感觉符合 NOIP 的风格，除开 T3 其他题不要快读的话基本不超过 1.5KB，也算是不错的题目了（

难度可能总体和 NOIP 差不多，主要是 T3 比较难写，我个人调了 2h 才调出来（

考试情况很抽象啊，怎么这么多人写了 T1 后面 1 pts 都没写。只有一个人过了 T2 好离谱，T3 的第一个性质都没人看出来吗？我还费尽心思写半天

各位能力仍需提高啊……

个人预测：

1. NOIP 一等奖线：100 + 30 + 15 + 5 - ? = 140
2. 三倍省队线：100 + 100 + 30 + 5 - ? = 225
3. 省队线：100 + 100 + 60 + 15 或者 100 + 100 + 35 + 55 - ? = 280

1 游戏（number）

Idea: Original.

1.1 Subtask 1(15 pts)

暴力枚举每次是否攻击即可。

期望得分 15 pts。

1.2 Subtask 2(10 pts)

任意多项式算法吧。我也不太清楚（

1.3 Subtask 3(20 pts)

考虑 DP，记录 $dp(i,s)$ 表示 i 次操作后血量能否为 s。用 `std::bitset` 记录状态并转移即可通过。

复杂度 $O(\frac{na}{w})$。期望得分 45 pts。

1.4 Subtask 4(15 pts)

对 Subtask 3 进行优化，注意到显然有一个 $a+b$ 的循环节，故可以优化。

复杂度 $O(\frac{a^2}{w})$。期望得分 60 pts。

1.5 Subtask 5(15 pts)

考虑贪心：能减则减，不能减就加。正确性证明略。

复杂度 $O(n)$。期望得分 60 pts。

1.6 Subtask 6(25 pts)

注意到只有加减的次数是有意义的，而且只需要满足最后血量 ≥ 0 即可（可以先操作所有的加，在操作剩下的减）。二分判断或者神秘数学方式计算即可。

（话说这么简单的算法放这么多 Subtask 合适吗）

复杂度 $O(\log n)$ 或 $O(1)$。期望得分 100 pts。

2 路径（path）

Idea: ARC150C。

2.1 Subtask 1(15 pts)

爆搜。略。

2.2 Subtask 2(10 pts)

给一些神秘的 $O(nm)$ 做法的。应该比较多的（

2.3 Subtask 3(15 pts)

显然当前是一棵树。直接记录到每个点能匹配的最长前缀即可。复杂度 $O(n)$。

2.4 Subtask 4(20 pts)

考虑 DP，记录 $dp(x,i)$ 表示运动到 x，长度为 i 的是否可行，初始均为 1，按照 i 从 $1 \to k$ 转移。考虑 $dp(x,i)$ 向 $dp(y,i+1)$ 转移，若 $dp(x,i)=0$ 并且 x,y 存在一条路径不含 b_{i+1}，那么 $dp(y,i+1)$ 也为 0。

复杂度 $O(mk)$，期望得分 45 pts，结合 Subtask 3 可以得到 60 pts。

2.5 Subtask 5,6(40 pts)

注意到存在某个限度 t 使得 $dp(x,i)(i \geq t)$ 都是 False。因此我们只需记录这个 t 即可。

记录 $f(x)$ 表示任意情况下的最长匹配，即对应题目中所求的答案。考虑按 $f(x)$ 从小到大转移，注意到若 $v_y = b_{f(x)+1}$，则让 $f(y)$ 对 $f(x)+1$ 取 min，否则就是对 $f(x)$ 取 min。

用一个 `std::priority_queue` 来让 $f(x)$ 从小到大即可转移。

复杂度 $O(m \log n)$。期望得分 100 pts。

3 回环（tree）

Idea: ZR 某道神秘题，记不得了。

3.1 Subtask 1(15 pts)

暴力枚举所有可能的排列。复杂度 $O(n!m)$，不知道能不能过。

3.2 Subtask 2,3(20 pts)

考虑直接计算距离。可以想到拆贡献，若一个边左右分别出现 x, y 个上课地点（$x+y=k$），可以看到这条边的贡献次数为 $\min(x, y)$。证明过程可以考虑将两边的点交叉排列。

因此我们只需统计子树内有多少上课地点就可以直接计算答案了并在每次修改后动态维护。答案应该为 $\sum \min\{sz(x), k-sz(x)\} \times w(x)$。

复杂度 $O(nm)$，期望得分 35 pts。

3.3 Subtask 4(15 pts)

注意到现在树高只有 $O(\log n)$，那么我们可以使用 $O(nh)$ 的算法来达到 $O(n\log n)$ 的复杂度。

注意到我们维护的信息只和子树内部的上课地点个数有关，因此我们每更改一次只需要修改其到 $root$ 这条链上的点的信息即可。

复杂度 $O(n\log n)$，期望得分 50 pts。

3.4 Subtask 5(10 pts)

可能有比较奇怪的和 k 复杂度相关的做法，比如把 k 个点排序后得到最佳顺序之类的？反正我还没想到（

3.5 Subtask 6,7(40 pts)

根据上面的分析，我们现在就是将 $O(h)$ 和高度相关的复杂度使用数据结构优化。

注意到我们加如某一个点的时候，会对它的祖先都产生影响，其中可能会是 $+1(sz < \frac{k}{2})$ 或者 $-1(sz \geq \frac{k}{2})$。据此维护。那么我们只需要重点关注那些 $\frac{k}{2}$ 附近的点产生的贡献会发生变化 $(+1 \to -1)$。

我们在其祖先上寻找 $sz = \frac{k}{2}$ 的点，因为他们原来贡献是 $+1$，这一次贡献会变成 -1，因此需要做出改变。由于在祖先这条链上 sz 是不减的，因此 $sz = \frac{k}{2}$ 的点是连续的，可以使用二分来找到。删除某个点同样是寻找 $sz = \frac{k}{2}$ 的点。

至于维护链的方式，可以使用树剖或者 LCT 来实现。本 std 使用的是树剖。

复杂度 $O(m \log^2 n)$ 或者 $O(m \log n)$。期望得分 75 - 100 pts。

4 发绳（rope）

Idea: Original.

4.1 Subtask 1, 2(10 pts)

高中数学几何概型手搓。$n = 3$ 可以利用面积理解。

4.2 Subtask 3(10 pts)

考虑本题难点。容易发现整数至少是可做的，而小数的存在使得本题无从下手。

注意到我们并不关心小数实际的大小，我们只需要他们的差和整数比较，同时任意两个小数相同的概率为 0，因此我们只需要这 $n - 1$ 切点的小数的相对大小。

那么我们可以**选择枚举 $n - 1$ 小数相对大小的排列，然后再按照大小顺序依次枚举**。注意如果 $p_i > p_{i+1}$，即 i 的小数比 $i + 1$ 的大，则 $i + 1$ 整数部分不能和 i 相同。

复杂度爆搜，期望得分 15 pts（能过 1 和 3）。

4.3 Subtask 5(15 pts)

注意到我们枚举相对大小的时候只关注了 $p_i > p_{i+1}$ 的个数（定义为排列的下降数），然后让 L 减去个数后再按照整数问题进行计算。

首先考虑前面的问题，我们使用 $dp(n, t, last)$ 表示长度为 n，已经有 t 个下降数，且最后一个数是 $last$ 的个数。转移即枚举新加在后面的数是多少即可。复杂度 $O(n^4)$。

然后考虑在剩 $L - cnt$ 的情况下计算 $\geq K$ 的情况。考虑 $f(len, i, t)$ 表示 len 切出 i 段，其中 t 段 $\geq K$。转移就枚举下一次切哪里即可。复杂度 $O(n^2L^2)$。

总复杂度 $O(n^2L^2+n^4)$，期望得分 30 pts。

4.4 Subtask 7(15 pts)

对上面的 DP 过程进行简单的优化，显然 $dp(n,t,last)$ 中 $last$ 一维是可前缀和转移的，$f(len,i,t)$ 中 len 是可前缀和转移的。

总复杂度 $O(n^2L+n^3)$，期望得分 55 pts。

4.5 Subtask 6, 8(20 pts)

给各位使用 $O(nL+n^3)$ 或者 $O(L+n^3)$ 之类神秘算法的。我也不太清楚（

4.6 Subtask 9, 10 (30 pts)

上面两个 DP 状态已经是三方的了，不好优化。我们考虑设计新的方式计算。

对于第一个计算下降数，可参考 P5825 非多项式的 $O(n^2)$ 的做法：考虑记录 $dp(n,m)$ 表示长度为 n 的排列，下降数为 m 的个数。转移即考虑 n 插入的位置，若从 $dp(n-1,m-1)$ 则可以插入在非下降位置或开头，从 $dp(n-1,m)$ 则可以插入在下降位置或结尾。那么可以得到转移式为 $dp(n,m)=(n-m)dp(n-1,m-1)+(m+1)dp(n-1,m)$

再考虑第二个整数形式的计算。我们考虑**钦定 i 个的长度大于 K（不是有 i 个而是钦定 i 个）**的方案数，显然是可以通过组合数插板法解决的。然后令 $f(i)$ 为钦定 i 个的答案，$g(i)$ 为实际答案，钦定 i 个就是可能把 j 个的答案计算了 $\binom{j}{i}$ 次，可以通过类似容斥计算。那么我们可以得到计算式：$g(i)=f(i)-\sum_{j>i}g(j)\binom{j}{i}$

输出概率就把 $g(i)$ 加起来做分母即可。

复杂度 $O(n^2+L)$，期望得分 100 pts。