NOIP 模拟赛 II

Prepared by water_tomato

题解

A 货币代码

- 20 分做法：
- 由于斐波那契数的增长是 log 级别的，我们可以枚举前两个数，在 $O(n^2\log n)$ 的时间内找出全部 n^2 个可能的字符串。可以暴力与之前的 n^2 的串进行对比来确定新串是否是重复的，通过暴力比对来确定两个串是否相同。复杂度为 $O(n^4\log n)$，常数较少甚至可以获得 40 分。
- 40 分做法：
- 使用字符串哈希判断两个串是否相同，复杂度变为 $O(n^4)$。
- 测试点 $9 \sim 12$：
- 问题转化为有字符串的长度有多少种可能，所以不再需要判断串是否相同，枚举出所有串，记录所有可能的长度即可。

A 货币代码

- 100 分做法：
- 法一：把字符串哈希的值存入一个 set 或 map 中，判断一个串是否出现过的复杂度变为了 $O(\log n)$，综合复杂度为 $O(n^2 \log n)$.
- 法二：建一棵 Trie 树，把每个串都塞入 Trie 树并在其终点打标记，如果串已经存在就不计入答案，综合复杂度也为 $O(n^2 \log n)$ (即总串长)。

B Sumplete 游戏

- 20 分做法：暴力枚举所有情况。
- 测试点 $3\sim4$ 留给各种神秘的 $O(n^3)$ 或 $O(n^4)$ 做法。
- 测试点 $5\sim7$：
- 正负本质相同，这里不妨考虑全正的情况。
- 我们首先可以肯定，每一行和每一列所需的总和介于 0 和 n 之间，接着我们依次确定每行保留哪些条目：
- 假设当前行的要求和为 s_i，那么保留要求和最大的 s_i 列的条目。如果在任何一步中，我们都无法找到求和为正的 s_i 列，那么原谜题就无解了，这个算法显然是正确的。
- 注意需要在处理行时对列的所需和进行排序。直接实现这种排序需要 $O(n^2\log n)$，这已经足以通过这个问题。不过，需要注意的是，所需的列总和介于 0 和 n 之间。因此，我们可以使用计数排序等方法在 $O(n^2)$ 内解决这个问题。不过由于算法常数不大，$O(n^2\log n)$ 足以通过。

B Sumplete 游戏

- 100 分做法：
- 考虑简化题意。对于数字 1 的条目，可以留 0（划掉）或留 1（保留）。对于数字 −1 的条目，可以留 0（划掉）或留 −1（保留）。通过在这些 −1 条目的相应行/列总和中加入 1 并翻转结果（划掉为 1，保留为 0），可以将问题简化为求解一个所有条目为 1 的 Sumplete 网格，转化为 5 ~ 7 的做法。
- 最后提一句，这题本来是构造题，还需要输出构造方案，不过我懒得写 spj 就没要求输出方案，可以想到这个方案的输出其实是比较简单的，只需要排序时记录一下是哪一列即可。

C 高速公路

- 首先我们发现我们根本不关心关键点的具体位置，只关心他们的相对位置，因此我们把所有关键点按照位置排序。
- 10 分做法：暴力枚举所有可能。
- 测试点 2 ~ 4：也许你们有一些绝妙的 DP 和小规模暴力+贪心。
- 测试点 5,6，引导你们去思考只放一种类型的收费亭应该怎么是最优的，具体上：

C 高速公路

- 我们把每个连续的 EEETTT（连续 E 后接着连续 T）看成一段，最终的段数为 k。
- 全放 b：在除了段与段之间的所有相邻关键点间全部放置 b 类收费站，代价为 $(n-k)\times b$
- 全放 a：除了第一段的最后一个 T 和最后一段的第一个 E，其他全部放 a 类收费站，代价为 $(n-2)\times a$。注意，如果总共只有一段，则应为 $(n-1)\times a$
- 混合放：对于第一段，除了最后一个 T，其余均放 a 类；对于最后一段，除了第一个 E，其他均放 a 类。对于中间段，除了最后一个 E 和最后一个 T，其他均放 a 类，再在该段的 ET 交界处放 b 类。代价为 $(n-k)\times a+(k-2)\times(b-a)$。注意，如果总共只有一段，则应为 $(n-1)\times a$
- 通过对于要求的分析，你知道（或猜到）不存在其他的更优方案，故三种方案取最小值即可。

D 加数

- 10 分做法：暴力枚举所有可能性。
- 测试点 3 ~ 8 都不关键，主要基于不同参数的各种 DP 做法等。
- 特殊性质 A：总共只有两种取值，可以通过一些贪心策略解决，比如先横扫，再找到较大高度的最大间隔，然后左右扫。

D 加数

- 特殊性质 B：解题的关键性质。手玩一些满足条件 B 的数据，发现答案等于 $\sum_{i=1}^{n} y_i$，具体地：
- **Key 1**: 考虑什么情况下，答案等于 $\sum_{i=1}^{n} y_i$。
- 发现当 $y_1 \geq \sum_{i=1}^{n-1} \max(0, y_i - y_{i+1})$ 或 $y_n \geq \sum_{i=1}^{n-1} \max(0, y_{i+1} - y_i)$ 时符合。
- 证明：如果对于每个 k 都有 $y_k \geq \sum_{i=k}^{n-1} \max(0, y_i - y_{i+1})$，那么对于任何一个 $y_i > y_{i+1}$ 的情况，我们都可以令 $y_1 \sim y_i$ 全部减去 $y_i - y_{i+1}$，这样最终会形成一个 $y_1 = y_2 = \cdots = y_k \neq y_k + 1 = \cdots = y_n$ 的形式，容易发现是符合的。然后你可以进行推导，如果 $y_k < \sum_{i=k}^{n-1} \max(0, y_i - y_{i+1})$，则必然有 $y_{\mathrm{k}-1} < \sum_{i=k-1}^{n-1} \max(0, y_i - y_{i+1})$，因此也有 $y_1 < \sum_{i=1}^{n-1} \max(0, y_i - y_{i+1})$，由逆否命题知，我们只需要 $y_1 \geq \sum_{i=1}^{n-1} \max(0, y_i - y_{i+1})$ 即可。同理可知 $y_n \geq \sum_{i=1}^{n-1} \max(0, y_{i+1} - y_i)$ 也成立。

D 加数

- 然后我们还可以得到 **Key 2**：由对称关系知，其中一者成立，另一者也成立，即 $y_1 + y_n \geq \sum_{i=1}^{n-1} |y_i - y_{i+1}|$ 时，有答案等于 $\sum_{i=1}^{n} y_i$。
- 对称关系怎么来的？首先，显然有 $y_1 + \sum_{i=1}^{k-1} (y_{i+1} - y_i) = y_n$，将求和式子中的负项留在左边，正向移到右边，则有 $y_1 - \sum_{i=1}^{n-1} \max(0, y_i - y_{i+1}) = y_n - \sum_{i=1}^{n-1} \max(0, y_{i+1} - y_i)$，故有上述两式等价。
- 有了以上两个结论，我们再考虑**满分解法**：

D 加数

- 由 Key 1&2, 问题转化为，对于题目给定的 y_i 数列，我们可以令其中的某些 y_i 加上某些数，形成一个新的数列 x_i，x_i 的和即为答案，那么我们要做的就是使这些加的数尽可能少。不妨记 $M = y_1 + y_n - sum_{i=1}^{n-1}|y_i - y_{i+1}|$，我们要使得 $M \geq 0$ 。
- 我们发现，若数列中有若干个相同的数 $y_l = y_{l+1} = \cdots = y_r$，如果有 $y_{l-1} \geq y_l$ 且 $y_{r+1} \geq y_r = y_l$，那么我们让 $y_l \sim y_r$ 全部加上 1，就可以使得 M 加上 2，而这一操作的代价是 $r - l + 1$。显然，我们希望找到尽可能短的 $y_l \sim y_r$ 序列满足上述条件，如果你暴力去找，根据你不同的找法，你可以通过测试点 $3 \sim 4$ 或 $7 \sim 8$（感觉这个部分分给少了，应该多给一点小范围的分的）。

D 加数

- **key 3**：考虑怎么尽可能快而准确地找到当前数列中最短的合法序列。
- 我们以 y_i 为权值，建立一棵大根堆的笛卡尔树，可以发现，对于每一个点，对其操作时的长度就是以其为根的子树的大小，其可以加的最大数值就是其与其父亲节点的差值。然后，我们按照子树大小从小到大遍历所有点并更新 M 和答案，直至 $M \geq 0$ 时结束遍历。
- 由于子树大小不超过 n，可以用桶排做到 $O(n)$ 的时间复杂度。也可直接 sort，时间复杂度为 $O(n\text{log}n)$，足够通过。
- 至此为满分做法。

后记

最后两周，我要做什么？

后记

- 看看 NOIp 大纲
- 看看 CSP
- 关注一下常考知识点
- 关注一下 CSP 没考的知识点

联系方式

- 潘伦可（water_tomato）
- QQ：3279309224
- 微信：water_tomato
- 博客：www.watertomato.com
- 欢迎一起玩

谢谢！