

必可高水平众筹赛 第一试

Para

2024 年 10 月 6 日

目录

- 1 吐槽环节
- 2 T1 数字 (number)
- 3 T2 加训 (train)
- 4 T3 排序 (sort)
- 5 T4 改造 (modification)
- 6 题目来源

吐槽环节

1 吐槽环节

2 T1 数字 (number)

3 T2 加训 (train)

4 T3 排序 (sort)

5 T4 改造 (modification)

6 题目来源

T1 数字 (number)

这就是一道简单题。

- 算法 1: 读完题就可以发现这是一道简单的数位 dp。注意一下前导 0 的处理即可。期望得分 100。
- 算法 2.1: 对于性质 ABC, 考虑 $a_1 = 9$ 的情况, 那么我们的数字集合只有零到八, n 可以看做一个九进制数, 我们想求不超过 n 的元素个数等价于将 n 这个九进制数转换为十进制数。在考虑 $a_1 \neq 9$ 的情况, 实质上是将 n 中大于 a_1 的元素减一后再转换为十进制数。
- 算法 2.2: 对于性质 AC, 仍然可以看做进制的转换, 只不过是 k 进制转为十进制。

T1 数字 (number)

- 算法 2.3: 对于性质 C, 此时增加了对 0 的处理, 要额外注意前导零的存在。
- 算法 2.4: 当 n 中包含了 a_i 时, 我们如果能快速的找到比 n 小且不包含 a_i 的元素中的最大值, 就转换为算法 2.3。我们找到 n 中为 a_i 的最高位, 设其为 t , 尝试将其减小。如果能减小, 则将比 t 的位全部变为最大值。否则, 我们依次去找比 t 更高的位, 依次进行相同的判断。若不成功, 我们直接将最高位变为 0, 其他位变为最大值。期望得分 100。

1 吐槽环节

2 T1 数字 (number)

3 T2 加训 (train)

4 T3 排序 (sort)

5 T4 改造 (modification)

6 题目来源

T2 加训 (train)

- 算法 1: 注意到从每个区域出发移动步数为 n^2 级别, 直接模拟复杂度为 $O(n^3)$ 。期望得分 15。
- 算法 2: 当 R 数量不超过 20 时, 转向的次数不会超过 40, 于是我们对连续的 L 段统一处理, 模拟可以解决。结合算法 1, 期望得分 30。
- 算法 3: 从区域 x 出发时, 最终只会停止在 x 左侧或右侧离 x 最近的 O 处。如何判断最终是停在左侧还是右侧? 只需比较 x 左侧的 R 的个数与右侧 L 的个数。在不存在连续的 L 和连续的 R 的条件下, 容易发现答案的规律。结合算法 1,2, 期望得分 45。

T2 加训 (train)

- 算法 4: 答案只与初始点, 最终点, 以及拐点 (移动过程中改变方向的点) 有关。其中拐点只会出现在左侧的 R 和右侧的 L 中, 并且两侧的拐点数量相差不会超过 1。可以通过二分快速找到会产生贡献的拐点, 由于两侧拐点个数的比较有单调性, 所以也可以使用队列维护。期望得分 100。

1 吐槽环节

2 T1 数字 (number)

3 T2 加训 (train)

4 T3 排序 (sort)

5 T4 改造 (modification)

6 题目来源

T3 排序 (sort)

- 可以感受到每一次变换后，序列更加的有序，当某次变换后序列不再发生变化，序列永远不在改变，现在我们来分析变换过程。
- 对于正在比较的两个数 a, b ，若 $a < b$ ，则在 a 之后的一连串的比 a 小数字都会顺势跟在 a 后面，我们基于此将整个序列分为若干个块，其中每一个块包含 a 与它身后的那些比它小的数字。
- 我们用每个块的块的首个元素来代表整个块，若只考虑这些元素，在进行一次变换后，序列有序。

T3 排序 (sort)

- 进一步分析，我们可以得到如下结论：
 - 称 $1 \dots n/2$ 为左侧，其他为右侧。如果一个块在某一次变换后完全位于右侧，这个块将永远位于右侧。
 - 如果所有的块在变换后都不跨越左右，变换后序列不变。反之，会仅仅出现一个块同时出现在左右，这个块的左手部分将形成一个新块，右手部分将再划分，重新拆分为若干个块。
 - 在上一条结论的基础上，每一次变换都会新增加至少 1 个块，最终序列至多有 n 个块，所以变换操作最多进行 n 次就会停止。因此询问中 $t > n$ 与 $t = n$ 等价。
- 预处理每一个数为开头形成块的长度，用权值线段树等方式直接维护块即可。

1 吐槽环节

2 T1 数字 (number)

3 T2 加训 (train)

4 T3 排序 (sort)

5 T4 改造 (modification)

6 题目来源

T4 改造 (modification)

- 算法 1: 对于每一个 k 暴力搜索在哪一些位置分段, 然后再暴力枚举如何重排这些子段。时间复杂度 $O(n!poly(n))$, 期望得分 15。
- 算法 2: 我们依旧暴力枚举在哪一些位置分段, 在重排的时候, 加入一点点贪心思想。对于每一段, 它的贡献只有 3 种情况, 全由 0 贡献/全由 1 贡献/由最长不下降子序列长度贡献, 其中只有一段能够贡献第三种, 将三种贡献处理出来, 简单贪心即可。时间复杂度 $O(2^n n)$, 期望得分 30。

T4 改造 (modification)

- 算法 3: 基于算法 2 的思想, 我们提前处理出每一个区间的三种贡献, 进行 dp。设 $f_{i,j,0/1}$ 表示将前 i 个数划分为 j 段, 之后可以/不可以有第三种贡献, 重排后的最长不下降子序列。时间复杂度 $O(n^3)$, 期望得分 45。
- 算法 4: 优化算法 3 中的 dp 状态。在 dp 时我们并不只记录子段划分出的状态, 而在每一个位置记录。具体来说, 定义 $f_{i,j,0/1,0/1}$ 表示将前 i 个划分为 j 段, 之后可以/不可以有第三种贡献, 下一个期望选 0/1, 重排后的最长不下降子序列。转移如下:

$$\begin{cases} f_{i+1,j,k,l} = \max(f_{i,j,k,l} + [s_i == l]) \\ f_{i+1,j,1,1} = \max(f_{i,j,0,0} + [s_i == 1]) \\ f_{i+1,j+1,k,0} = \max(f_{i,j,k,l} + [s_i == 0]) \\ f_{i+1,j+1,k,1} = \max(f_{i,j,k,l} + [s_i == 1]) \end{cases}$$

时间复杂度 $O(n^2)$, 期望得分 65。

T4 改造 (modification)

- 算法 5: 如果存在一段连续的相同的值, 将它们放在同一个子段中是优秀的。把连续相同段先合并起来。这样我们转化为一个 01 相间的序列, 只不过每个元素实际都代表了一段长度。设转换后共有 m 个元素。我们来处理这个看着非常麻烦的第三种贡献 (算法 2 中提到)。只考虑最后会产生贡献的元素, 是若干段连续的 01 交错排列, 那么中间一定会出现一段 $000\dots111$ ($k > 2$ 时)。这引导我们将问题转化为如下形式:
选出一个尽量长的子序列, 使得其连续段数不超过 $k + 1$, 这子序列的长度就是划分 k 段时的答案。

T4 改造 (modification)

- 算法 5: 接上。倒过来考虑,我们要做的其实是在其中删去一些元素,让未被删去的元素来产生贡献。首先一定不会删除连续的两个(即相邻的一组 0 和一组 1),否则保留其中的一组一定会更优。当我们将中间的一个元素删除时,它两侧的两个元素会合并成一个新的元素(因为 01 间隔),这样我们进一步完成转化。

对于每个 x , 选择一些价值之和为 x 的元素(准备将它们删除),要求它们两两不相邻,且长度之和最小。其中靠边的元素价值为 1,中间的元素价值为 2。

这是一个经典的反悔贪心问题,可以用堆来维护,在实现时可以先讨论靠边的元素是否删除,时间复杂度 $O(n \log n)$,期望得分 100。

T4 改造 (modification)

- 算法 6: 运用算法 5 中 $k + 1$ 个连续段的结论。猜想答案关于连续段个数是上凸的。分治加闵可夫斯基和优化 dp。时间复杂度 $O(n \log n)$, 期望得分 100。

1 吐槽环节

2 T1 数字 (number)

3 T2 加训 (train)

4 T3 排序 (sort)

5 T4 改造 (modification)

6 题目来源

题目来源

- T1: <https://codeforces.com/gym/105257/problem/G>
- T2: <https://codeforces.com/problemset/problem/733/E>
- T3: <https://loj.ac/p/3813>
- T4: <https://qoj.ac/contest/1031/problem/5013>（好题，也推荐大家去看官方题解）

Thanks!