

NOIP2017模拟题day1题解

noname

2017 年 9 月 29 日

1 Lucky Transformation

1.1 solution 0

我们可以有梦想! 直接输出原序列。

时间复杂度 $O(1)$, 期望得分10分。

1.2 solution 1

我们可以打暴力! 直接模拟每次操作。

时间复杂度 $O(nk)$, 期望得分50分。

1.3 solution 2

很容易发现, 我们枚举到一个 x 后, 由于只会改变当前位置, 不需要再枚举比它小1以上的 x , 那么扫一遍即可。

时间复杂度 $O(n + k)$, 期望得分70分。

如果没有考虑改变第 x 位后新的 x 可能减一, 那么仍可以通过测试点6。

1.4 solution 3

考虑遇到一个首位为奇数的233或者223, *Tgopknight*会使其在这二者之间互相转变, 那么只需要考虑剩余的操作数的奇偶性即可。

时间复杂度 $O(n)$, 期望得分100分。

2 Snake vs Block

2.1 solution 0

我们可以有梦想! 直接输出0。

时间复杂度 $O(1)$, 期望得分5分。

2.2 solution 1

我们可以打暴力! 枚举每一行在哪个区间内移动, 从哪个位置离开。

时间复杂度不超过 $O(35^n)$ ($35 = 1 * 5 + 2 * 4 + 3 * 3 + 4 * 2 + 5 * 1$), 期望得分50 分。

2.3 solution 2

对于保证所有砖块所在的行数都比豆豆所在的行数大的数据, 可以考虑在第一阶段吃掉所有豆豆 (假设吃完后蛇的长度为 st), 然后 dp 砖块的部分。

令 $f[i][j][k]$ 表示前 i 行, 蛇的长度还剩下 j , 从第 k 列离开第 i 行的可行性。

令 $g[j_1][l][r]$ 表示蛇的长度还剩下 j_1 , 当前行在第 l 列到第 r 列之间移动后仍然未死亡的可行性。

$f[0][st][3] = true$ 为初始状态。

对于每个 i , 首先令 $g[j_1][k][k] = f[i-1][j_1 - a[i][k]][k]$ 为初始状态。

状态转移方程为 $g[j_1][l][r] = g[j_1 - a[i][l]][l+1][r] \parallel g[j_1 - a[i][r]][l][r-1]$

最后 $f[i][j][k]$ 为 $g[j][l][r]$ ($1 \leq l \leq k \leq r \leq 5$) 的或。

找到一个最小的 j 使得存在 i, k , $f[i][j][k] = true$, 则 $st - j$ 即为答案。

时间复杂度 $O(n * (5 * \max(a_{i,j}) * n) * 35)$, 期望得分20 分。结合solution 1期望得分50分。

2.4 solution 3

把豆豆和砖块丢在一起 dp 即可。

令 $f[i][j][k]$ 表示前 i 行, 蛇的长度还剩下 j , 从第 k 列离开第 i 行的最大得分。

令 $g[j_1][l][r]$ 表示蛇的长度还剩下 j_1 , 当前行在第 l 列到第 r 列之间移动后仍然未死亡的最大得分。

$f[0][4][3] = 0$ 为初始状态。

对于每个 i , 首先令 $g[j_1][k][k] = f[i-1][j_1 - a[i][k]][k] + \max(-a[i][k], 0)$ 为初始状态。

状态转移方程为 $g[j_1][l][r] = \max(g[j_1 - a[i][l]][l+1][r] + \max(-a[i][l], 0), g[j_1 - a[i][r]][l][r-1] + \max(-a[i][r], 0))$

最后 $f[i][j][k] = \max\{g[j][l][r] | (1 \leq l \leq k \leq r \leq 5)\}$ 。

$ans = \max\{f[i][j][k]\}$

具体做法参考 std 。

时间复杂度 $O(n * (5 * \max(a_{i,j}) * n) * 35)$, 期望得分100 分。

P.S. $m = 0$ 的数据其实出题人没有想到做法只是看看你们有没有啥做法emmm

P.P.S.题目出自手游Snake vs Block, 游戏规则有所改动

3 Ping

3.1 solution 0

我们可以有梦想! 直接输出0。
时间复杂度 $O(1)$, 期望得分5分。

3.2 solution 1

我们可以打暴力! 枚举哪些站点损坏。
时间复杂度 $O(2^{nk})$, 期望得分30分。

3.3 solution 2

对于一条链的情况, 考虑贪心。
此时站点相当于数轴上的一些线段。
每次找到未被切断的线段中右端点最靠左的, 令其右端点为损坏站点。用单调栈维护即可。
时间复杂度 $O(k)$, 期望得分20分, 结合 *solution 1* 期望得分50分。
P.S.证明可自行搜索, 这是经典题。

3.4 solution 3

对于 $k \leq 3$ 的情况, 考虑分类讨论。
 $k = 1$ 时任意删掉路径上的一个站点即可。
 $k = 2$ 时, 若两条路径相交, 则删掉相交部分的任意一个站点即可, 否则在两条路径上分别删掉一个站点。
 $k = 3$ 类似, 这里不再赘述。
时间复杂度 $O(\log n)$, 期望得分10分, 结合 *solution 1, 2* 期望得分60分。

3.5 solution 4

考虑将 *solution 2* 扩展到树上。
任意将一个站点设置为树的根, 首先求出 *DFS* 序, 倒序处理, 对于每个站点, 如果有以其为 *LCA* 的且未被切断的路径, 则将这个站点标记为损坏。用树链剖分维护即可。
具体做法参考 *std*。
时间复杂度 $O(k \log^2 n)$, 期望得分100分。
证明如下:
考虑树的后序遍历的序列, 我们需要从中选取一些点, 使得一些集合被覆盖。如果是一维情况, 这是覆盖一些区间。像一维情况一样, 我们将这些集合按照最后一个点排序。考虑排序后第一个集合, 我们一定要在这个集合上选择一个点。最优的情况是这个点能够覆盖之后的一些集合。
如果第一个集合是 a_1, a_2, \dots, a_n , 其中 a_n 为其 *LCA*

若对于任意 a_i ，它正好也是之后第 k 个集合的元素，由于我们是在后序遍历上按最后一个点排序的，那么第 k 个集合的 LCA 一定是 a_n 的祖先，则 a_n 也一定在第 k 个集合。

令包含 a_i 的集合的集合为 A_i ，一定有 $A_1 \cup A_2 \cup A_3 \cup \dots \cup A_n \subseteq A_n$

故删去 a_n ，即路径的 LCA ，一定最优。