Solutions to NOIP Simulation

NickH

2018 年 9 月 30 日

1 calculator

1.1 算法1

暴力枚举下一步是加一还是减一，然后验证解的合法性。

复杂度$O(2^K)$，期望得分30分。

1.2 算法2

首先我们发现S, T完全可以互换，所以不妨假设$S \le T$。使用动态规划，用$f_{i,j}$表示进行了i次操作之后计算器上的数字是j的方案数，转移非常简单，只需要枚举下一次操作是加一还是减一即可。

复杂度$O((T+K)K)$，期望得分60分。

1.3 算法3

这个题目需要一些数学技巧来进行分析。试想我们有一个坐标系，点A的坐标是$(0,0)$，点B的坐标是$(i, T-S)(T \ge S)$，我们要从A走到B，走的规则是：如果我们当前在(x,y) 处，下一步我们可以走到$(x+1, y+1)$或者$(x+1, y-1)$，同时允许走到直线$y = -S$以下的区域（不包括这条直线）。那么我们发现从A走到B的方案数恰好是使用i次操作将S合法地变成T的方案数。

现在我们来考虑这个新的问题，我们发现我们每次走的都是45度角的斜线，不能穿过的区域的边界是一条横着的直线，这个问题似乎有点无法处理。我们可以把坐标系旋转45度，让$x+y=0$这条直线成为新的x 轴，$x=y$这条直线成为新的y轴。那么现在我们的走法要么横着往右走一步，要么竖着往上走一步，禁区线是一条形如$x-y=S$的直线。

这是一个标准的格路问题：从$(0,0)$走到第一象限内某点，只能向右走一步或者向上走一步，不允许穿过直线$x-y=S$，求方案数。这个问题是一个经典的组合问题，解法非常巧妙：不妨假设$S=0$，并且我们要算的是从$(0,0)$到某个点(x,y)的方案数。如果没有不许穿过直线$x-y=0$的限制，那么答案显然是$\binom{x+y}{x}$；在有了这个限制以后，我们首先把问题转化一下，变成不允许触碰（不是穿过）$x-y=-1$这条线。$(0,0)$关于$x-y=-1$的对称点是$(1,-1)$，注意到一旦某个方案触碰

了这条线，我们可以等价地找到一条从$(1,-1)$走到(x,y)的路径，于是这个问题的答案就是$(0,0)$无限制走到(x,y)的方案数减去$(1,-1)$无限制走到(x,y)的方案数。

预处理之后复杂度$O(K)$，期望得分100分。

2 work

一个非常简单的动态规划题。我们发现如果用f_i表示考虑前i台机器，要选择第i台机器的最大效率，似乎有些不太好转移。于是我们令f_i为不选择第i台机器的最大效率，并且在最前面和最后面都加上一台效率为0的机器，分别标号为0号机和$n+1$号机，那么答案就是f_{n+1}。

f的转移也非常简单，只需要枚举前一台关闭的机器就行：

$$f_i = \max_{i-k-1 \le j \le i-1} \{f_j + e_{j+1} + e_{j+2} + \cdots + e_{i-1}\}$$

这个dp的暴力实现是$O(nk)$，但是这是一个单调队列的标准模板题，可以用其优化到$O(n)$。

3 cubes

3.1 算法1

可以用链表暴力模拟全过程，复杂度是$O(m^2)$，期望得分40分。

3.2 算法2

这里只涉及到柱子的合并，所以可以考虑使用并查集。

在并查集当中每个积木需要至少两个信息：这个积木所在的柱子，和这个积木下方的积木数量。我们可以尝试将一个柱子的代表元设为最上面那个积木或者最下面那个积木，发现前一种方法不可行，所以我们把每个柱子的代表元设为最下面的柱子。

如果要合并两个柱子（假设两个柱子的代表元分别是x和y），那么x下方的积木本来是0个，现在需要进行更新。此时x上方的积木可能会出现信息错误，不过没有关系，对于x上方的任何一个积木，在询问这个积木的时候，通过并查集的一路查找，一定在某一步会找到x，此时再进行更新也不迟。具体实现可以参见std。

此时的期望复杂度是$O(m\log m)$，期望得分100分。

4 总结

本套题难度较低，除了第一题有一些技巧性之外，剩下的两个题目都是比较常规的基础题，只要基础牢固就不难拿到满分。同时题目当中还有一些部分分可以使用暴力算法得到，所以本套试题的期望得分比较高。如果同学们发现自己的成绩不太理想，除了要认真思考自己在考试当中的失误之外，还要反省自己的基础是否已经足够牢固，不够的地方及时改进。