

Dynamic Programming Review

Dez

2017 年 12 月 9 日

1 消防局的设立

最后用贪心做了这题。把所有的点按深度放入堆中，每次选出一个深度最大的点，让它的父亲的父亲被选中，这样应该是最优的。然而实际上实现感觉还是有问题的，数据太水了，没有出错。

2 小 a 和 uim 之大逃离

设 $f[i][j][k][l]$ 为在位置 (i, j) 小 a 比 uim 多吸收 k 点，当前格子由 $l[0: \text{小 a}, 1: \text{uim}]$ 吸收的方案数。把每个格子 (i, j) 的 $f[i][j][a[i][j]][0]$ 初始化为 1, 其中 $a[i][j]$ 表示这个格子的点数。

3 有线电视网

树形动归。设 $f[i][j]$ 为以点 i 为根的子树中选取 j 个用户所能获得的最大利益。那么这就成了一个树形的背包问题，只要记住 j 的枚举要从大到小，还有叶子节点要特判即可。

4 垃圾陷阱

非常奇怪。可以枚举的变量太多了，反而不好下手。设 $f[i]$ 为到高度 i 时所能存活的最大时间。那么每碰到一个垃圾，都有两种选择，要么吃，要么踩。并且踩和吃的前提都是你当前状态的存活时间能到达它的下落时间。所以我们先把垃圾按时间从小到大排序。然后枚举高度， $f[i] = f[i] + r.t, f[i+r.h] =$

$\max(f[i + r.h], f[i])$ 。当 $i + r.h \geq$ 要求的高度时，答案即为 r 的下落时间。如果不能到达指定高度，那么输出 $f[0]$ ，也就是把垃圾全都吃掉。注意初始化 $f[0] = 10$ 。