JOISC2024 滑雪 2。

题意：给定 n 个点，第 i 个点初始海拔为 h_i，代价为 c_i。你可以花 k 的代价让 $h_i := h_i + 1$。接下来，你需要连 $n-1$ 条边，形成一棵根向生成树，满足 $\forall u \neq rt$，$h_u > h_{fa_u}$，花费 $\sum_u \max(d^-(u)-1, 0) \times c_u$ 的代价。求最小代价。

$1 \leq n \leq 300$，$0 \leq h_i \leq 10^9$，$1 \leq c_i \leq 10^9$。

考虑找一些很强的特例，得到一些特殊情况下的观察，再尝试推广。下面给出一些可能的弱化方向：

- $k=+\infty$；
- $h_1\le h_i$，$c_1\le c_i$；
- 所有输入的变量大小不超过 40；
- 所有输入的变量大小不超过 300。

考虑 $k=+\infty$ 的情况。此时应该先最小化提高海拔的操作次数，再最小化生成树的代价。此时把海拔最低的点中代价最小的点定为根最优。对于海拔与根相同的点，至少要提高它们的海拔一次。接下来考虑根是唯一一个海拔最低的点的情况。

考虑用这样的方式直观表示根向生成树的结构：把点画在二维平面上，海拔相等的点画在同一层，则可以先以 k 的代价把任意一个点向上推一层，然后每个点要向它下方的一个点连边。

可以得到一个扫描线贪心，按照海拔从低到高的顺序考虑问题，因为对于每个点，它的代价关于入度的函数是下凸的，所以可以直接用小根堆维护斜率进行贪心。时间复杂度 $O(n\log n)$。

考虑 $h_1\le h_i$，$c_1\le c_i$ 的情况，显然此时以 1 为根最优。同样先转化成根是唯一一个海拔最低的点的情况。观察一下固定每个点的海拔时贪心的过程。初始时有一片叶子，设当前有 $leaf$ 个叶子，每加入一层，设该层有 x 个点，当 $x\le leaf$ 时，不用付出额外代价，$leaf$ 不变；当 $x>leaf$ 时，需付出 $(x-leaf)\times c_1$ 的代价，$leaf:=x$。容易发现总代价等于最终的 $(leaf-1)\times c_1$。

所以可以考虑枚举 $leaf$，对于每一层，如果这一层的点数超过 $leaf$ 就提升到下一层。时间复杂度 $O(n^2)$。

考虑所有输入的变量不超过 40 的情况。先枚举根节点，并转化成根是唯一一个海拔最低的点的情况。现在乘的数不一定等于 c_1 了，假设现在在考虑海拔为 i 的这一层，则乘的数应等于 $h<i$ 的点的最小代价，因为显然这是下界，并且如果这个点被推到了第 i 层则可以调整，考虑这个点被推上来的过程中经过的最后一个没有被推上来的点，改成把这个点推上来。显然这样的点一定是存在的，因为把整层的点一起推上来一定不优。

容易想到 DP，以海拔作为阶段，按照后效性分类，状态里只需记录 $leaf$ 和向上推的点数 up。转移时枚举向上推多少个点，单次 DP 的复杂度 $O(n^3V)$，总复杂度 $O(n^4V)$。

考虑所有输入的变量不超过 300 的情况。我们的目标是降低多项式次数。感觉上状态数不太能继续降了，能不能确定哪个点是根呢？

实际上是可以的，取所有海拔最低的点中代价最小的点为根即可。证明可以考虑调整法。

然后还有一个剪枝是每一层如果点数大于等于 $leaf$，但是留在这一层的点数小于 $leaf$ 一定亏了。所以可以卡紧枚举上界。卡紧枚举上界后可以发现转移的代价变成了一次函数，所以可以类似完全背包不枚举物品个数，进行整体 DP。

这样时间复杂度就优化到了 $O(n^2V)$。

考虑一般情况。可以发现有用的层中剩余的点数一定至少减少 1，可以跳过无用层。这样复杂度就和值域无关了。时间复杂度 $O(n^3)$。