

题解

Line

注意到区间与区间之间是不会有交集的

因为要求一段区间夹着的所有数都比两边的小，又要求最后身高尽可能大，所以只要把 $[l+1, r-1]$ 所有数-1。

Line

注意到区间与区间之间是不会有交集的

因为要求一段区间夹着的所有数都比两边的小，又要求最后身高尽可能大，所以只要把 $[l+1, r-1]$ 所有数-1。

当两个区间无交集的时候，对于这两个区间分别这样做显然是不会互相影响的。可以保证正确性。

所以问题就是一开始所有元素全是 $\max H$ ，每次对于一个区间全部减一，最后要输出所有数。

Line

注意到区间与区间之间是不会有交集的

因为要求一段区间夹着的所有数都比两边的小，又要求最后身高尽可能大，所以只要把 $[l+1, r-1]$ 所有数-1。

当两个区间无交集的时候，对于这两个区间分别这样做显然是不会互相影响的。可以保证正确性。

所以问题就是一开始所有元素全是 $\max H$ ，每次对于一个区间全部减一，最后要输出所有数。

差分序列 + 前缀和。

Node

对于 60 % 的数据，暴力，对每个点 i 遍历子树中的所有点 j ，
若 i 到 j 距离为 v_j 则将 i 的关联点个数加上 1

时间复杂度 $O(n^2)$ ，可得 60 分

Node

如果 j 是 i 的关联点，那么 i 一定是 j 往父节点方向走 v_j 步到的点，显然对每个 j ，这样的 i 是唯一的

于是对于每个 j ，找出它往上走 v_j 步到的点 i ，并根据 j 在哪个子树中，将 i 的左(右)关联点数加上 1

使用倍增法，记 $f(i, k)$ 为从 i 往上走 2^k 步到达的点，则 $f(i, k) = f(f(i, k - 1), k - 1)$

Node

DFS 遍历树，实际上就是用了—个栈：进入某个子树时，将根入栈，回溯时，将根出栈.

于是进入点 i 时，栈中的元素正好就是从 1 到 i 的路径上的所有点

因此可以这么做：DFS 遍历树时维护一个栈，进入某节点时将其入栈，回溯时将其出栈，于是可以在 $O(1)$ 时间内找出每个点 i 往上走 v_i 步到的点

复杂度 $O(n)$ ，可得 100 分.

Box

- 如果不熟悉套路的话，很容易想一些奇怪的贪心解
- 看到这种答案符合二分性质的，先思考一下二分答案以后怎么做

Box

- 二分箱子容量后，怎么判断是否合法呢？
- 如果一个时刻，两个箱子都没有溢出，拿哪个箱子里的蛋糕是很难判断的。
- 于是不妨把这个判断留到以后，把扩容的机会先缓存一下，当一个箱子溢出的时候，再消耗缓存中的机会。
- 判断复杂度是 $O(n)$ 的，二分复杂度 $O(\log n)$

Lucky

- 对于 30% 的数据，路径可以通过 bfs 获得
- 怎么判断一个数列能否组成四边形呢？
- 排序，考虑相邻的四个数是否满足 $x_1 + x_2 + x_3 > x_4$
- 复杂度 $O(qn \log n)$

Lucky

- 对于 100% 的数据，需要发现一个性质
- 考虑构造一个最小的不能构成四边形的数列
- 1, 1, 1, 3, 5, 9, 17...
- 三项斐波那契数列！

Lucky

- 这个数列的增长是指数级的。
- 所以一条路径足够长的话，一定是幸运路径。
- 路径长度大于 50，则一定是幸运路径，怎么证明？
- 于是只要判断一下路径长度如果大于 50，直接返回 Y
- 暴力找路径？
- 复杂度 $O(q)$