Solution of 三角形

题目简介

Snuke 有一棵 n $(n \le 2 \times 10^5)$ 个点的有根树，每个点有权值 w_i，初始每个结点上都没有石子。

Snuke 准备了一些石子，并把它们拿在手中。她可以进行以下两种操作任意多次：

1. 从手中取 w_i 个石子放在结点 i 上，进行该操作要求结点 i 的所有孩子 j 上都有 w_j 个石子。
2. 将结点 i 上的所有石子收回手中。

Takahashi 想知道对于每个 i，为了在结点 i 上放 w_i 个石子，Snuke 至少需要准备多少石子。

算法 1

对于子任务 1 ，按题意模拟，状压dp。

时间复杂度 $O\left(2^n \times n\right)$ 或 $O\left(2^n \times n^2\right)$
可以通过子任务 1 ，期望得分 9 分。

算法 2

对于子任务 5 ，$n \le 2000$ 且所有结点的度数 ≤ 2 。

暴力dp记录根下的两条链的状态。

时间复杂度 $O\left(n^2\right)$
可以通过子任务 5 ，获得 5 分。结合算法 1 可获得 14 分。

算法 3

对于子任务 6 ， 除了根结点外所有结点的度数 ≤ 2 。

问题转化，有若干个队列，初始 $tot = 0$，每次选择一个祖先（前驱）已经被选择的元素，加入到 tot 里，直到所有元素都被选择，最小化 tot 的历史最大值。

定义二元组 (sum , max) 中， sum 为所有元素的权值和， max 为最大前缀和。

不难发现这样的二元组易于合并且有结合律。

定义二元组 (sum , max) 的优先级。

- sum 非正的比 sum 为正的优先级高。
- sum 都非正的时候，max 越小的优先级越高。
- sum 都为正的时候，$max - sum$ 越大的优先级越高。

每次选择全局优先级最大的元素，如果它的祖先（前驱）已经被选择就立刻选择，否则将其和前一个元素合并。

【关于这个贪心的正确性的证明见 `附录一` 。】

根结点的答案跑一遍如上贪心，非根结点直接模拟。

时间复杂度 $O\left(n\log n\right)$
可以通过子任务 $5,6$ ，获得 18 分。结合算法 1 可获得 27 分。

算法 4

对于子任务 3 ，$w_i=w_{i+1}$ 。

当 $w_i\le w_{i+1}$ 的时候，每一个子树内的操作一定是连续的。

【关于这个结论的证明见 `附录二` 。】

求 ans_i 的时候，dp出 i 的所有孩子 j 的答案 ans_j ，
由于 w_j 全相同，贪心按照 ans_j 从大到小处理。

时间复杂度 $O\left(n\log n\right)$
可以通过子任务 3 获得 6 分。结合算法 $1,3$ 可获得 33 分。

算法 5

对于子任务 4 ，$w_i\le w_{i+1}$。

同样运用当 $w_i\le w_{i+1}$ 的时候，每一个子树内的操作一定是连续的。

求 ans_i 的时候，dp出i的所有亲孩子 j 的答案 ans_j，
由于 w_j 全为正整数，贪心按照 ans_j-w_j 从大到小处理。

时间复杂度 $O\left(n\log n\right)$
可以通过子任务 $3,4$ ，获得 18 分。结合算法 $1,3$ 可获得 45 分。

算法 6

对于子任务 2： $n\le 2000$ 。

正难则反，转化题意：

令 $val_i=-w_i+\sum_{j\text{是}i\text{的孩子}}w_j$

初始 $tot=w_1$

依次选择一个父亲已经被选择的点，将 $tot+=val_i$，最小化 tot 的历史最大值。

依旧使用子任务六的贪心，每次选择全局优先级最大的元素，
如果它的父亲已经被选择就选择，否则和其父亲合并。

时间复杂度 $O\left(n^2\log n\right)$
可以通过子任务 $1,2,5$ ，获得 33 分。结合算法 $3,5$ 可获得 64 分。

正解

对于所有数据，无特殊性质。

观察一下算法 6 的性质。

每一棵子树内部的选择顺序是全局的子序列。

不难想到只做一遍整棵树的贪心，用简单数据结构维护一次选择对所有询问的影响。

【具体实现方式见 `附录三` 】

时间复杂度 $O\left(n\log n\right)$ 或 $O\left(n\log^2 n\right)$
可以通过所有子任务，获得 100 分。

附录

附录一

假设二元组 a 的优先级比 b 大，且 a 和 b 在操作排列中相邻，
那么 a 前 b 后一定不劣于 b 前 a 后。

分类讨论可得任意两个二元组合并之后不会比这两个二元组的优先级都高。

所以当前优先级最高的二元组永远都会是优先级最高的。

所以当优先级最高的二元组：

- 祖先（前驱）已经被选时，一定直接被选。
- 祖先（前驱）还没被选时，等到祖先被选的时候它依然是优先级最高的，故祖先被选时它会立刻被选。

附录二

因为这个子树的根的点权是最小的，相当于操作完这棵子树后总和会变少。总和减少的优先级是最大的。

附录三

实现方法一：用算法六求出整棵树的操作排列，对一个结点操作的时候给这个结点到根的路径上的所有询问打上这个结点的标记。

实现方法二：定一个集合的代表元为这个集合深度最小的结点。用算法六做整棵树，将一个后代集合合并到祖先集合的时候，将后代集合的代表元（闭）到祖先集合的代表元（开）的路径上的所有询问打上后代集合的标记。

路径打标记使用树链剖分套线段树或LCT实现。