

Solution

1 T1

先考虑给定二叉树的情况，那么最少步数一定是 n 减去最长链长度。因为每次旋转最多会让最长链增加 1

然后就可以直接 dp 了，统计有多少棵树的最长链长度为 i

直接 dp 是 n^3 的，有两种优化方法。第一种是直接再开一个数组存 dp 的前缀和，另一种是统计最长链 $\leq i$ 的方案数

这样，最长链长度为 1 的就被算了 n 遍，长度为 2 的被算了 $n-1$ 遍，以此类推，可以直接得出答案

2 T2

设 $dp[u][i]$ 表示只考虑 u 的子树，并且只能选 $a \geq i$ 的点，能得到的最大权值和

直接 dp 是 n^2 的，而且还会 MLE

注意到 dp 数组是单调不升的，是一个分段函数，我们可以维护这个函数的所有拐点，比如开一个 set 维护函数的差分数组

合并两棵子树可以直接启发式合并，合并结束之后再插入节点 u ，弹掉前面不优秀的拐点就可以了

时间复杂度 $n \log^2 n$

3 T3

考虑最后所有的树构成的序列，容易发现每次修改影响到的树在这个序列上一定构成一段连续的区间

具体来讲，如果将复制操作也看成一棵树状结构，那么一个在 t 时刻的修改 $[l, r]$ 影响到的区间的左端点是 l 这棵树，右端点是操作树上 r 这个点的最后一个出现时间大于 t 的儿子的子树中，最靠右的那棵树

现在考虑如何处理询问

前面的若干个复制操作将 1 到 u 的链分成了若干段，我们现在考虑这样一种情况：树 b 是由树 a 复制出来的，并且 b 的根是 u 。那么在复制的那个时刻，所有影响到 u 的点一定也会影响到 b 的全树

因此我们可以使用一种类似树形 dp 的方法，将每一段分开考虑，当复制出一棵树时，我们可以处理出这棵树的点权需要额外加上多少。假如复制出的树的根节点是 u ，那么它额外加上的权值就是复制时刻时 u 节点的值

询问时，我们只需要考虑那些落在当前树的询问，即询问点到当前树根的那一条链。在一开始的树上树剖，我们将每个修改 $[l, r]$ 放在对应的节点上，问题转化为统计一条链上，包含 i 的修改个数，可以树套树解决

时间复杂度 $n \log^3 n$