串联-解题报告

简要题意

给定一棵树，每个点有两个权值 a_i, b_i。对于树上的一条简单路径，若这条路径上 b 之和乘上 a 大于等于一个常数 V，那么这条路径被称作一条好的路径。

即：对于一条简单路径，设 $p_1, p_2, \ldots, p_k$ 为路径上的点。这条简单路径是好的，当且仅当 $\min_{i=1}^{k} a_{p_i} \times \sum_{i=1}^{k} b_{p_i} \geq V$。

求所有好的路径中，$\sum b$ 的最小值。

数据范围

Subtask	$n \leq$	特殊性质	分值
1	2×10^3	/	15
2	10^4	/	15
3	2×10^5	存在一个点度数为 $n-1$	15
4	2×10^5	第 i 条边连接 i 和 $i+1$	15
5	5×10^4	/	20
6	2×10^5	/	20

对于所有测试点均满足 $1 \leq V \leq 10^{18}, 1 \leq a_i, b_i \leq 10^9$。

解题过程

算法一

考虑直接枚举这条路径的两个端点 x, y，然后使用倍增的方式求出这条路径上的 $\min a$ 以及 $\sum b$ 即可判断。

时间复杂度 $O(n^2 \log n)$，可以通过子任务 1。

算法二

考虑枚举这条路径的一个起点 x，接着以 x 为根进行一次 DFS，就可以求出任意一个点 y 到 x 路径上的 $\min a$ 以及 $\sum b$ 了。

时间复杂度 $O(n^2)$，可以通过子任务 $1, 2$。

算法三

考虑树是一个菊花图的情况。

不妨设菊花图的根为 r。容易发现一条路径上最多只会有三个点，我们进行分类讨论：

1. 路径上有一个点。

直接枚举是哪个点即可。

2. 路径上有两个点。

   容易发现 r 一定在路径上，再枚举一个叶子即可。

3. 路径上有三个点。

   发现 r 还是会在路径上，然后再接上两个叶子。对于这两个叶子，我们考虑枚举 a 更小的那一个即可求答案。

   记 $(a_1,b_1),(a_2,b_2)$ 为我们要找的那两个点，不妨设 $a_1\le a_2$。

   那么我们枚举 (a_1,b_1)，就知道了 $\min a=\min\{a_1,a_r\}$，即 $b_2\ge \frac{V}{\min a}-b_1-b_r$。

   我们要做的就是对 $a_i\ge \min a, b_i\ge \frac{V}{\min a}-b_1-b_r$ 的 i 求出 $\min b_i$。

   具体地，我们按照 a_i 从大到小加点，那么每次要加入的点的 a 在所有叶子中就是最小的。用 set 维护已经加入的点的 b。那么每次只需要在 set 上二分找到 $\frac{V}{\min a}-b_1-b_r$ 的后继即可。

时间复杂度 $O(n\log n)$，结合算法二可以通过子任务 $1,2,3$。

算法四

考虑树是一条链的情况。

我们沿用算法三的思路，考虑枚举 $\min a$ 是哪个点。然后可以使用启发式合并解决。

具体地，我们按 a_i 从大到小加点，然后合并其左右的两个区间。合并时，考虑所有跨过这个点 p 的路径，即对于左右两个区间，一个端点在左边，一个端点在右边。我们在区间长度较小的那个区间里枚举端点，由于 a_p 是 $\min a$，我们可以知道 $\sum b$ 的范围，且由于 b 非负，在长度较长的那个区间里进行二分即可。

复杂度由启发式合并可以得知是 $O(n\log^2 n)$ 的，结合算法二三可以通过子任务 $1,2,3,4$。

算法五

我们可以将算法四中提到的做法倒过来想：

每次找到区间中的最小值，将这个区间按这个最小值的位置劈成左右两个区间。枚举短区间，在长区间上二分。然后递归到两边。

这个做法很像分治，由于原题是一棵树，那么我们可以考虑边分治。那么原树会被划分成两棵子树 T_1,T_2。

考虑对所有跨过分治中心的路径进行求解。那么需要记录每个点到分治中心路径上的的 $(\min a,\sum b)$，简写为 (A,B)。然后我们需要将 T_1 中的一条路径 (A_1,B_1) 和 T_2 中的一条路径 (A_2,B_2) 拼起来，变为 $(\min(A_1,A_2),B_1+B_2)$。

我们考虑枚举 A_1,A_2 中较小的一个，不妨设为 A_1。那么我们需要在满足 $A_2\ge A_1, B_2\ge \frac{V}{A_1}-B_1$ 的 (A_2,B_2) 中找到最小的 B_2。这是一个二维偏序问题，可以使用线段树套 set 解决。

时间复杂度 $O(n\log^3 n)$，结合算法三四可以通过子任务 $1,2,3,4,5$。

算法六

我们继续优化算法五的做法。

瓶颈在于求解二维偏序问题。我们将 (A_1,B_1) 和 (A_2,B_2) 按照 A 分别从大到小进行排序，然后我们从大到小枚举 A_1，那么满足 $A_2\ge A_1$ 的(A_2,B_2) 就是一段前缀，并且随着 A_1 的减小，这个前缀会越来越长。剩下的 $B_2\ge \frac{V}{A_1}-B_1$ 可以直接使用 set 维护得到答案。

时间复杂度 $O(n\log^2 n)$，可以通过本题。

算法七

算法六可以进一步优化，不过感觉没必要。

首先可以把 set 优化掉。我们按照 B_2 从小到大排序，问题就变成了对于一个序列，求一段后缀中第一个没被删掉的数。这个将序列按照 $\log n$ 的大小分块，块间并查集，块内位运算就可以做到线性。

然后排序可以将所有东西离线下来，使用基数排序优化掉。

时间复杂度 $O(n \log n)$。

参考资料

无。