

轻重路径

【故事背景】

JYY 最近学习了一种处理树形结构的高级技巧,叫“轻重路径剖分”。这种技术会将树中的边划分成轻边和重边。相连的重边会形成一些树上相离的路径。“轻重路径剖分”可以使得从树上任意一点走到根,都至多只会经过 $O(\log N)$ 条不同的重路径。

【问题描述】

如果你不了解轻重路径剖分, JYY 在这里简单介绍一下:

对于一棵有根树中的任意一个点 u , 我们用 $\text{size}(u)$ 表示其为根的子树中的点的数量。对于 u 的所有孩子中, 我们选出 $\text{size}(\cdot)$ 值最大的孩子 v , 并将边 $\langle u, v \rangle$ 设置成重边, u 和其他孩子之间的边我们均设置为轻边。

为了简化问题, 这里 JYY 仅考虑一颗 N 个点的有根二叉树。这 N 个点由 1 到 N 编号。并且如果 u 存在两个 $\text{size}(\cdot)$ 值一样的孩子, 则我们默认 u 和其左孩子的连边为重边。

现在 JYY 希望执行额外 Q 次删点操作, 每次 JYY 会随机删掉一个当前二叉树的叶子节点, 而你则需要动态的维护这棵树的轻重路径剖分。

为了方便输出, 你只需要在每次操作后输出所有重边指向的点的权值和即可。

如果删除一个点之后, 存在一个点 u 拥有两个 $\text{size}(\cdot)$ 值一样的孩子, 则我们保持 u 在该操作执行之前的重边划分。

【输入格式】

从文件 *heavy.in* 中读入数据。

输入文件的第一行包含一个整数 N 。

接下来 N 行, 第 i 行包含两个整数 L_i, R_i , 表示编号为 i 的点的左孩子编号和右孩子编号; $L_i = 0$ 表示点 i 没有左孩子, $R_i = 0$ 表示点 i 没有右孩子。

第 $N+2$ 行包含一个整数 Q , 表示 JYY 进行的删点操作。

第 $N+3$ 行包含 Q 个空格分开的正整数, 表示 JYY 删去的叶子的编号。

输入数据保证每次删除操作均删除了一个叶子。

【输出格式】

输出到文件 *heavy.out* 中。

输出 $Q+1$ 行, 每行包含一个整数, 表示在轻重路径剖分中所有重边指向的点的编号的和。其中第一行对应初始的路径剖分, 之后的 Q 行对应进行了相应删点操作之后路径划分。

【样例输入】

```
8
2 3
4 5
0 0
6 7
0 8
0 0
0 0
0 0
7
6 7 8 5 4 2 3
```

【样例输出】

```
20
21
15
7
6
2
3
0
```

【数据规模】

对于 30%的数据满足 $N \leq 1,000$;
对于 50%的数据满足 $N \leq 50,000$;
对于 100%的数据满足 $N \leq 200,000$ 。