Problem H. Heavy-Light Decomposition

Input file:	heavylight.in
Output file:	heavylight.out
Time limit:	3 seconds
Memory limit:	512 megabytes

Heavy-Light decomposition is the way of partitioning a tree to a set of paths such that walking from any vertex to the root of the tree requires $O(\log n)$ changes between paths. In this problem you have to maintain heavy-light decomposition of a binary tree which has some of its leaves removed one after one.

Consider a binary tree T with n vertices numbered from 1 to n. Each vertex can have a left child and a right child. Vertex number 1 is the root of the tree, it is not a child of another vertex. Each other vertex is the child of some vertex. Vertex that has no children is called a *leaf*. For each vertex the size of this vertex is the number of vertices in its subtree. An edge that leads from vertex u to its child v is said to be *heavy* if the size of v is greater than the size of another u's child w or if v is the only child of u. If u has two children and both of them have the same size, initially the heavy edge from u leads to its left child.

First you must find all heavy edges in the tree and print the sum of numbers of vertices they lead to. After that you must process m requests: remove leaf vertex u_i from the tree, update all heavy edges, and print the sum of numbers of vertices that they lead to now. If after removing the vertex from the tree some vertex u has children of equal sizes, heavy edge from u is not changed.

Input

The input file contains several test cases.

The first line of each test case contains n — the number of vertices in a tree ($2 \le n \le 200\,000$). The following n lines contain description of the tree. The i-th of these lines contains two integers: L_i and R_i — numbers of left and right children of the i-th vertex, or 0 if the i-th vertex has no corresponding child.

The following line contains m — the number of leaves to be removed ($1 \le m \le n - 1$). The following line contains m integers: $u_1, u_2, \ldots, u_m$ — the vertices to remove. It is guaranteed that after all previous removals, the vertex u_i is a leaf.

Input is terminated by a line that contains $n = 0$, it must not be processed. The sum of values of n in all tests in one input file doesn't exceed 200 000.

Output

For each test case output $m + 1$ integers. The first of them must be the sum of vertices that heavy edges lead to in the initial tree. The following numbers must be equal to this sum after the corresponding leaves have been deleted.

Examples

heavylight.in	heavylight.out
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 0	20 21 15 7 6 2 3 0