This is the third problem from the recently held Kharagpur regional ICPC. You can find the orginal description at http://acm.iitkgp.ac.in (problem MAFIASYN).

More formally, the input consists of a rooted tree with nodes numbered 1 to n, in which node 1 is the root. Two positive integers, value $v_i(0)$ and capacity c_i are specified for each node $i > 1$. $v_1(0)$ and c_1 may be assumed to be 0. The values change in each unit time step as defined below.

```
for (i = 1; i <= n; i++)
if ((i == 1) || (v_i(t) > c_i))
    v_i(t+1) = v_i(t);
else
    v_i(t+1) = 0;
for (i = 2; i <= n; i++)
if (v_i(t) <= c_i) v_j(t+1) += v_i(t);
\\ where j is the parent of i
```

Informally, if the current value of a node other than 1 is less than or equal to its capacity, at the next step, this value is passed to the parent, that is, subtracted from the value of the node and added to the value of the parent. If it is greater than the capacity, or if it is the root, nothing is done for this node, but the values of some of its children nay be added to its value.

After some number of time steps, the values v_i will not change. The problem is to compute the final value of v_1 which will remain constant.

Input Format

The first line of input specifies the number n.
The next $n - 1$ lines contain the numbers $v_i(0)$ and c_i, for $i = 2$ to n.
The next $n - 1$ lines specify the tree by giving the parent of node i, for $2 \leq i \leq n$.

Output Format

Output the final value on one line without any spaces.

Constraints

$1 \leq n \leq 10^7$.
$1 \leq v_i(0), c_i \leq 10^9$.

Sample Input

```
5
3 6
4 2
4 5
5 6
5
1
5
1
```

Sample Output

```
5
```

Optional (Don't include in submitted file).
Find all nodes i with a non-zero final value of v_i, and the value v_i for these nodes.