Christmas Tree (xmastree)

Each year the Christmas decorations are prepared earlier and earlier. This year is no exception, and William wants to be prepared! William is planning to put a really big Christmas tree in his garden, bigger and brighter than everyone else!

Figure 1: A Christmas tree for sure nicer that William's one.

The tree is built starting from a big pine tree, where many lights are installed on the branches. Starting from the root one or more cables are installed, and when there's a branch a light is placed there and one or more new cables come out of it. When a leaf is reached a new light is placed together with a switch. Pressing the switch will toggle all the lights in the path from that leaf to the root (turning on the switched off lights and vice-versa).

After all this work William turns on all the N lights but realized that his tree is pretty boring. To avoid re-routing all the lights he decides to assign a niceness level A_i to all the N lights (A_i might be negative as well). If a light is turned off its niceness level goes to zero. Now he's questioning what is the maximum level of niceness achievable acting only on the switches.

☞ Among the attachments of this task you may find a template file `xmastree.*` with a sample incomplete implementation.

Input

The first line contains the only integer N. The following $N-1$ lines contain two integers a and b, meaning that the light a is connected with the light b. The next line contains N integers, the values of A_i.

Output

You need to write a single line with an integer: the maximum possible sum of the A_i of the turned on lights.

Constraints

- $2 \leq N \leq 200\,000$.
- $-10^9 \leq A_i \leq 10^9$ for each $i = 0 \ldots N-1$.
- The root light has index 0.
- There cannot be a switch on the root.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.
🟧⬜⬜⬜⬜

– **Subtask 2** (10 points) There is only one switch and $N < 1000$.
🟧⬜⬜⬜⬜

– **Subtask 3** (20 points) There are exactly $N-1$ switches.
🟧⬜⬜⬜⬜

– **Subtask 4** (15 points) $N \leq 20$.
🟧🟧⬜⬜⬜

– **Subtask 5** (20 points) $N \leq 1000$.
🟧🟧🟧⬜⬜

– **Subtask 6** (35 points) No additional limitations.
🟧🟧🟧⬜⬜

Examples

input	output

4
0 3
0 2
1 3
-10 8 3 5
	13

5
0 1
1 3
1 4
0 2
1 2 4 -4 -5
	7

Explanation

In the **first sample case** by pressing the switch on the light 2 the root and the light 2 get tuned off, keeping a total niceness of $5 + 8 = 13$.

In the **second sample case** you can proceed as follows:

- Switch light 3 (that turns off also lights 0 and 1)
- Switch light 4 (that turns on again lights 0 and 1)

In the end the lights 0, 1 and 2 are powered, with a total niceness of $1 + 2 + 4 = 7$.