

Task 2. Miners

There is a vertical mine with N chambers, where each chamber has exactly one vertical tunnel that goes into it (from some other chamber which is closer to the surface). Chamber number one is connected directly to the outside. Formally, the mine is a **tree**, rooted in vertex number one.

Every tunnel has some score associated with it (which depends on a lot of factors, but in this problem we're given these scores directly). Unfortunately, sometimes going through a tunnel is risky, so these scores can also be **negative**.

Currently we have some number of miners in each chamber. We want to create a mining assignment, which should include some of the miners (possibly none), and assign to each of them a vertical path. **A vertical path can only go deeper** - that is if a miner is in some chamber, he can only go to chambers which are further away from the surface. More formally, from a vertex in the tree, we can only go to its children. We define the score of such a path, as the sum of the scores in the tunnels we passed through. Similarly, the score of the assignment is the sum of the scores of the paths in it (if no miner was assigned a path, we consider the score being 0).

However, we also have some additional constraints! We don't want to have "crowded" chambers, that is, for each chamber we have a constraint on the maximal number of miners that can end their path there. The miners who weren't assigned a path will just leave the mine and **won't be counted towards these constraints**.

We're interested in the largest score of some mining assignment. Given the structure of the mine, the initial number of miners in each chamber, and the maximal number of miners that can end in each chamber, you should write a program called **miners.cpp**, which computes this value.

Input

From the first line of the standard input, your program should read a single integer N - the number of chambers (we consider chamber 1 as being connected to the outside). The second line contains s_1, \dots, s_N - the initial number of miners for each chamber. The third line contains e_1, \dots, e_N - the maximal number of miners that can end in each chamber. Finally, the last $N - 1$ lines contain a description of the tunnel: the i^{th} of these lines contains p_{i+1} and w_{i+1} , which means that there is a vertical tunnel from chamber number p_{i+1} to chamber number $i + 1$ with a score of w_{i+1} .

Output

On a single line, your program should print the largest possible score of some mining assignment.

Constraints

$$2 \leq N \leq 5 \times 10^5$$

$$0 \leq s_i, e_i \leq 2000, \text{ for all } 1 \leq i \leq N$$

$$1 \leq p_i < i, \text{ for all } 2 \leq i \leq N$$

$$|w_i| \leq 2000, \text{ for all } 2 \leq i \leq N$$

Subtasks

Subtask	Points	N	Additional constraints
1	6	≤ 8	No additional constraints
2	12	≤ 100	No additional constraints
3	14	≤ 2000	No additional constraints
4	18	$\leq 10^5$	The tree forms a line, that is, for each $2 \leq u \leq N$ its parent $p_u = u - 1$. Also, $s_u = e_u = 1$ for all $1 \leq u \leq N$.
5	4	$\leq 10^5$	The tree forms a line, that is, for each $2 \leq u \leq N$ its parent $p_u = u - 1$.
6	20	$\leq 10^5$	No additional constraints
7	26	$\leq 5 \times 10^5$	No additional constraints

Sample input

Input	Output	Explanation
<pre> 5 5 1 0 0 0 100 1 1 2 4 1 6 1 1 2 2 2 -1 </pre>	32	One possible solution is: <ol style="list-style-type: none"> 1) 1 -> 2 -> 4 with a score of 8 2) 1 -> 2 -> 4 with a score of 8 3) 1 -> 2 with a score of 6 4) 1 -> 2 -> 5 with a score of 5 5) 1 -> 2 -> 5 with a score of 5