2015 SESQUICENTENNIAL REDUX

The family firm of *Widget, Whatsit & Doodah* (est. 1862) had planned to commemorate its sesquicentennial by reproducing classic widgets from the last 150 years. Finely crafted limited editions were prepared and the company was keen to get each one produced as efficiently as possible. They now realise that efficient does not mean fast but are quietly confident that this year will see their actual release.

To produce a widget n different processes are required. These can be carried out in any order — indeed each activity can be temporarily halted and resumed at a later date — but only by workers with specific skills. Processes take a fixed amount of time. If more than one worker has the skills for a specific process, *all* of those workers need to work simultaneously on that process for that period. If two processes have no overlap in their skilled workers, those processes can be done simultaneously.

In these times of financial austerity there have been cutbacks in the number of workers available. For each widget n-1 workers have been assigned, each of whom is able to assist with two different processes. No two workers are skilled in the same two processes. Given any set of fewer than n processes, at least one worker assigned to one of those processes is also assigned to a process that is not in the set.

For example, suppose p_1 and p_3 each take 10 ticks and p_2 and p_4 each take 1 tick, and that A is skilled in p_1 and p_2 , B in p_2 and p_3, and C in p_3 and p_4:

- A could complete p_1 in 10 ticks, then A and B complete p_2 in 1 tick, then B and C complete p_3 in 10 ticks and finally C completes p_4 in 1 tick. 22 ticks in total and each worker carries out their lowest numbered process first.
- A could work on p_1 for 1 tick at the same time that C completes p_4. A and B could now complete p_2 in 1 tick. Now A could finish p_1 in 9 ticks while B and C work on p_3. B and C then continue with p_3 for 1 final tick. 12 ticks in total.
- A completes p_1 in 10 ticks while B and C complete p_3. A and B now complete p_2 in 1 tick while C completes p_4. 11 ticks in total.

Write a program that finds the *fastest* time for completing a widget. The first line of the input will be a single integer n ($2 \leq n \leq 50{,}000$) indicating the number of processes. The next n lines will consist of a single integer t_i ($1 \leq t_i \leq 50{,}000$) indicating the amount of time required to complete the i^{th} process. The next n-1 lines will consist of two integers, with line i indicating the two processes which worker i is skilled in performing. No pairing of processes will be duplicated.

You should output a single integer, the minimum amount of time required to complete all the processes.

SAMPLE INPUT

```
4
10
1
10
1
1 2
2 3
3 4
```

SAMPLE OUTPUT

```
11
```