

BSSPC '21 J6 - Lakshy and Orz Trees

Time limit: 2.0s **Memory limit:** 256M

Lakshy was walking around in the forest when he sees an odd-looking tree which he called the **Orz Tree**. An **Orz Tree** is a tree where each node is adjacent to at most 3 other nodes.

A tree is defined as a connected graph with X nodes and $X - 1$ non-repeated bi-directional edges. There are no cycles in a tree.

Lakshy has a rooted tree that is rooted at node 1 back at his house. However, a node in Lakshy's tree may be adjacent to more than 3 nodes. In addition, he can only remove a node if that node does not disconnect his current graph. To remove node i will cost him v_i kneecaps, and to keep his tree alive while he is pruning it, Lakshy may not remove the root (node 1) from his tree. Can you help Lakshy determine the minimum amount of kneecaps he needs to modify his tree to make it into an **Orz Tree**?

Constraints

$$1 \leq N \leq 500\,000$$

$$1 \leq v_i \leq 10^9$$

$$1 \leq a, b \leq N$$

Input Specification

The first line contains an integer N , the number of nodes in Lakshy's tree.

The second line contains $N - 1$ space-separated integers v_i , the cost to remove the i^{th} node for $2 \leq i \leq N$ from Lakshy's tree.

The next $N - 1$ lines each contain two integers a, b , describing that node a connects to node b in Lakshy's tree.

Output Specification

Output the minimum cost (in kneecaps) for Lakshy to make his tree into an **Orz Tree**.

Sample Input

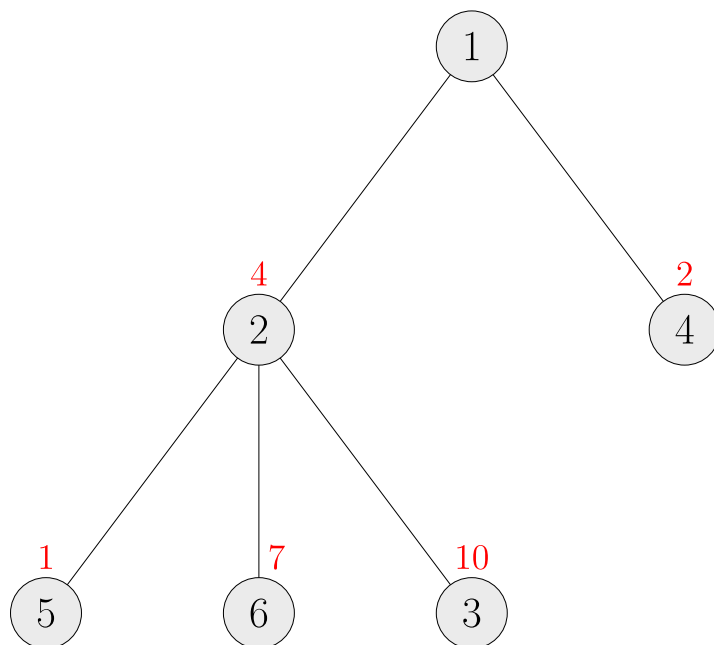
```
6
4 10 2 1 7
1 2
2 3
1 4
2 5
2 6
```

Sample Output

```
1
```

Explanation for Sample Output

Lakshy's tree is depicted below (the cost to remove each node is written in red):



It can be proven that it is optimal to remove node 5 for a total cost of 1.