E. Ehab and a component choosing problem

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You're given a tree consisting of n nodes. Every node u has a weight a_u. You want to choose an integer k $(1 \leq k \leq n)$ and then choose k connected components of nodes that don't overlap (i.e every node is in at most 1 component). Let the set of nodes you chose be s. You want to maximize:

$$\frac{\sum\limits_{u \in s} a_u}{k}$$

In other words, you want to maximize the sum of weights of nodes in s divided by the number of connected components you chose. Also, if there are several solutions, you want to **maximize** k.

Note that adjacent nodes **can** belong to different components. Refer to the third sample.

Input

The first line contains the integer n $(1 \leq n \leq 3 \cdot 10^5)$, the number of nodes in the tree.

The second line contains n space-separated integers a_1, a_2, ..., a_n $(|a_i| \leq 10^9)$, the weights of the nodes.

The next $n - 1$ lines, each contains 2 space-separated integers u and v $(1 \leq u, v \leq n)$ which means there's an edge between u and v.

Output

Print the answer as a **non-reduced** fraction represented by 2 space-separated integers. The fraction itself should be maximized and if there are several possible ways, you should maximize the denominator. See the samples for a better understanding.

Examples

input
3 1 2 3 1 2 1 3

output
6 1

input
1 -2

output
-2 1

input
3 -1 -1 -1 1 2

```
2 3
```

output `Copy`

```
-3 3
```

input `Copy`

```
3
-1 -2 -1
1 2
1 3
```

output `Copy`

```
-2 2
```

Note

A connected component is a set of nodes such that for any node in the set, you can reach all other nodes in the set passing only nodes in the set.

In the first sample, it's optimal to choose the whole tree.

In the second sample, you only have one choice (to choose node 1) because you can't choose 0 components.

In the third sample, notice that we could've, for example, chosen only node 1, or node 1 and node 3, or even the whole tree, and the fraction would've had the same value (-1), but we want to maximize k.

In the fourth sample, we'll just choose nodes 1 and 3.