L-E-R Trees

The team of professor Fabinaris Suchbaum, located in Max Planck Institute for Software Systems in Saarbrücken, Germany, is an internationally acclaimed group of researches who had made many fundamental contributions to numerous fields of computer science, especially to various kinds of binary tree structures. You can read about some of their most famous past projects here and here and here and here and here and here.
This year, the team started experimenting with so called L-E-R trees described below.

Let T be a binary tree in which each node is coloured white or black. Let x be a node in T. We define $w(x)$ and $ww(x)$ to be the number of white nodes in the left subtree and in the right subtree of x, respectively. Analogously, we define $b(x)$ and $bb(x)$ to be the number of black nodes in the left subtree and in the right subtree of x, respectively. We say that x is a principal node if all four values $w(x)$, $ww(x)$, $b(x)$, $bb(x)$ are positive. Furthermore, if x is a principal node then we say that

- x is a L-node if $w(x)/b(x) > ww(x)/bb(x)$,
- x is a E-node if $w(x)/b(x) = ww(x)/bb(x)$,
- x is a R-node if $w(x)/b(x) < ww(x)/bb(x)$.

L-E-R tree is a binary tree each of which nodes is coloured either white or black and which contains at least one principal node.

Before the team conducts any further experiments they have to establish basic properties of selected L-E-R trees.

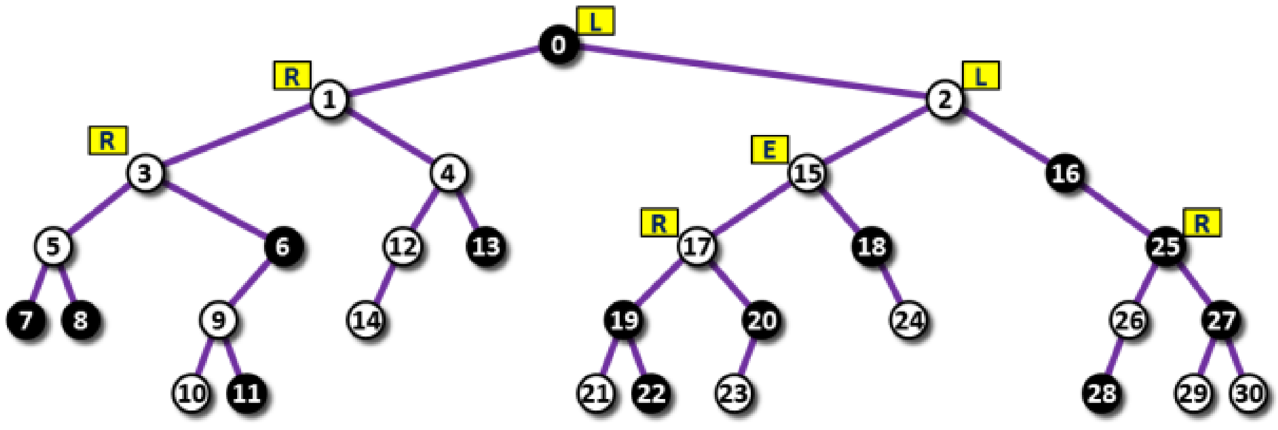

Image 1. An example of a L-E-R tree with 31 nodes and 7 principal nodes. L-nodes, E-nodes and R-nodes are marked by corresponding labels L, E and R. The image illustrates Example 1 below.

The task

You are given a L-E-R tree. Count the number of L-nodes, E-nodes and R-nodes in it.

Input

The first input line contains one integer N representing the number of nodes in the tree. We assume that the nodes are labeled 0, 1, ..., $N-1$ and the root of the tree is labeled by 0.
The next line contains N integers, separated by spaces, which represent the list of colors of particular nodes. The colors are listed in ascending order of node labels. The first integers represents the color of node 0, the second integer represents the color of node 1, etc. White color is coded by 0, black color is coded by 1.
Next, there are $N-1$ text lines, each specifies one edge in the tree. The line contains three integers a, b, c separated by space. Integer a is the label of the parent of the node which label is b. The value of c is either 0 or 1. If $c = 0$ then b is the left child of a, otherwise it is the right child of a. The edges are listed in no particular order.
It holds, $2 \le N \le 150000$.

Implementation note

The depth of a L-E-R tree might reach nearly 40 000. Because of internal stack limitations in some programming languages, recursive approach to the solution should be used with great caution or possibly even omitted completely.

Output

The output contains one text line with three integers L, E, R separated by space and representing (in this order) the number of L-nodes, E-nodes, R-nodes, in the input tree.

Example 1

Input

```
31
1 0 0 0 0 0 1 1 1 0 0 1 0 1 0 0 1 0 1 1 1 0 1 0 0 1 0 1 1 0 0
9 10 0
9 11 1
19 21 0
19 22 1
20 23 0
26 28 0
27 29 0
27 30 1
25 27 1
25 26 0
18 24 1
17 20 1
17 19 0
12 14 0
6 9 0
5 8 1
5 7 0
3 5 0
3 6 1
4 12 0
4 13 1
15 17 0
15 18 1
16 25 1
2 16 1
2 15 0
1 4 1
1 3 0
0 1 0
0 2 1
```

Output

```
2 1 4
```

The tree in Example 1 is depicted in Image 1.

Example 2

Input

```
5
0 0 1 0 1
1 2 0
0 1 0
0 3 1
3 4 1
```

Output

```
0 1 0
```

Example 3

Input

```
15
0 0 0 1 0 0 1 1 0 1 0 1 0 1 0
0 1 0
0 2 1
1 3 0
1 4 1
2 5 0
2 6 1
3 7 0
3 8 1
4 9 0
4 10 1
5 11 0
5 12 1
6 13 0
6 14 1
```

Output

```
1 1 1
```

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the public dataset and the program output to stdout and stderr is available to him/her.
Link to public data set