Dangling paths in a binary tree

Two nodes *x* and *y* in a binary rooted tree are said to be neighbours if *x* is a parent of *y* or *y* is a parent of *x*. The degree of a node is the number of its neighbours. A path of length N in a binary rooted tree is a sequence of nodes (x_1, x_2, ..., x_N), in which x_k is a neighbour of x_{k+1} for each k = 1, 2, ..., N−1. Also, no node can appear in a path more than once.
We say that a path of length N is a dangling path if

- the degree of each node in the path is at most 2,
- x_1 is a leaf,
- the parent of x_N either does not exist or its degree is 3.

The cost of a path is the sum of keys in all its nodes.

� � �

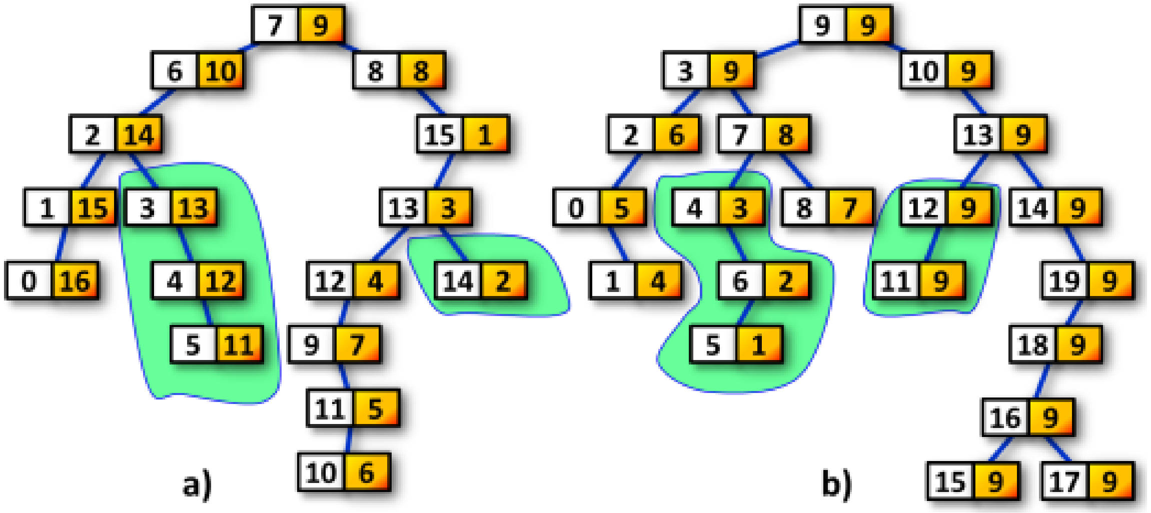

Image 1. Examples of binary rooted trees. The left number in each node is the label of the node, the right number is the key of the node. The dangling paths with the minimum cost and the maximum cost are highlighted. Case a) depicts Example 1 and case b) depicts Example 2 below.

The task

You are given a binary rooted tree. Find the minimum cost and the maximum cost of a dangling path in the tree.

Input

The first input line contains two integers N and R separated by space. The values indicate (in this order) the number of nodes in the tree and the label of the root. The nodes are labeled by integers 0, 1, ..., N−1. The second line contains the keys of the nodes listed in ascending order of node labels. The keys are separated by spaces. Next, there are M text lines, each describe one pair of neighbour nodes. Each line contains the labels of two neighbour nodes separated by space. The order of node labels in a pair and the order of pairs of nodes in the input is arbitrary.
It holds, $1 \leq N \leq 10^6$. The value of all node keys is non-negative and less than or equal to 10^3.

Output

The output contains one text line with two integers separated by space. The first integer represents the minimum cost of a dangling path in the given tree, the second integer represents the maximum cost of a dangling path in the given tree.

Example 1

Input

```
16 7
16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
1 0
2 1
2 3
3 4
4 5
6 2
7 6
7 8
8 15
9 11
11 10
12 9
13 12
13 14
15 13
```

Output

```
2 36
```

The data and solution of Example 1 are depicted in Image 1a).

Example 2

Input

```
20 9
5 4 6 9 3 1 2 8 7 9 9 9 9 9 9 9 9 9 9 9
0 1
2 0
3 2
3 7
7 4
4 6
6 5
7 8
9 3
9 10
10 13
12 11
13 12
13 14
14 19
16 15
16 17
18 16
19 18
```

Output

```
6 18
```

The data and solution of Example 2 are depicted in Image 1b).

Example 3

Input

```
7 3
10 20 30 40 50 60 70
1 0
1 2
3 1
3 5
5 4
5 6
```

Output

```
10 70
```

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