

XOR Tree Path

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

You are given a rooted tree with N vertices, labeled from 1 to N , where vertex 1 is the root. The i -th edge ($1 \leq i \leq N - 1$) connects vertices U_i and V_i .

Each vertex of the tree is painted either white or black. Vertex i ($1 \leq i \leq N$) is painted white if $A_i = 0$, and black if $A_i = 1$.

You can perform the following operation any number of times (possibly zero):

- Choose a **leaf** vertex x , and flip the color (change white vertices to black and black vertices to white) of all vertices on the path from the root to the vertex x (including the root and the vertex x) .

Your goal is to maximize the number of black vertices. What is the maximum number of black vertices that can be achieved?

Input

The input is given from Standard Input in the following format:

```
N
A1 A2 ... AN
U1 V1
U2 V2
⋮
UN-1 VN-1
```

- All values in the input are integers.
- $2 \leq N \leq 10^5$
- $0 \leq A_i \leq 1$ ($1 \leq i \leq N$)
- $1 \leq U_i, V_i \leq N$ ($1 \leq i \leq N - 1$)
- The given graph is a tree.

Output

Output the maximum number of black vertices that can be achieved by performing any number of operations.

Examples

standard input	standard output
5 1 0 0 1 0 1 2 1 3 3 4 3 5	5
6 1 1 0 0 1 0 3 1 2 5 1 2 4 1 2 6	5
9 1 0 1 0 1 0 1 0 1 2 9 1 2 6 9 3 8 4 5 5 9 2 8 7 8	6

Note

In the first example, it is possible to make all vertices black by performing the following operations.

1. choose vertex 2 and perform the operation. This makes vertex 1 white and vertex 2 black.
2. choose vertex 5 and perform the operation. This makes vertex 1,3,5 black.