

About thirty years ago, young Krešo participated for the first time in the national informatics competition. Similar to today, the opening of the competition consisted of a series of speakers who tried to demonstrate the importance of this event to the contestants through motivational messages. The audience, with enthusiasm, began applauding every couple of seconds, but Krešo was irritated by one sentence. Namely, one of the speakers claimed he was more appreciative of the logical operation *AND* than the logical operation *OR* because, regardless of the winner's identity, to him both Mirko **and** Slavko were winners of the national competition, instead of the winner being Mirko **or** Slavko. Krešo went mad, got up and started explaining to the audience that this is an operation known as *exclusive OR* (popular *XOR*). After having given his lecture, he gave the distinguished speaker the next task to verify his understanding.

There is a given tree consisting of N nodes, where each node is assigned a value. The value of a path on that tree is defined as the exclusive *OR* of all nodes' values on that path. Your task is to determine the sum of the values of all paths of the tree, including paths containing only one node.

Thirty years later, Krešo has finally persuaded the authors of the COCI tasks to include this task in one of the rounds. Help us return Krešo's faith in the future of competitive programming.

Note: *Exclusive OR* (\oplus) is a binary operation that is applied separately on each pair of corresponding bits of its two operands so that some bit in the result is set to 1 if and only if that bit in exactly one operand is set to 1.

INPUT

The first line contains a positive integer N ($1 \leq N \leq 100\,000$) that denotes the number of nodes in the tree.

The second line contains N integers v_i ($0 \leq v_i \leq 3\,000\,000$) separated by space, i^{th} value representing the value of the i^{th} node.

The following $(N-1)$ lines contain two numbers a_j and b_j ($1 \leq a_j, b_j \leq N$) that indicate that there is an edge between the nodes a_j and b_j .

OUTPUT

Print the required sum of values for all tree paths.

SCORING

In test cases worth 30% of total points, N will be less than or equal to 200.

In test cases worth 50% of total points, N will be less than or equal to 1000.

In test cases worth additional 20% of total points, each node $x = 1, 2, \dots, N-1$ will be connected to node $x + 1$.

SAMPLE TESTS

input

| input

| input

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

output

10

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

output

64

```
6
5 4 1 3 3 3
3 1
3 5
4 3
4 2
2 6
```

output

85

Clarification of the first sample test:

- The value of the path from node 1 to node 1 is 1
- The value of the path from node 1 to node 2 is $1 \oplus 2 = 3$
- The value of the path from node 1 to node 3 is $1 \oplus 2 \oplus 3 = 0$
- The value of the path from node 2 to node 2 is 2
- The value of the path from node 2 to node 3 is $2 \oplus 3 = 1$
- The value of the path from node 3 to node 3 is 3

The sum of all paths is $1 + 3 + 0 + 2 + 1 + 3 = 10$.