Willie's Subtree

Time limit: 1.2s Java: 4.5s **Memory limit:** 512M

Willie was tired of social distancing so he decided to take up the hobby of tree growing! Recently, he grew a tree with N nodes and $N - 1$ branches, rooted at node 1. As well, this is no ordinary tree but an apple tree, so each node has a certain amount of apples growing on it. This amount is denoted by a_i $(1 \leq i \leq N)$.

Utilizing the unique structure of his apple tree, Willie has created a game for himself. He makes a list of Q questions, each in the form of w_i l_i r_i $(1 \leq i \leq Q)$. This means that he considers all nodes in the subtree of node w_i (w_i included) that are at least distance l_i away, but no greater than distance r_i. The distance between two nodes in the tree is the minimal number of branches it takes to get from one node to the other.

Once Willie has all the nodes that satisfy this condition, he sums up all the apples growing on them, writes down this sum, and moves on to the next question. Note that if no node satisfies the question's conditions, the sum is 0. Having played his game a bunch of times already, Willie realizes that mentally calculating the result of each question might not be the most efficient method (he wants to finish as fast as possible so he can go read manga). Help him code a program!

Input Specification

The first line of the input will contain two integers, N and Q $(1 \leq N, Q \leq 5 \cdot 10^5)$, the number of nodes in Willie's tree and the number of questions on his list.

The second line will contain N space separated integers, a_i $(0 \leq a_i \leq 10^9)$, the number of apples growing on the ith node.

The next $N - 1$ lines will each contain two integers, u_i and v_i $(1 \leq u_i, v_i \leq N)$, indicating that nodes u_i and node v_i are connected with a branch.

The next Q lines will contain three integers, w'_i $(1 \leq w_i \leq N)$, l'_i and r'_i $(0 \leq l_i \leq r_i \leq N - 1)$.

Note that this problem is **online enforced**, meaning that the queries will be encrypted. After reading in w'_i, l'_i, and r'_i, decrypt these values by XORing them with the answer to the previous query. Query 1 will not be encrypted. Since some results may exceed the range of a 32 bit integer, make sure to use 64 bit integers when reading the input of the queries.

Output Specification

Output Q lines, the answers to Willie's questions.

Sample Input 1 (Not Encrypted)

```
7 3
6 5 1 4 3 7 2
2 1
4 2
1 3
2 5
7 3
3 6
1 1 1
5 0 6
7 1 3
```

Sample Output 1

```
6
3
0
```

Sample Input 2 (Encrypted)

```
16 4
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
1 2
2 3
3 4
4 6
6 13
4 7
7 14
3 5
5 8
8 16
5 9
9 15
2 10
10 11
11 12
1 3 5
123 120 122
32 43 43
7 10 10
```

Sample Output 2

```
120
42
11
0
```