Question 1:

Problem Description

You are given a rooted tree with N vertices labelled from 1 to N with 1 as a root. Each vertex is assigned a integer value given by an array A

Subtree sum of any vertex v is summation of all the values of the vertices present in the subtree of including itself

Find two Non-overlapping subtrees such that Bitwise xor of their subtree sum is maximum, return this Maximum xor value.

NOTE:

- Two subtrees are said to non-overlapping if they do not have any vertex in common.

- Return 0 if no such subtrees exists.

Problem Constraints

2<= N = 50000

1A] <= 1000

1<= B[i][0], B[i][1] <= N

Input Format

First argument is an integer array A of size N denoting the values assigned to the vertices.

Second argument is an 2D integer matrix B of size (N-1) x 2 where (B[i][0], B[i][1]) represents an undirected edge between node B[i][0] and B[i][1]

Example Input

Input 1:

A=[1, 5] B=[[1, 2]]

Input 2:

A = [8, 4, 3] B=[[1, 2] [1, 3]]

Example Output

Output 1:
0
Output 2:
7

Question 2:

Problem Description

You are given a tree with A vertics. At each vertex, some number of candies are placed Vertex i contains B[i] candies, given in the input.

You are also given a 2-D array C which denotes that there is an undirected edge between C[i][0] and C[i][1].

You have to choose a starting vertex in the beginning or a root, and you are only permitted to move from vertes to vertex only if an edge connects them. You are not allowed to visit any visited vertex again.

For every possible selection of the of the tree, you write the value of Maximum candies you can obtain - Minimum candies you can obtain using the same root, in your notebook.

Note: While traversing the tree, you can stop at any vertex you want. In other words, it is not necessary to start from the root and end in a leaf.

You have to tell the maximum of all the values written in the notebook

Problem Constraints

1 <= A <= 10^5

1<= B[i]<=10^3

1<=C[i][0], C[i][1] <= A

Input Format

The first argument is the integer A.

The second argument is an integer array B

Example Input

Input 1:

A= 3

B=[3, 1, 1]
C= [[1, 2], [1, 3]]

Input 2:

A= 2

B = [10, 20]
C = [[1, 2]]

Example Output

Output 1:

4

Output 2:

20

Question 3:

Problem Description

You live in Orange town. There are a lot of markets around that are connected with roads. These markets sell oranges at some prices. The town is not very well developed and they still use carts to transport goods from one place to the other. The roads connect two markets together and have one attribute associated with them. The attribute is the price to go from one market to the other in an empty cart. The town also has a tax factor, the tax factor is the number by which the price associated with a road needs to be multiplied so it can go from one market to the other if you a carrying oranges in your cart. So if a road's original price was 5 coins and tax factor of the town was 6 then in an empty cart it would take 5 coins to travel the road but if the cart cointained oranges, it would cost 5 x 6 =30 coins.

You wonder what would be the cheapest way to buy oranges if you were initially at each market. You can either buy at the market you're at or travel to some other market, buy oranges there and travel back to the original market.

You are given an integer A denoting the number of total markets in orange town.
An integer array B denoting the price of purchasing oranges at each market.

A 2-D array C containing the information about the roads where each row contains three values. The first two values denote the market numbers that are bi-directionally connected via a road and the third value is the price

You are also given an integer D, this is the tax factor for the Orange town.
Find and return the required array where each element is the minimum cost to buy oranges at each market such that the starting and ending point is that market.

Problem Constraints

2 < A <= 10^5

B.size() == A

1<B[i] <= 10^7

1 < C.size()<=2*10^5

Output Format

Return an integer array as per the given problem.

Example Input

Input 1:

A =4

B=[3, 1, 10, 1]

C=[[2, 1, 2], [3, 1, 2],[4, 1, 4], [3, 2, 1], [4, 2, 2], [4, 3, 1]]
D=1

Input 2:

A=2

B = [1, 3]

C=[[2, 1, 3]]

D= 1

Example Output

Output 1:

[3, 1, 3, 1]

Output 2:

[1, 3]