Problem Set 4: Disjoint Sets
15-295 Spring 2018

A. Destroying Array

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array consisting of n non-negative integers $a_1, a_2, ..., a_n$.

You are going to destroy integers in the array one by one. Thus, you are given the permutation of integers from 1 to n defining the order elements of the array are destroyed.

After each element is destroyed you have to find out the segment of the array, such that it contains no destroyed elements and the sum of its elements is maximum possible. The sum of elements in the empty segment is considered to be 0.

Input

The first line of the input contains a single integer n $(1 \le n \le 100\ 000)$ — the length of the array.

The second line contains n integers $a_1, a_2, ..., a_n$ $(0 \le a_i \le 10^9)$.

The third line contains a permutation of integers from 1 to n — the order used to destroy elements.

Output

Print n lines. The i-th line should contain a single integer — the maximum possible sum of elements on the segment containing no destroyed elements, after first i operations are performed.

Examples

input

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

output

```
5
4
3
0
```

input

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

output

```
6
5
5
1
0
```

input
8 5 5 4 4 6 6 5 5 5 2 8 7 1 3 4 6

output
18 16 11 8 8 6 6 0

Note

Consider the first sample:

1. Third element is destroyed. Array is now $1\ 3\ *\ 5$. Segment with maximum sum 5 consists of one integer 5.
2. Fourth element is destroyed. Array is now $1\ 3\ *\ *$. Segment with maximum sum 4 consists of two integers $1\ 3$.
3. First element is destroyed. Array is now $*\ 3\ *\ *$. Segment with maximum sum 3 consists of one integer 3.
4. Last element is destroyed. At this moment there are no valid nonempty segments left in this array, so the answer is equal to 0.

B. Minimum spanning tree for each edge

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Connected undirected weighted graph without self-loops and multiple edges is given. Graph contains n vertices and m edges.

For each edge (u, v) find the minimal possible weight of the spanning tree that contains the edge (u, v).

The weight of the spanning tree is the sum of weights of all edges included in spanning tree.

Input

First line contains two integers n and m ($1 \le n \le 2 \cdot 10^5$, $n - 1 \le m \le 2 \cdot 10^5$) — the number of vertices and edges in graph.

Each of the next m lines contains three integers u_i, v_i, w_i ($1 \le u_i, v_i \le n, u_i \ne v_i, 1 \le w_i \le 10^9$) — the endpoints of the i-th edge and its weight.

Output

Print m lines. i-th line should contain the minimal possible weight of the spanning tree that contains i-th edge.

The edges are numbered from 1 to m in order of their appearing in input.

Examples

input

```
5 7
1 2 3
1 3 1
1 4 5
2 3 2
2 5 3
3 4 2
4 5 4
```

output

```
9
8
11
8
8
8
9
```

C. Swaps in Permutation

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a permutation of the numbers $1, 2, ..., n$ and m pairs of positions (a_j, b_j).

At each step you can choose a pair from the given positions and swap the numbers in that positions. What is the lexicographically maximal permutation one can get?

Let p and q be two permutations of the numbers $1, 2, ..., n$. p is lexicographically smaller than the q if a number $1 \le i \le n$ exists, so $p_k = q_k$ for $1 \le k < i$ and $p_i < q_i$.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^6)$ — the length of the permutation p and the number of pairs of positions.

The second line contains n distinct integers p_i $(1 \le p_i \le n)$ — the elements of the permutation p.

Each of the last m lines contains two integers (a_j, b_j) $(1 \le a_j, b_j \le n)$ — the pairs of positions to swap. Note that you are given a `positions`, not the values to swap.

Output

Print the only line with n distinct integers p'_i $(1 \le p'_i \le n)$ — the lexicographically maximal permutation one can get.

Example

input

```
9 6
1 2 3 4 5 6 7 8 9
1 4
4 7
2 5
5 8
3 6
6 9
```

output

```
7 8 9 4 5 6 1 2 3
```

D. Table Compression

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little Petya is now fond of data compression algorithms. He has already studied *gz*, *bz*, *zip* algorithms and many others. Inspired by the new knowledge, Petya is now developing the new compression algorithm which he wants to name *dis*.

Petya decided to compress tables. He is given a table a consisting of n rows and m columns that is filled with positive integers. He wants to build the table a' consisting of positive integers such that the relative order of the elements in each row and each column remains the same. That is, if in some row i of the initial table $a_{i,j} < a_{i,k}$, then in the resulting table $a'_{i,j} < a'_{i,k}$, and if $a_{i,j} = a_{i,k}$ then $a'_{i,j} = a'_{i,k}$. Similarly, if in some column j of the initial table $a_{i,j} < a_{p,j}$ then in compressed table $a'_{i,j} < a'_{p,j}$ and if $a_{i,j} = a_{p,j}$ then $a'_{i,j} = a'_{p,j}$.

Because large values require more space to store them, the maximum value in a' should be as small as possible.

Petya is good in theory, however, he needs your help to implement the algorithm.

Input

The first line of the input contains two integers n and m ($1 \leqslant n, m \text{ and } n \cdot m \leqslant 1\,000\,000$), the number of rows and the number of columns of the table respectively.

Each of the following n rows contain m integers $a_{i,j}$ $(1 \le a_{i,j} \le 10^9)$ that are the values in the table.

Output

Output the compressed table in form of n lines each containing m integers.

If there exist several answers such that the maximum number in the compressed table is minimum possible, you are allowed to output any of them.

Examples

input

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

output

```
1 2
2 3
```

input

```
4 3
20 10 30
50 40 30
50 60 70
90 80 70
```

output

```
2 1 3
5 4 3
5 6 7
9 8 7
```

Note

In the first sample test, despite the fact $a_{1,2} \neq a_{21}$, they are not located in the same row or column so they may become equal after the compression.

E. Envy

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

For a connected undirected weighted graph G, MST (minimum spanning tree) is a subgraph of G that contains all of G's vertices, is a tree, and sum of its edges is minimum possible.

You are given a graph G. If you run a MST algorithm on graph it would give you only one MST and it causes other edges to become jealous. You are given some queries, each query contains a set of edges of graph G, and you should determine whether there is a MST containing all these edges or not.

Input

The first line contains two integers n, m ($2 \le n, m \le 5 \cdot 10^5$, $n - 1 \le m$) — the number of vertices and edges in the graph and the number of queries.

The i-th of the next m lines contains three integers u_i, v_i, w_i ($u_i \neq v_i$, $1 \le w_i \le 5 \cdot 10^5$) — the endpoints and weight of the i-th edge. There can be more than one edges between two vertices. It's guaranteed that the given graph is connected.

The next line contains a single integer q ($1 \le q \le 5 \cdot 10^5$) — the number of queries.

q lines follow, the i-th of them contains the i-th query. It starts with an integer k_i ($1 \le k_i \le n - 1$) — the size of edges subset and continues with k_i distinct space-separated integers from 1 to m — the indices of the edges. It is guaranteed that the sum of k_i for $1 \le i \le q$ does not exceed $5 \cdot 10^5$.

Output

For each query you should print "YES" (without quotes) if there's a MST containing these edges and "NO" (of course without quotes again) otherwise.

Example

input

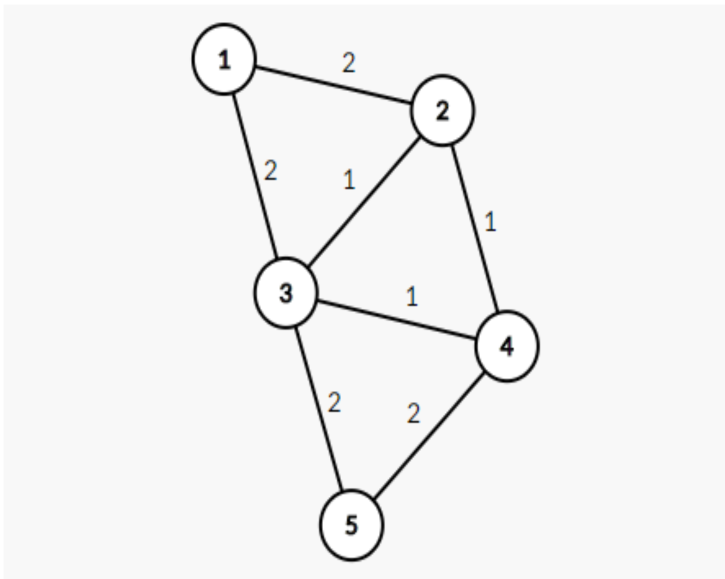
```
5 7
1 2 2
1 3 2
2 3 1
2 4 1
3 4 1
3 5 2
4 5 2
4
2 3 4
3 3 4 5
2 1 7
2 1 2
```

output

```
YES
NO
YES
NO
```

Note

This is the graph of sample:



Weight of minimum spanning tree on this graph is 6.

MST with edges $(1, 3, 4, 6)$, contains all of edges from the first query, so answer on the first query is "`YES`".

Edges from the second query form a cycle of length 3, so there is no spanning tree including these three edges. Thus, answer is "`NO`".

F. Imbalance Value of a Tree

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a tree T consisting of n vertices. A number is written on each vertex; the number written on vertex i is a_i. Let's denote the function $I(x, y)$ as the difference between maximum and minimum value of a_i on a simple path connecting vertices x and y.

Your task is to calculate $\sum_{i=1}^{n} \sum_{j=i}^{n} I(i, j)$.

Input

The first line contains one integer number n ($1 \le n \le 10^6$) — the number of vertices in the tree.

The second line contains n integer numbers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^6$) — the numbers written on the vertices.

Then $n - 1$ lines follow. Each line contains two integers x and y denoting an edge connecting vertex x and vertex y ($1 \le x, y \le n, x \ne y$). It is guaranteed that these edges denote a tree.

Output

Print one number equal to $\sum_{i=1}^{n} \sum_{j=i}^{n} I(i, j)$.

Example

input

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

output

```
6
```