# GPC-UPC Minimum Spanning Tree Contest

## A. st-Spanning Tree

4 seconds, 256 megabytes

You are given an undirected connected graph consisting of $n$ vertices and $m$ edges. There are no loops and no multiple edges in the graph.

You are also given two distinct vertices $s$ and $t$, and two values $d_s$ and $d_t$. Your task is to build any spanning tree of the given graph (note that the graph is not weighted), such that the degree of the vertex $s$ doesn't exceed $d_s$, and the degree of the vertex $t$ doesn't exceed $d_t$, or determine, that there is no such spanning tree.

The *spanning tree* of the graph $G$ is a subgraph which is a tree and contains all vertices of the graph $G$. In other words, it is a connected graph which contains $n - 1$ edges and can be obtained by removing some of the edges from $G$.

The degree of a vertex is the number of edges incident to this vertex.

### Input

The first line of the input contains two integers $n$ and $m$ ($2 \le n \le 200\,000$, $1 \le m \le min(400\,000, n \cdot (n - 1) / 2)$) — the number of vertices and the number of edges in the graph.

The next $m$ lines contain the descriptions of the graph's edges. Each of the lines contains two integers $u$ and $v$ ($1 \le u, v \le n$, $u \ne v$) — the ends of the corresponding edge. It is guaranteed that the graph contains no loops and no multiple edges and that it is connected.

The last line contains four integers $s, t, d_s, d_t$ ($1 \le s, t \le n$, $s \ne t$, $1 \le d_s, d_t \le n - 1$).

### Output

If the answer doesn't exist print "No" (without quotes) in the only line of the output.

Otherwise, in the first line print "Yes" (without quotes). In the each of the next $(n - 1)$ lines print two integers — the description of the edges of the spanning tree. Each of the edges of the spanning tree must be printed exactly once.

You can output edges in any order. You can output the ends of each edge in any order.

If there are several solutions, print any of them.

input
```
3 3
1 2
2 3
3 1
1 2 1 1
```
output
```
Yes
3 2
1 3
```

input
```
7 8
7 4
1 3
5 4
5 7
3 2
2 4
6 1
1 2
6 4 1 4
```
output
```
Yes
1 3
5 7
3 2
7 4
2 4
6 1
```

## B. Highway Decommission

0.3 seconds, 256 megabytes

Nlogonia's government is eager to cut down public debt. One of the measures about to take place is the decommission of some highways as most of them incur a high maintenance cost. Each highway connects two different cities and can be traveled in both directions. Using the existing highways it is possible to reach any city from any other city.

Government promises that the impact of the decommission will be minimal in the lives of Nlogonians. In particular they guarantee that after the decommission, for each city the minimum distance needed to travel from that city to the capital of the country will remain the same as it is now, when all the highways can be used.

The Department of Roads of Nlogonia believes that interns are not there just to get coffees or run errands but should do meaningful work instead and that's why you are assigned the following task. Given the length and maintenance cost of each highway, you must decide which highways will be kept active and which will be decommissioned. As you might guess, the sum of maintenance costs for the remaining highways must be minimum.

### Input

The first line contains two integers $N$ $(2 \le N \le 10^4)$ and $M$ $(1 \le M \le 10^5)$, indicating respectively the number of cities and the number of highways. Cities are identified by distinct integers from $1$ to $N$, where city $1$ is the capital of Nlogonia. Each of the following M lines describes a highway with four integers $A$, $B$, $L$ and $C$ $(1 \le A, B \le N, A \ne B$ and $1 \le L, C \le 10^9)$, indicating that there is a highway between cities $A$ and $B$ that has length $L$ and maintenance cost $C$. Using the existing highways it is possible to reach any city from any other city.

### Output

Output a single line with an integer indicating the minimum possible sum of maintenance costs for a set of highways to be kept active. This set of highways must ensure that for each city the minimum distance needed to travel from that city to the capital of Nlogonia remains the same using only those highways.

input
```
3 4
2 3 2 4
2 3 2 2
1 2 5 1
1 3 1 4
```
output
```
6
```

input
```
2 2
1 2 10 5
2 1 6 11
```
output
```
11
```

## C. Edges in MST

2 seconds, 256 megabytes

You are given a connected weighted undirected graph without any loops and multiple edges.

Let us remind you that a graph's spanning tree is defined as an acyclic connected subgraph of the given graph that includes all of the graph's vertexes. The weight of a tree is defined as the sum of weights of the edges that the given tree contains. The minimum spanning tree (**MST**) of a graph is defined as the graph's spanning tree having the minimum possible weight. For any connected graph obviously exists the minimum spanning tree, but in the general case, a graph's minimum spanning tree is not unique.

Your task is to determine the following for each edge of the given graph: whether it is either included in **any** MST, or included **at least in one** MST, or **not included in any** MST.

### Input

The first line contains two integers $n$ and $m$ ($2 \le n \le 10^5$, $n-1 \le m \le min(10^5, \frac{n(n-1)}{2})$) — the number of the graph's vertexes and edges, correspondingly. Then follow $m$ lines, each of them contains three integers — the description of the graph's edges as "$a_i$ $b_i$ $w_i$" ($1 \le a_i, b_i \le n$, $1 \le w_i \le 10^6$, $a_i \ne b_i$), where $a_i$ and $b_i$ are the numbers of vertexes connected by the $i$-th edge, $w_i$ is the edge's weight. It is guaranteed that the graph is connected and doesn't contain loops or multiple edges.

### Output

Print $m$ lines — the answers for all edges. If the $i$-th edge is included in any MST, print "`any`"; if the $i$-th edge is included at least in one MST, print "`at least one`"; if the $i$-th edge isn't included in any MST, print "`none`". Print the answers for the edges in the order in which the edges are specified in the input.

**input**
```
4 5
1 2 101
1 3 100
2 3 2
2 4 2
3 4 1
```
**output**
```
none
any
at least one
at least one
any
```

**input**
```
3 3
1 2 1
2 3 1
1 3 2
```
**output**
```
any
any
none
```

**input**
```
3 3
1 2 1
2 3 1
1 3 1
```
**output**
```
at least one
at least one
at least one
```

In the second sample the MST is unique for the given graph: it contains two first edges.

In the third sample any two edges form the MST for the given graph. That means that each edge is included at least in one MST.

## D. Hierarchy

2 seconds, 64 megabytes

Nick's company employed $n$ people. Now Nick needs to build a tree hierarchy of «supervisor-surbodinate» relations in the company (this is to say that each employee, except one, has exactly one supervisor). There are $m$ applications written in the following form: «employee $a_i$ is ready to become a supervisor of employee $b_i$ at extra cost $c_i$». The qualification $q_j$ of each employee is known, and for each application the following is true: $q_{a_i} > q_{b_i}$.

Would you help Nick calculate the minimum cost of such a hierarchy, or find out that it is impossible to build it.

### Input

The first input line contains integer $n$ ($1 \le n \le 1000$) — amount of employees in the company. The following line contains $n$ space-separated numbers $q_j$ ($0 \le q_j \le 10^6$)— the employees' qualifications. The following line contains number $m$ ($0 \le m \le 10000$) — amount of received applications. The following $m$ lines contain the applications themselves, each of them in the form of three space-separated numbers: $a_i$, $b_i$ and $c_i$ ($1 \le a_i, b_i \le n$, $0 \le c_i \le 10^6$). Different applications can be similar, i.e. they can come from one and the same employee who offered to become a supervisor of the same person but at a different cost. For each application $q_{a_i} > q_{b_i}$.

### Output

Output the only line — the minimum cost of building such a hierarchy, or `-1` if it is impossible to build it.

**input**
```
4
7 2 3 1
4
1 2 5
2 4 1
3 4 1
1 3 5
```
**output**
```
11
```

**input**
```
3
1 2 3
2
3 1 2
3 1 3
```
**output**
```
-1
```

In the first sample one of the possible ways for building a hierarchy is to take applications with indexes 1, 2 and 4, which give 11 as the minimum total cost. In the second sample it is impossible to build the required hierarchy, so the answer is `-1`.

## E. Envy

2 seconds, 256 megabytes

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.

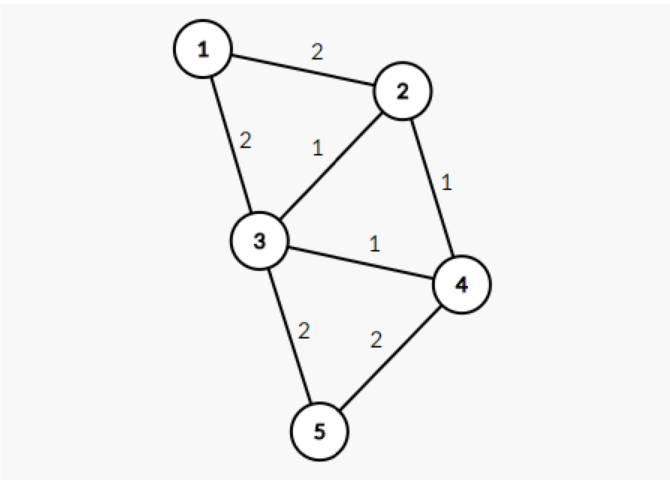
**input**

```
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
```

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. MST Unification

3 seconds, 256 megabytes

You are given an undirected weighted **connected** graph with $n$ vertices and $m$ edges **without loops and multiple edges**.

The $i$-th edge is $e_i = (u_i, v_i, w_i)$; the distance between vertices $u_i$ and $v_i$ along the edge $e_i$ is $w_i$ ($1 \le w_i$). The graph is **connected**, i. e. for any pair of vertices, there is at least one path between them consisting only of edges of the given graph.

A minimum spanning tree (MST) in case of **positive** weights is a subset of the edges of a connected weighted undirected graph that connects all the vertices together and has minimum total cost among all such subsets (total cost is the sum of costs of chosen edges).

You can modify the given graph. The only operation you can perform is the following: increase the weight of some edge by $1$. You **can** increase the weight of each edge multiple (possibly, zero) times.

Suppose that the initial MST cost is $k$. Your problem is to increase weights of some edges **with minimum possible number of operations** in such a way that the cost of MST in the obtained graph remains $k$, but MST is **unique** (it means that there is only one way to choose MST in the obtained graph).

Your problem is to calculate the **minimum** number of operations required to do it.

### Input

The first line of the input 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 the number of edges in the initial graph.

The next $m$ lines contain three integers each. The $i$-th line contains the description of the $i$-th edge $e_i$. It is denoted by three integers $u_i, v_i$ and $w_i$ ($1 \le u_i, v_i \le n, u_i \neq v_i, 1 \le w \le 10^9$), where $u_i$ and $v_i$ are vertices connected by the $i$-th edge and $w_i$ is the weight of this edge.

It is guaranteed that the given graph **doesn't contain loops and multiple edges** (i.e. for each $i$ from $1$ to $m$ $u_i \neq v_i$ and for each unordered pair of vertices $(u, v)$ there is at most one edge connecting this pair of vertices). It is also guaranteed that the given graph is **connected**.

### Output

Print one integer — the **minimum** number of operations to unify MST of the initial graph without changing the cost of MST.

**input**

```
8 10
1 2 1
2 3 2
2 4 5
1 4 2
6 3 3
6 1 3
3 5 2
3 7 1
4 8 1
6 2 4
```

**output**

```
1
```

**input**

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

**output**

```
0
```

**input**

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

**output**

```
0
```

**input**

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

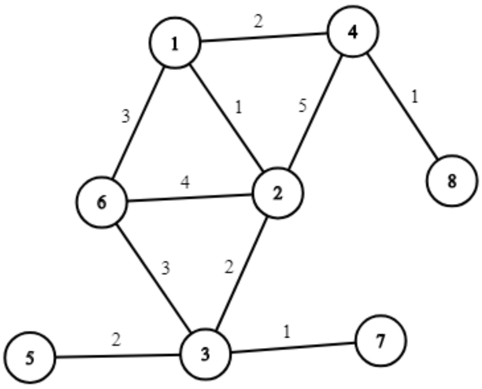
**output**

```
1
```

**input**
```
1 0
```
**output**
```
0
```

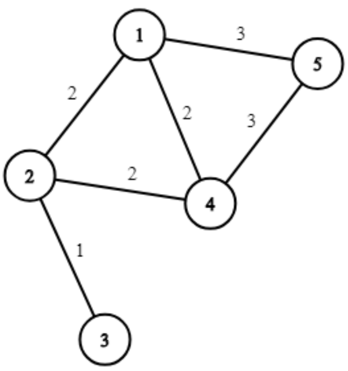
**input**
```
5 6
1 2 2
2 3 1
4 5 3
2 4 2
1 4 2
1 5 3
```
**output**
```
2
```

The picture corresponding to the first example:




You can, for example, increase weight of the edge $(1, 6)$ or $(6, 3)$ by $1$ to unify MST.

The picture corresponding to the last example:




You can, for example, increase weights of edges $(1, 5)$ and $(2, 4)$ by $1$ to unify MST.

## G. MST Company

8 seconds, 256 megabytes

The MST (Meaningless State Team) company won another tender for an important state reform in Berland.

There are $n$ cities in Berland, some pairs of the cities are connected by roads. Each road has its price. One can move along any road in any direction. The MST team should carry out the repair works on some set of roads such that one can get from any city to any other one moving only along the repaired roads. Moreover, this set should contain exactly $k$ capital roads (that is, the roads that start or finish in the capital). The number of the capital is 1.

As the budget has already been approved, the MST Company will profit by finding the set with minimum lengths of roads.

### Input

The first input line contains three integers $n, m, k$ $(1 \le n \le 5000; 0 \le m \le 10^5; 0 \le k < 5000)$, where $n$ is the number of cities in the country, $m$ is the number of roads in the country, $k$ is the number of capital roads in the required set. Then $m$ lines enumerate the roads in question. Each road is specified by three numbers $a_i, b_i, w_i$ $(1 \le a_i, b_i \le n;\ 1 \le w \le 10^5)$, where $a_i, b_i$ are the numbers of cities linked by a road and $w_i$ is its length.

Between each pair of cities no more than one road exists. There are no roads that start and finish in one city. The capital's number is 1.

### Output

In the first line print the number of roads in the required set. The second line should contain the numbers of roads included in the sought set. If the sought set does not exist, print -1.

**input**
```
4 5 2
1 2 1
2 3 1
3 4 1
1 3 3
1 4 2
```
**output**
```
3
1 5 2
```

## H. The Child and Zoo

2 seconds, 256 megabytes

Of course our child likes walking in a zoo. The zoo has $n$ areas, that are numbered from $1$ to $n$. The $i$-th area contains $a_i$ animals in it. Also there are $m$ roads in the zoo, and each road connects two distinct areas. Naturally the zoo is connected, so you can reach any area of the zoo from any other area using the roads.

Our child is very smart. Imagine the child want to go from area $p$ to area $q$. Firstly he considers all the simple routes from $p$ to $q$. For each route the child writes down the number, that is equal to the minimum number of animals among the route areas. Let's denote the largest of the written numbers as $f(p, q)$. Finally, the child chooses one of the routes for which he writes down the value $f(p, q)$.

After the child has visited the zoo, he thinks about the question: what is the average value of $f(p, q)$ for all pairs $p, q$ $(p \ne q)$? Can you answer his question?

### Input

The first line contains two integers $n$ and $m$ $(2 \le n \le 10^5;\ 0 \le m \le 10^5)$. The second line contains $n$ integers: $a_1, a_2, ..., a_n$ $(0 \le a_i \le 10^5)$. Then follow $m$ lines, each line contains two integers $x_i$ and $y_i$ $(1 \le x_i, y_i \le n;$ $x_i \ne y_i)$, denoting the road between areas $x_i$ and $y_i$.

All roads are bidirectional, each pair of areas is connected by at most one road.

### Output

Output a real number — the value of $\frac{\sum_{p,q,p \neq q} f(p,q)}{n(n-1)}$.

The answer will be considered correct if its relative or absolute error doesn't exceed $10^{-4}$.

**input**

```
4 3
10 20 30 40
1 3
2 3
4 3
```

**output**

```
16.666667
```

**input**

```
3 3
10 20 30
1 2
2 3
3 1
```

**output**

```
13.333333
```

**input**

```
7 8
40 20 10 30 20 50 40
1 2
2 3
3 4
4 5
5 6
6 7
1 4
5 7
```

**output**

```
18.571429
```

Consider the first sample. There are $12$ possible situations:

- $p = 1, q = 3, f(p, q) = 10$.
- $p = 2, q = 3, f(p, q) = 20$.
- $p = 4, q = 3, f(p, q) = 30$.
- $p = 1, q = 2, f(p, q) = 10$.
- $p = 2, q = 4, f(p, q) = 20$.
- $p = 4, q = 1, f(p, q) = 10$.

Another $6$ cases are symmetrical to the above. The average is $\frac{(10+20+30+10+20+10)\times 2}{12} \approx 16.666667$.

Consider the second sample. There are $6$ possible situations:

- $p = 1, q = 2, f(p, q) = 10$.
- $p = 2, q = 3, f(p, q) = 20$.
- $p = 1, q = 3, f(p, q) = 10$.

Another $3$ cases are symmetrical to the above. The average is $\frac{(10+20+10)\times 2}{6} \approx 13.333333$.

## I. Lazy Student

2 seconds, 256 megabytes

Student Vladislav came to his programming exam completely unprepared as usual. He got a question about some strange algorithm on a graph — something that will definitely never be useful in real life. He asked a girl sitting next to him to lend him some cheat papers for this questions and found there the following definition:

*The minimum spanning tree* $T$ of graph $G$ is such a tree that it contains all the vertices of the original graph $G$, and the sum of the weights of its edges is the minimum possible among all such trees.

Vladislav drew a graph with $n$ vertices and $m$ edges containing no loops and multiple edges. He found one of its minimum spanning trees and then wrote for each edge its weight and whether it is included in the found tree or not. Unfortunately, the piece of paper where the graph was painted is gone and the teacher is getting very angry and demands to see the original graph. Help Vladislav come up with a graph so that the information about the minimum spanning tree remains correct.

### Input

The first line of the input contains two integers $n$ and $m$ ($2 \leq n \leq 100\,000, 1 \leq m \leq 100\,000, n-1 \leq m \leq \frac{n(n-1)}{2}$) — the number of vertices and the number of edges in the graph.

Each of the next $m$ lines describes an edge of the graph and consists of two integers $a_j$ and $b_j$ ($1 \leq a_j \leq 10^9, b_j = \{0, 1\}$). The first of these numbers is the weight of the edge and the second number is equal to $1$ if this edge was included in the minimum spanning tree found by Vladislav, or $0$ if it was not.

It is guaranteed that exactly $n - 1$ number $\{b_j\}$ are equal to one and exactly $m - n + 1$ of them are equal to zero.

### Output

If Vladislav has made a mistake and such graph doesn't exist, print $-1$.

Otherwise print $m$ lines. On the $j$-th line print a pair of vertices $(u_j, v_j)$ ($1 \leq u_j, v_j \leq n, u_j \neq v_j$), that should be connected by the $j$-th edge. The edges are numbered in the same order as in the input. The graph, determined by these edges, must be connected, contain no loops or multiple edges and its edges with $b_j = 1$ must define the minimum spanning tree. In case there are multiple possible solutions, print any of them.

**input**

```
4 5
2 1
3 1
4 0
1 1
5 0
```

**output**

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

**input**

```
3 3
1 0
2 1
3 1
```

**output**

```
-1
```

## J. Hamiltonian Spanning Tree

2 seconds, 256 megabytes

A group of $n$ cities is connected by a network of roads. There is an undirected road between every pair of cities, so there are $\frac{n \cdot (n-1)}{2}$ roads in total. It takes exactly $y$ seconds to traverse **any** single road.

A *spanning tree* is a set of roads containing exactly $n - 1$ roads such that it's possible to travel between any two cities using only these roads.

Some spanning tree of the initial network was chosen. For every road in this tree the time one needs to traverse this road was changed from $y$ to $x$ seconds. Note that it's not guaranteed that $x$ is smaller than $y$.

You would like to travel through all the cities using the shortest path possible. Given $n$, $x$, $y$ and a description of the spanning tree that was chosen, find the cost of the shortest path that starts in any city, ends in any city and visits all cities **exactly once**.

### Input

The first line of the input contains three integers $n$, $x$ and $y$ ($2 \le n \le 200\,000$, $1 \le x, y \le 10^9$).

Each of the next $n - 1$ lines contains a description of a road in the spanning tree. The $i$-th of these lines contains two integers $u_i$ and $v_i$ ($1 \le u_i, v_i \le n$) — indices of the cities connected by the $i$-th road. It is guaranteed that these roads form a spanning tree.

### Output

Print a single integer — the minimum number of seconds one needs to spend in order to visit all the cities exactly once.

**input**

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

**output**

```
9
```

**input**

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

**output**

```
8
```

In the first sample, roads of the spanning tree have cost $2$, while other roads have cost $3$. One example of an optimal path is $5 \to 3 \to 4 \to 1 \to 2$.

In the second sample, we have the same spanning tree, but roads in the spanning tree cost 3, while other roads cost 2. One example of an optimal path is $1 \to 4 \to 5 \to 2 \to 3$.