A. Weird Sum

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

Egor has a table of size $n \times m$, with lines numbered from 1 to n and columns numbered from 1 to m. Each cell has a color that can be presented as an integer from 1 to 10^5.

Let us denote the cell that lies in the intersection of the r-th row and the c-th column as (r, c). We define the manhattan distance between two cells (r_1, c_1) and (r_2, c_2) as the length of a shortest path between them where each consecutive cells in the path must have a common side. The path can go through cells of any color. For example, in the table 3×4 the manhattan distance between $(1, 2)$ and $(3, 3)$ is 3, one of the shortest paths is the following: $(1, 2) \to (2, 2) \to (2, 3) \to (3, 3)$.

Egor decided to calculate the sum of manhattan distances between each pair of cells of the same color. Help him to calculate this sum.

Input

The first line contains two integers n and m ($1 \leq n \leq m$, $n \cdot m \leq 100\,000$) — number of rows and columns in the table.

Each of next n lines describes a row of the table. The i-th line contains m integers $c_{i1}, c_{i2}, \ldots, c_{im}$ ($1 \le c_{ij} \le 100\,000$) — colors of cells in the i-th row.

Output

Print one integer — the the sum of manhattan distances between each pair of cells of the same color.

Examples

input

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

output

```
7
```

input

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

output

```
76
```

input

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

output

```
129
```

Note

In the first sample there are three pairs of cells of same color: in cells $(1, 1)$ and $(2, 3)$, in cells $(1, 2)$ and $(2, 2)$, in cells $(1, 3)$ and $(2, 1)$. The manhattan distances between them are 3, 1 and 3, the sum is 7.

B. Integral Array

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given an array a of n positive integers numbered from 1 to n. Let's call an array *integral* if for any two, not necessarily different, numbers x and y from this array, $x \ge y$, the number $\left\lfloor \frac{x}{y} \right\rfloor$ (x divided by y with rounding down) is also in this array.

You are guaranteed that all numbers in a do not exceed c. Your task is to check whether this array is integral.

Input

The input consists of multiple test cases. The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases. Description of the test cases follows.

The first line of each test case contains two integers n and c ($1 \le n \le 10^6$, $1 \le c \le 10^6$) — the size of a and the limit for the numbers in the array.

The second line of each test case contains n integers a_1, a_2, ..., a_n ($1 \le a_i \le c$) — the array a.

Let N be the sum of n over all test cases and C be the sum of c over all test cases. It is guaranteed that $N \le 10^6$ and $C \le 10^6$.

Output

For each test case print Yes if the array is integral and No otherwise.

Examples

input

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

output

```
Yes
No
No
Yes
```

input

```
1
1 1000000
1000000
```

output

```
No
```

Note

In the first test case it is easy to see that the array is integral:

- $\left\lfloor \frac{1}{1} \right\rfloor = 1$, $a_1 = 1$, this number occurs in the arry
- $\left\lfloor \frac{2}{2} \right\rfloor = 1$
- $\left\lfloor \frac{5}{5} \right\rfloor = 1$
- $\left\lfloor \frac{2}{1} \right\rfloor = 2$, $a_2 = 2$, this number occurs in the array
- $\left\lfloor \frac{5}{1} \right\rfloor = 5$, $a_3 = 5$, this number occurs in the array
- $\left\lfloor \frac{5}{2} \right\rfloor = 2$, $a_2 = 2$, this number occurs in the array

Thus, the condition is met and the array is integral.

In the second test case it is enough to see that

$\left\lfloor \frac{7}{3} \right\rfloor = \left\lfloor 2\frac{1}{3} \right\rfloor = 2$, this number is not in a, that's why it is not integral.

In the third test case $\left\lfloor \frac{2}{2} \right\rfloor = 1$, but there is only 2 in the array, that's why it is not integral.

C. Tyler and Strings

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

While looking at the kitchen fridge, the little boy Tyler noticed magnets with symbols, that can be aligned into a string s.

Tyler likes strings, and especially those that are lexicographically smaller than another string, t. After playing with magnets on the fridge, he is wondering, how many distinct strings can be composed out of letters of string s by rearranging them, so that the resulting string is lexicographically smaller than the string t? Tyler is too young, so he can't answer this question. The alphabet Tyler uses is very large, so for your convenience he has already replaced the same letters in s and t to the same integers, keeping that different letters have been replaced to different integers.

We call a string x lexicographically smaller than a string y if one of the followings conditions is fulfilled:

- There exists such position of symbol m that is presented in both strings, so that before m-th symbol the strings are equal, and the m-th symbol of string x is smaller than m-th symbol of string y.
- String x is the prefix of string y and $x \neq y$.

Because the answer can be too large, print it modulo $998\,244\,353$.

Input

The first line contains two integers n and m ($1 \leq n, m \leq 200\,000$) — the lengths of strings s and t respectively.

The second line contains n integers $s_1, s_2, s_3, \ldots, s_n$ ($1 \leq s_i \leq 200\,000$) — letters of the string s.

The third line contains m integers $t_1, t_2, t_3, \ldots, t_m$ ($1 \leq t_i \leq 200\,000$) — letters of the string t.

Output

Print a single number — the number of strings lexicographically smaller than t that can be obtained by rearranging the letters in s, modulo $998\,244\,353$.

Examples

input

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

output

```
2
```

input

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

output

```
23
```

input

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

output

```
1
```

Note

In the first example, the strings we are interested in are $[1\,2\,2]$ and $[2\,1\,2]$. The string $[2\,2\,1]$ is lexicographically larger than the string $[2\,1\,2\,1]$, so we don't count it.

In the second example, all strings count except $[4\,3\,2\,1]$, so the answer is $4! - 1 = 23$.

In the third example, only the string $[1\,1\,1\,2]$ counts.

D. Serious Business

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Dima is taking part in a show organized by his friend Peter. In this show Dima is required to cross a $3 \times n$ rectangular field. Rows are numbered from 1 to 3 and columns are numbered from 1 to n.

The cell in the intersection of the i-th row and the j-th column of the field contains an integer $a_{i,j}$. Initially Dima's score equals zero, and whenever Dima reaches a cell in the row i and the column j, his score changes by $a_{i,j}$. Note that the score can become negative.

Initially all cells in the first and the third row are marked as available, and all cells in the second row are marked as unavailable. However, Peter offered Dima some help: there are q special offers in the show, the i-th special offer allows Dima to mark cells in the second row between l_i and r_i as available, though Dima's score reduces by k_i whenever he accepts a special offer. Dima is allowed to use as many special offers as he wants, and might mark the same cell as available multiple times.

Dima starts his journey in the cell $(1, 1)$ and would like to reach the cell $(3, n)$. He can move either down to the next row or right to the next column (meaning he could increase the current row or column by 1), thus making $n + 1$ moves in total, out of which exactly $n - 1$ would be horizontal and 2 would be vertical.

Peter promised Dima to pay him based on his final score, so the sum of all numbers of all visited cells minus the cost of all special offers used. Please help Dima to maximize his final score.

Input

The first input line contains two integers n and q ($1 \le n, q \le 500\,000$) — the number of columns in the field and the number of special offers.

The next three lines describe the field, i-th of them contains n integers a_{i1}, a_{i2}, ..., a_{in} ($-10^9 \le a_{ij} \le 10^9$) — the values in the i-th row.

The next q lines describe special offers: the i-th offer is described by 3 integers l_i, r_i and k_i ($1 \le l_i \le r_i \le n$, $1 \le k_i \le 10^9$) — the segment that becomes unblocked and the cost of this special offer.

Output

Output one integer — the maximum final score Dima can achieve.

Examples

input

```
4 3
1 0 2 -1
-3 1 9 2
3 2 4 1
1 2 5
2 3 4
1 4 14
```

output

```
13
```

input

```
5 4
-20 -10 -11 -10 1
1 3 3 6 3
14 -20 3 6 2
1 5 13
1 2 2
3 5 3
2 3 1
```

output

```
-4
```

Note

In the first example, it is optimal to use Peter's second offer of 4 rubles and go through the cells $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 3)$, $(3, 3)$, $(3, 4)$, earning $1 + 0 + 2 + 9 + 4 + 1 - 4 = 13$ rubles in total.

In the second example, it is optimal to use Peter's second and third offers of 2 and 3 rubles, respectively, and go through the cells $(1, 1)$, $(2, 1)$, $(2, 2)$, $(2, 3)$, $(2, 4)$, $(3, 4)$, $(3, 5)$, earning $-20 + 1 + 3 + 3 + 6 + 6 + 2 - 2 - 3 = -4$ rubles.

E. Air Reform

time limit per test: 3 seconds

memory limit per test: 512 megabytes

input: standard input

output: standard output

Berland is a large country with developed airlines. In total, there are n cities in the country that are historically served by the Berlaflot airline. The airline operates bi-directional flights between m pairs of cities, i-th of them connects cities with numbers a_i and b_i and has a price c_i for a flight in both directions.

It is known that Berlaflot flights can be used to get from any city to any other (possibly with transfers), and the cost of any route that consists of several consequent flights is equal to the cost of the most expensive of them. More formally, the cost of the route from a city t_1 to a city t_k with $(k - 2)$ transfers using cities $t_2,\ t_3,\ t_4,\ \ldots,\ t_{k-1}$ is equal to the maximum cost of flights from t_1 to t_2, from t_2 to t_3, from t_3 to t_4 and so on until the flight from t_{k-1} to t_k. Of course, all these flights must be operated by Berlaflot.

A new airline, S8 Airlines, has recently started operating in Berland. This airline provides bi-directional flights between all pairs of cities that are not connected by Berlaflot direct flights. Thus, between each pair of cities there is a flight of either Berlaflot or S8 Airlines.

The cost of S8 Airlines flights is calculated as follows: for each pair of cities x and y that is connected by a S8 Airlines flight, the cost of this flight is equal to the minimum cost of the route between the cities x and y at Berlaflot according to the pricing described earlier.

It is known that with the help of S8 Airlines flights you can get from any city to any other with possible transfers, and, similarly to

Berlaflot, the cost of a route between any two cities that consists of several S8 Airlines flights is equal to the cost of the most expensive flight.

Due to the increased competition with S8 Airlines, Berlaflot decided to introduce an air reform and change the costs of its flights. Namely, for the i-th of its flight between the cities a_i and b_i, Berlaflot wants to make the cost of this flight equal to the minimum cost of the route between the cities a_i and b_i at S8 Airlines. Help Berlaflot managers calculate new flight costs.

Input

Each test consists of multiple test cases. The first line contains one integer t ($1 \le t \le 10\,000$) — the amount of test cases.

The first line of each test case contains two integers n and m ($4 \le n \le 200\,000$, $n - 1 \le m \le 200\,000$, $m \le \frac{(n-1)(n-2)}{2}$) — the amount of cities in Berland and the amount of Berlaflot flights.

The next m lines contain the description of Berlaflot flights. The i-th line contains three integers a_i, b_i and c_i ($1 \le a_i, b_i \le n$, $1 \le c_i \le 10^9$) — the numbers of cities that are connected with i-th Berlaflot flight and the price of i-th Berlaflot flight.

It is guaranteed that no flight connects a city with itself, no two flights connect the same pair of cities. It is guaranteed that by using Berlaflot flights it is possible to get from any city to any other and by using S8 Airlines flights it is possible to get from any city to any other.

Let N be the sum of n over all test cases and M be the sum of m over all test cases. It is guaranteed that $N, M \le 200\,000$.

Output

For each test case you should print m integers in a single line, i-th of them should be the price of i-th Berlaflot flight after the air reform.

Example

input

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

output

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

Note

In the first test case S8 Airlines will provide flights between these pairs of cities: $(1, 3)$, $(1, 4)$ and $(2, 4)$.

The cost of a flight between cities 1 and 3 will be equal to 2, since the minimum cost of the Berlaflot route is 2 — the route consists of a flight between cities 1 and 2 costing 1 and a flight between cities 2 and 3 costing 2, the maximum cost is 2.

The cost of a flight between cities 1 and 4 will be 3, since the minimum cost of the Berlaflot route is 3 — the route consists of a flight between cities 1 and 2 costing 1, a flight between cities 2 and 3 costing 2 and a flight between cities 3 and 4 costing 3, the maximum cost is 3.

The cost of a flight between cities 2 and 4 will be 3, since the minimum cost of the Berlaflot route is 3 — the route consists of a flight between cities 2 and 3 costing 2 and a flight between cities 3 and 4 costing 3, the maximum cost is 3.

After the air reform, the cost of the Berlaflot flight between cities 1 and 2 will be 3, since the minimum cost of the S8 Airlines route between these cities is 3 — the route consists of a flight between cities 1 and 4 costing 3 and a flight between cities 2 and 4 costing 3, the maximum cost is 3.

The cost of the Berlaflot flight between cities 2 and 3 will be 3, since the minimum cost of the S8 Airlines route between these cities is 3 — the route consists of a flight between cities 2 and 4 costing 3, a flight between cities 1 and 4 costing 3 and a flight between 1 and 3 costing 2, the maximum cost is 3.

The cost of the Berlaflot flight between cities 3 and 4 will be 3, since the minimum cost of the S8 Airlines route between these cities is 3 — the route consists of a flight between cities 1 and 3 costing 2 and a flight between cities 1 and 4 costing 3, the maximum cost is 3.

In the second test case S8 Airlines will have the following flights: between cities 1 and 4 costing 1, between cities 2 and 3 costing 1, between cities 2 and 5 costing 2, between cities 3 and 4 costing 1 and between cities 3 and 5 costing 2.

F. Two Avenues

time limit per test: 8 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

In order to make the capital of Berland a more attractive place for tourists, the great king came up with the following plan: choose two streets of the city and call them avenues. Certainly, these avenues will be proclaimed extremely important historical places, which should attract tourists from all over the world.

The capital of Berland can be represented as a graph, the vertices of which are crossroads, and the edges are streets connecting two crossroads. In total, there are n vertices and m edges in the graph, you can move in both directions along any street, you can get from any crossroad to any other by moving only along the streets, each street connects two different crossroads, and no two streets connect the same pair of crossroads.

In order to reduce the flow of ordinary citizens moving along the great avenues, it was decided to introduce a toll on each avenue in both directions. Now you need to pay 1 tugrik for one passage along the avenue. You don't have to pay for the rest of the streets.

Analysts have collected a sample of k citizens, i-th of them needs to go to work from the crossroad a_i to the crossroad b_i. After two avenues are chosen, each citizen will go to work along the path with minimal cost.

In order to earn as much money as possible, it was decided to choose two streets as two avenues, so that the total number of tugriks paid by these k citizens is maximized. Help the king: according to the given scheme of the city and a sample of citizens, find out which two streets should be made avenues, and how many tugriks the citizens will pay according to this choice.

Input

Each test consists of multiple test cases. The first line contains one integer t ($1 \le t \le 10^5$) — the number of test cases.

The first line of each test case contains two integers n and m ($3 \le n \le 500\,000$, $n-1 \le m \le 500\,000$, $m \le \frac{n(n-1)}{2}$) — the number of crossroads and streets, respectively.

The next m lines contain the description of streets, i-th line contains two integers s_i and f_i ($1 \le s_i, f_i \le n$, $s_i \ne f_i$) — indexes of crossroads which are connected by i-th street. It is guaranteed that no two streets connect the same pair of crossroads, you can get from any crossroad to any other by moving only along the streets.

The next line contains a single integer k ($1 \le k \le 500\,000$) — the amount of citizens in the sample.

The next k lines contain the description of citizens, i-th line contains two integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) — i-th citizen goes to work from crossroad a_i to crossroad b_i.

Let M be the sum of m over all test cases and K be the sum of k over all test cases. It is guaranteed that $M, K \le 500\,000$.

Output

For each test case print the answer to the problem.

In the first line print the total amount of tugriks that will be paid by citizens.

In the second line print two integers x_1 and y_1 — the numbers of crossroads that will be connected by the first avenue.

In the third line print two integers x_2 and y_2 — the numbers of crossroads that will be connected by the second avenue.

The numbers of crossroads connected by an avenue can be printed in any order, each of the printed streets should be among m streets of the city, chosen streets should be different.

Example

input

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

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

output

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