Riot Rumble!

You are given a directed graph **G**, with **n** cities and **m** unidirectional roads. Since its around 1947, riots are on the rise, and you have decided to build **anti-riot stations** at some of the cities and ensure all cities are **protected**.

An anti-riot station in city **i** can protect a city **j** if and only if:

1) **i** is equal to **j**

OR

2) You can reach city **j** from city **i** and also reach city **i** from city **j**

Each city **i** requires **cost[i]** rupees to build an anti-riot station there. You have to find the MINIMUM amount of money required to protect all **n** cities, using MININUM amount of riot-stations, and also find the number of ways in which this minimum amount of money can be achieved. Two ways are different if the sets of cities you have installed anti-riot stations in, are not the same.

So first minimize money, then minimize number of stations, and then output number of ways in which this can be done.

INPUT

First line contains **T**, number of testcases.
Each testcase contains **n** and **m**. Next line contains n values denoting the cost array. Next m lines contain two integers **u** and **v**, indicating that there is a directed edge from u to v.

OUTPUT

Output T lines, one for each test case containing - minimum money required, a whitespace, and (number of ways to do that)%1000000007. Do not include any extra whitespaces or new lines in your output.

CONSTRAINTS

1<= T <=10
1<= n <=100000
0<= m <=300000
1<= u,v <=n
0<= cost[i] <=1000
Time Limit : 2 seconds

SAMPLE INPUT

```
4
3 3
1 2 3
1 2
2 3
3 2
5 6
2 8 0 6 0
1 4
1 3
2 4
3 4
4 5
5 1
10 12
1 3 2 2 1 3 1 4 10 10
1 2
2 3
3 1
3 4
4 5
5 6
5 7
6 4
7 3
8 9
9 10
10 9
2 2
7 91
1 2
2 1
```

SAMPLE OUTPUT

```
3 1
8 2
15 6
7 1
```