Problem 2

The war-torn country of VirtualLand consists of N cities. There are roads connecting the cities. A road connects two cities, only one road may exist for a pair of cities, and a road can be used for travel in both ways. VirtualLand has been in a war for a few years and many of the roads are destroyed. Now the citizens of VirtualLand want to rebuild the road system so that each pair of cities is connected by **one and only one** path. A path is a continuous sequence of roads. For military reasons they do not want multiple paths connecting a pair of cities. It may not be possible to build a road between some pairs of cities, so for a pair of cities there are three possible cases where: a road already exists, it is possible to build a road, and it is not possible to build a road between them. Note that you may have to destroy an existing road during the rebuilding process. There are costs when building or destroying roads and the costs may vary according to each road. (This VirtualLand exists in a virtual world and you don't have to worry about cases where two different roads have to cross each other.)

The above example shows a country with 5 cities. On the left is the initial situation where the 3 solid lines indicate the roads that already exist and the 3 dotted lines indicate the roads that may be built. The best way is to destroy the road between cities 1 and 2 (with cost 1) and build the roads between cities 3 and 5 and between cities 4 and 5 (with cost 6). The total cost amounts to 7. A careful consideration of all possible cases shows that the best answer is unique.

You are to write a program that, given the number of cities, the roads that already exist with the cost of destroying each one, and the roads that may be built with the cost of building each one, finds the minimum cost of satisfying the conditions mentioned above and the uniqueness of the answer.

[Input]

The first line of the input file contains the number T of test cases in the file. In each test case, the first line has three integers N ($1 \le N \le 100,000$), M ($0 \le M \le 300,000$), and K ($0 \le K \le 300,000$). Note that M or K may be 0. However, in no cases both will be 0. The cities are

numbered from 1 to N. In each of the next M lines, three integers are given which indicate the cities that are connected by each existing road and the cost for destroying the road. In each of the next K lines, three integers are given which indicate the cities that are connected by each road that may be built and the cost for building it. All costs are positive integer values that can be stored in a 4 byte integer variable. For each test case it is guaranteed that a solution exists and it is also guaranteed that the minimum cost can be correctly stored in a 64-bit integer (long long type) variable.

Because of the large size of the input files, it is likely required that the fastest I/O functions for the programming language of your choice to be used.

There are two kinds of inputs listed as follows.

- Small Set: T ≤ 100, 1 ≤ N ≤ 100, 0 ≤ M ≤ 500, 0 ≤ K ≤ 500
- Large Set: T ≤ 100, 1 ≤ N ≤ 100,000, 0 ≤ M ≤ 300,000, 0 ≤ K ≤ 300,000

[Output]

Print one line for each test case. In the line, separated by a single space, you should print the nonnegative integer cost and then print a 1 if the solution is unique and 0 otherwise.

[I/O Example]

Input

```
2
5 3 3
1 2 1
1 3 4
2 3 2
3 5 2
3 4 7
4 5 4
3 3 0
1 2 3
1 3 3
2 3 3
```

Output

```
7 1
3 0
```