J. Quelling Blade

[Description]



Mr. Sheep lost himself in a computer game. In this game, he plays the part of a super hero and fight with the evil. The equipment is very important in this game and Mr. Sheep thinks the Quelling Blade is the most powerful weapon.

In this game, each type of weapon costs hero some money, and brings the hero benefits. If the hero buys two weapons (no matter they have the same type or not), the benefit values are accumulated. That is to say, if the hero buys two weapons with benefit 3 and 5, the hero will get total benefit value 8=3+5.

There are some requirements for each weapon. If the hero wants to buy a certain weapon, he may need some other weapons first. For example, if hero wants to buy a “Divine Rapier”, he needs a “Demon Edge” and a “Scared Relic”. Of course, if he wants to buy the second “Devine Rapier”, he must buy another “Demon Edge” and another “Scared Relic” first. Notice that the existing weapon will not disappear after the trade. Note that a weapon may need multiple weapons with same type. And you may assume that a type of weapon is required by at most one other type of weapon.

The hero is busy with combat and has no time to earn money. Fortunately, the game will give the hero 1 coin per second. So if the hero wants to buy a “Quelling Blade”, the minimum total time for him to achieve his goal can be easily calculated.

Mr. Sheep is a perfectionist. He not only wants to get the “Quelling Blade” as soon as possible, but also wants to optimize every second during the game. He defines the utility of the game as the sum of the benefit value of the hero in each second. He calculates the utility from the start of the game until the second he gets “Quelling Blade”, exclusively. You may refer to the samples for further clarification. In the other words, you should define a way of process to buy the weapons for the hero, which minimize the total time to get “Quelling Blade” and optimize the utility of the game.

[Input]

There are hundreds of test cases, the number of test case are in the first line of the input. Notice that most of the test cases are relatively small.

For each test case, the first line contains a single integer N denoting the number of different types of weapons. (1 <= N <= 1000)

The next lines are describing the weapons. For each weapon, the first line contains two integers B and C, denoting the benefit value and the cost of this kind

of weapon. (1 <= B, C <= 2^31-1) Then a single integer P in the next line describes the number of requirements of this weapon. Next P lines, each line contains two integers I and A, means that this weapon needs A weapons of index I.

The indexes of weapons are start from 1 to N. The “Quelling Blade” is the first type of weapon. And you may assume that the total numbers of weapons which are needed by the “Quelling Blade” is less than 1000000. Also notice that “Quelling Blade” can be brought in a finite game time and a type of weapon can be required by at most one other type of weapon.

[Output]

For each test case, output a single integer denoting the optimal utility. You may assume that the answer is fit in 64-bit signed integer.

[Sample Input]

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

[Sample Output]

```
Case #1: 14
Case #2: 17
```