

Come Rain or Shine

Filename: longpath

The Problem

You have noticed that each time you leave home to go somewhere the maps app on your GPS by default always gives you directions to reach your destination in the shortest possible time. You then remember Ramya telling you to enjoy the journey and to look at the clear skies, the pretty daisies, and feel the gentle breeze against your face. You decide to take her wisdom literally and change the app so that if the weather is sunny, it gives you the *longest* distance to your destination. Otherwise, if the weather is rainy, you want the app to give you the shortest distance as usual. You want the app to be flexible, so it should work in several cities.

For the purposes of this problem, so that a longest path is always well-defined, the set of roads you will be allowed to take will only be one-way roads with no cycles formed by the set and there will always exist a path from home to the final destination. *Note: It may be the case that there is more than one road that connects the same pair of intersections.*

The Input

The first line of the input file will contain a single positive integer, c ($c \leq 200$), representing the number of cities to analyze. The first line of each city will have two space separated positive integers, v ($v \leq 1000$), and e ($e \leq 5000$), representing the number of intersections in the city and the number of one way streets, respectively. The following e lines will contain three integers each: s ($0 \leq s < v$), d ($0 \leq d < v$, $d \neq s$), m ($0 < m \leq 1000000$), representing the starting intersection, destination intersection and number of minutes it takes to traverse the street from s to d . Intersection 0 will always represent home (starting point) while intersection $v-1$, will always represent the final destination.

The Output

For each test case, output the shortest distance from home to the final destination, followed by the longest distance from home to the final destination, separated by a space.

Sample Input

```
2
6 7
0 1 1
1 2 2
2 5 3
1 5 6
0 3 5
3 4 6
4 5 7
3 3
0 1 7
0 2 19
1 2 8
```

Sample Output

```
6 18
15 19
```