Competitive Programming SS23

Submit until end of contest

Problem: battery (1.5 second timelimit)

You are buying a new electric car. The most expensive part of the car is the battery. The price increases with the capacity of the battery. To save money, you decide to buy the smallest battery, which fulfills your requirements.

You are living in city `0` and you have to reach at least 75% of all *other* cities. You can recharge your car in every city, but due to the lacking infrastructure, you can't recharge your battery on highways.

Please decide what the capacity of your battery should be.

Input The first line contains the number t $(1 \leq t \leq 1000)$, the number of different city maps (test cases).

Each test case starts with two numbers n and m $(2 \leq n \leq 10^5, 1 \leq m \leq 2 \cdot 10^5)$. n is the number of cities in that map, m the number of streets. Each of the following m lines contains one street a, b and w $(0 \leq a, b < n, 1 \leq w \leq 10^9)$. a and b are the connected cities, w the length of the street (= how much battery you need). You can drive every street in both directions.

The sum over all values of n does not exceed 10^5. The same goes for the values of m with the bound $2 \cdot 10^5$. You furthermore may assume that the city map is connected.

Output For each testcase print the minimum battery capacity needed to reach at least 75% of the other cities per city map.

Sample input

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

Sample output

```
2
2
```