

Tour de Byteland

As the mayor of Byteland's term of office draws to a close, he starts his preparations for reelection. For the first time in the 40 years of his political career his chances of victory seem somewhat uncertain. His main cause of worry are the disturbing results of an opinion poll which state that over 90% of the citizens regard the mayor as a portly, heavily smoking individual who sleeps in his armchair more or less all day.

After careful consultation with his public relations director, the mayor has decided to change his image. He is going to organise, sponsor and compete in... Byteland's first bicycle race! Quite naturally, the only relevant part of the race is the media coverage of the mayor; everything else is to be done at minimum cost. The street-map of Byteland consists of a not necessarily planar system of bi-directional street segments connecting intersections, in such a way that between 0 and 4 street segments meet at an intersection. The cyclists are to ride round and round a simple loop (a fixed, closed route consisting of several street segments, such that a cyclist goes along a street and through an intersection exactly once in each round). For innumerable reasons (not so difficult to guess at) the mayor would like to choose the shortest possible route for the race (in the sense of total street length). Help him determine the length of such a loop, and tell him how many different shortest loops he can choose from when organising the race.

Input

The input starts with a line containing a single integer $t \leq 200$, the number of test cases. t test cases follow.

Each test case begins with a line with two integers n m , denoting the number of intersections and the number of streets in Byteland, respectively ($1 \leq n \leq 1000$). m lines follow, each containing three integers u_i v_i d_i , denoting the end points and the length of the i -th street segment, respectively ($1 \leq u_i \leq v_i \leq n$, $1 \leq d_i \leq 10^6$).

Output

For each test case output a single line containing exactly two space separated non-negative integers d c - the length of the shortest possible race loop, and the number of routes of this length in the graph. Output 0 0 if the race cannot be held.

Example

Input:

```
2
3 2
1 2 1
1 3 2
4 6
1 2 5
1 4 5
2 3 4
2 4 5
3 4 5
3 1 5
```

Output:

0 0

14 2