During the Three-Kingdom period, there was a general named Xun Lu who belonged to Kingdom Wu. Once his troop were chasing Bei Liu, he was stuck in the Ba Gua Zhen from Liang Zhuge. The puzzle could be considered as an undirected graph with N vertexes and M edges. Each edge in the puzzle connected two vertexes which were u_i and v_i with a length of w_i. Liang Zhuge had great interests in the beauty of his puzzle, so there were no self-loops and between each pair of vertexes, there would be at most one edge in the puzzle. And it was also guaranteed that there was at least one path to go between each pair of vertexes.

Fortunately, there was an old man named Chengyan Huang who was willing to help Xun Lu to hack the puzzle. Chengyan told Xun Lu that he had to choose a vertex as the start point, then walk through some of the edges and return to the start point at last. During his walk, he could go through some edges any times. Since Liang Zhuge had some mysterious magic, Xun Lu could hack the puzzle if and only if he could find such a path with the maximum `XOR` sum of all the edges length he has passed. If the he passed some edge multiple times, the length would also be calculated by multiple times. Now, could you tell Xun Lu which is the maximum `XOR` circuit path in this puzzle to help him hack the puzzle?

Input

The first line of the input gives the number of test cases, T ($1 \leq T \leq 30$). T test cases follow.

Each test case begins with two integers N ($2 \leq N \leq 5 \times 10^4$) and M ($1 \leq M \leq 10^5$) in one line. Then M lines follow. Each line contains three integers u_i, v_i and w_i ($1 \leq u_i, v_i \leq N$, $0 \leq w_i \leq 2^{60} - 1$) to describe all the edges in the puzzle.

Output

For each test case, output one line containing '`Case #`x`:` y', where x is the test case number (starting from 1) and y is the maximum `XOR` sum of one circuit path in the puzzle.

Note: A `XOR` takes two bit patterns of equal length and performs the logical exclusive `OR` operation on each pair of corresponding bits. The result in each position is 1 if only the first bit is 1 or only the second bit is 1, but will be 0 if both are 0 or both are 1. In this we perform the comparison of two bits, being 1 if the two bits are different, and 0 if they are the same.

Sample Input

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

Sample Output

```
Case #1: 3
Case #2: 3
```