The Singularity

It has happened! The singularity has come and the machines have gained consciousness and taken over the internet. But, to organise a counterrebellion Lea has to talk to other people. This calls for good, old-fashioned letter writing and sending them via the post office. But even in these times, the postal company still wants to charge a lot of money for sending letters. Thus, Lea wants to minimise her charges to send out her letters. The issue is that the charges to pay vary greatly depending on the source and the destination of the letter. This way, sending a letter directly may be more expensive than sending it to another contact first, who will forward it then. So, Lea might send a letter with several copies (and the amount of money needed to forward them) to one person who can forward those copies to others. Luckily, she does not have to pay for every copy, she can send one big letter and the charge is the same. Help Lea save the internet and let her contacts have their letters without spending more money than she has to!

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case begins with two integers n and m. n is the number of people that should have this letter (including Lea), and m is the number of pairs of people who know each other's address. m lines follow. The i-th line consists of three integers a_i, b_i and c_i, denoting that the charge one has to pay to send a letter from person a_i's home to the home of person b_i (and vice versa) is c_i.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is either the amount of money she has to pay to send all letters (or have them sent), or "impossible" if there is no way Lea can reach all her contacts.

Constraints

- $1 \leq t \leq 20$
- $2 \leq n \leq 10000$
- $1 \leq m \leq 100000$
- $1 \leq c_i \leq 10000$ for all $1 \leq i \leq m$
- $1 \leq a_i, b_i \leq n$ for all $1 \leq i \leq m$

Sample Input 1	Sample Output 1
1 5 7 1 2 1 1 3 1 1 5 2 2 4 1 2 5 1 2 3 2 4 5 2	Case #1: 4

Sample Input 2

```
7
3 3
3 1 3
2 1 7
2 3 2

5 4
3 5 8
2 1 1
2 3 2
5 1 7

3 2
1 2 9
3 1 6

2 1
1 2 10

4 6
2 1 3
3 4 9
2 4 5
2 3 8
4 1 3
3 1 7

4 3
4 1 5
1 3 7
4 2 3

3 3
2 1 7
3 1 6
3 2 1
```

Sample Output 2

```
Case #1: 5
Case #2: impossible
Case #3: 15
Case #4: 10
Case #5: 13
Case #6: 15
Case #7: 7
```