4409   Ironman Race in Treeland

In Treeland, towns are connected as a tree, by two-way roads. That means, between each pair of different towns u and v, there is exactly one way to travel from u to v (and vice versa), if you don't travel along a road twice.

   The King of Treeland is a sports fan. He wants to hold an Ironman Race. The race track will be the unique simple path between two towns. During the race, civil vehicles **cannot** travel along roads which are part of the track. This definitely damages Treeland's economy. Precisely, every road is assigned to an estimate of damage value. The King does not want his Ironman Race to do too much damage, so the total damage value of the race should not exceed some predefined value, m. But recall that the King is a sports fan. In order to make the race as exciting as possible, the total length of the track should be maximized without exceeding the damage value.
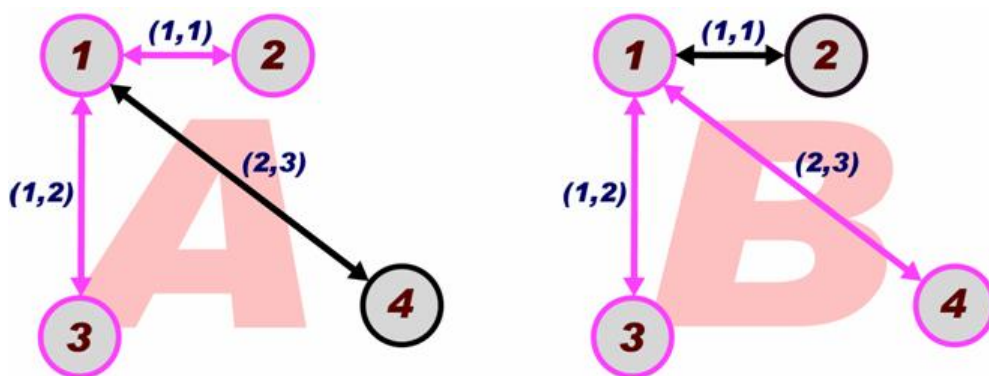

Figure: Figure A corresponds to first sample input and figure B corresponds to second sample input. The race track is marked with pink color. The track and place not included in the race track is colored black.

   Write a program to compute the maximal total length of the track, under the condition that the total damage does not exceed m.

Input

The first line of input contains one integer T ($1 \le T \le 10$), the number of cases followed. Each case begins with two integers n and m ($1 \le n \le 30000$, $1 \le m \le 100000000$), the number of towns and the maximal damage. In the next $n - 1$ lines, each line describes a road by four integers a, b, D, L, that means town a and town b are connected by a road with damage D and length L ($1 \le a, b \le n$, $1 \le D, L \le 1000$). It is guaranteed that the towns will be connected as a tree and there will be no such input for which a race track cannot be formed.

Output

For each case, print the case number and the maximal length. Look at the output for sample input for details.

Sample Input

2

```
4 2
1 2 1 1
1 3 1 2
1 4 2 3
4 3
1 2 1 1
1 3 1 2
1 4 2 3
```

Sample Output

```
Case 1: 3
Case 2: 5
```