

Rapid Rerouting

Filename: reroute

The Problem

While driving to EA Sports last week, our bus driver apparently had the wrong address. He took us on the shortest route from the school to where he thought we were going, only to find out we had not actually made it to EA! From the wrong address, we then had to find a new route to EA Sports (thankfully Arup knew where to go). We then took the shortest route from the wrong address to EA Sports.

Naturally, had we not gone to the wrong address, we would have had a shorter trip. Now you start to wonder just how much our trip was extended due to going to the wrong address to start with. You decide to write a program to figure out how much shorter the direct route would have been compared with going to the wrong address first and then going to our destination.

For this problem we will simply model each major intersection as a single point and describe the roads connecting each major intersection.

The Input

There will be several test cases. Input will begin with a single positive integer T giving the number of test cases. Each test case will begin with two positive integers N ($3 \leq N \leq 100$) and R ($1 \leq R \leq N*(N-1)/2$) indicating the number of intersections and roads, respectively. This will be followed by exactly R lines each containing three integers a b c indicating there is a road between intersection a and b ($1 \leq a, b \leq N$) which takes c ($1 \leq c \leq 1000$) minutes to travel. All roads are bidirectional, and no road will occur more than once in the input. The graph is guaranteed to be connected. The last line of input for each case will contain three distinct integers s , d , and w ($1 \leq s, d, w \leq N$), indicating the start intersection, the destination intersection, and the intersection we went to first instead of the of the destination, respectively.

The Output

For each case, output a single line

Case #X: Took Y minutes, should have taken Z minutes

where X is the case number beginning with 1, Y is how long our route took, and Z is how long it would have taken if we had gone straight to the destination.

Sample Input

```
2
5 6
1 2 6
1 3 40
2 3 25
2 5 20
3 4 3
4 5 15
1 4 5
5 6
1 2 6
1 3 40
2 3 25
2 5 20
3 4 3
4 5 15
1 4 3
```

Sample Output

```
Case #1: Took 41 minutes, should have taken 34 minutes
Case #2: Took 34 minutes, should have taken 34 minutes
```