

# Wandern

Lea ist gerne draußen in der Natur, fast jedes Wochenende geht sie wandern. Auch heute ist sie extra ganz früh aufgestanden, an den Fuß eines Berges gefahren und hat pünktlich zum Mittagessen den Gipfel erreicht. Während ihres Desserts (“Apfelstrudel”) fällt ihr plötzlich ein: sie hat am Nachmittag noch einen unglaublich wichtigen Termin. In größter Eile schaut sie auf die Wanderkarte, um den schnellsten Weg zu ihrem Auto zu finden. Dabei ist es nicht förderlich, dass auf der Karte hunderte verschiedene Wege verzeichnet sind, die sich auch noch ständig überschneiden und so tausende mögliche Routen ins Tal ergeben. Zum Glück hat sie ihr Satellitentelefon dabei und kann dich anrufen. Nachdem sie dir kurz alle möglichen Wege beschrieben hat, will sie von dir wissen, wie lange der Weg zu ihrem Auto dauern wird.

## Eingabe

Die erste Zeile der Eingabe enthält eine Ganzzahl  $t$ . Darauf folgen, jeweils durch eine Leerzeile getrennt,  $t$  Testfälle.

Jeder Testfall beginnt mit zwei Ganzzahlen  $n$  und  $m$ , wobei  $n$  die Zahl der Wegkreuzungen (diese sind von 1 bis  $n$  nummeriert) und  $m$  die Zahl der Wanderwege ist. Weitere  $m$  Zeilen folgen. Die  $i$ -te davon enthält drei Ganzzahlen  $v_i$ ,  $w_i$  und  $c_i$ , wovon  $v_i$  und  $w_i$  zwei Wegkreuzungen angeben, zwischen denen ein Wanderweg verläuft, und  $c_i$  die Dauer in Minuten ist, die Lea für diesen Weg benötigt. Alle Wanderwege sind in beide Richtungen begehbar, zwei Wegkreuzungen können jedoch durch mehrere Wanderwege verbunden sein. Lea befindet sich gerade an Wegkreuzung Nummer 1, ihr Auto hat sie an Wegkreuzung  $n$  geparkt.

## Ausgabe

Gibt für jeden Testfall eine Zeile “Case # $i$ :  $d$ ” aus, wobei  $i$  bei 1 beginnend die Nummer des Testfalls und  $d$  die Länge der kürzesten Route von Wegkreuzung 1 zu Wegkreuzung  $n$  in Minuten ist.

## Beschränkungen

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$
- $1 \leq m \leq 50000$
- $1 \leq c_i \leq 1000$  für alle  $1 \leq i \leq m$
- $1 \leq v_i, w_i \leq n$  für alle  $1 \leq i \leq m$

### Sample Input 1

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

3 3
1 2 1
1 3 3
2 3 1

3 2
1 2 1
2 3 2

3 2
1 3 5
2 3 4
```

### Sample Output 1

```
Case #1: 3
Case #2: 2
Case #3: 3
Case #4: 5
```

**Sample Input 2**

```
7
4 4
1 3 2
1 4 2
2 4 5
3 4 3

5 4
1 3 2
1 4 1
2 5 5
3 2 4

5 5
1 3 2
1 5 2
2 4 5
2 5 2
4 5 2

3 2
1 2 1
2 3 5

2 1
1 2 3

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

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

**Sample Output 2**

```
Case #1: 2
Case #2: 11
Case #3: 2
Case #4: 6
Case #5: 3
Case #6: 1
Case #7: 8
```

# Hiking

Lea enjoys nature a lot, therefore she often goes hiking at the weekend. Last Sunday, she got up early, drove to the foot of a mountain and reached the top just at the right time for lunch. During her dessert, “Apfelstrudel”, she suddenly remembered: She had a very important appointment this afternoon which she was about to miss. In a big hurry, she looked at the map to figure out the fastest way to her car. To her amazement, there were hundreds of hiking trails which crossed multiple times forming thousands of possible routes down! Lea was helpless. Luckily she had her satellite phone with her and called... you! After giving you the list of all the trails she wants to know how far away from her car she currently is. Help her out!

## 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 starts with two integers  $n$  and  $m$ , where  $n$  is the number of intersections (those are numbered from 1 to  $n$ ) and  $m$  is the number of hiking trails.  $m$  lines follow. The  $i$ -th of those lines contains three integers  $v_i$ ,  $w_i$  and  $c_i$ , meaning that intersections  $v_i$  and  $w_i$  are connected by a hiking trail of length  $c_i$ . Hiking trails are undirected, but a pair of intersections may be connected by multiple hiking trails. Lea is currently at intersection 1, her car at intersection  $n$ .

## Output

For each test case, output one line containing “Case # $i$ :  $d$ ” where  $i$  is its number, starting at 1, and  $d$  is the shortest distance of intersection 1 to intersection  $n$ . Each line of the output should end with a line break.

## Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$
- $1 \leq m \leq 50000$
- $1 \leq c_i \leq 1000$  for all  $1 \leq i \leq m$
- $1 \leq v_i, w_i \leq n$  for all  $1 \leq i \leq m$

### Sample Input 1

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

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

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

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

### Sample Output 1

```
Case #1: 3
Case #2: 2
Case #3: 3
Case #4: 5
```

**Sample Input 2**

```
7
4 4
1 3 2
1 4 2
2 4 5
3 4 3

5 4
1 3 2
1 4 1
2 5 5
3 2 4

5 5
1 3 2
1 5 2
2 4 5
2 5 2
4 5 2

3 2
1 2 1
2 3 5

2 1
1 2 3

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

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

**Sample Output 2**

```
Case #1: 2
Case #2: 11
Case #3: 2
Case #4: 6
Case #5: 3
Case #6: 1
Case #7: 8
```