

Graph Theory - S008 (E044)Solved Challenges **0/1**[Back To Challenges List](#)**Single Source Shortest Path****ID:11151 Solved By 458 Users**

There are **N** cities in a country which are numbered from 1 to N. The N cities are connected by **L** links. Each link contains the **source** city, the **destination** city and the **distance** between them. The program must accept the values of N and L links as the input. The program must print the shortest distance from the city 1 to all the other cities as the output.

Boundary Condition(s): $2 \leq N \leq 100$ $1 \leq L \leq 1000$ $1 \leq \text{Distance between any two cities} \leq 10^5$ **Input Format:**

The first line contains N and L separated by a space.

The next L lines, each containing three integers representing the source city, the destination city and the distance between them.

Output Format:

The first line contains N-1 integers representing the shortest distance from the city 1 to all the other cities.

Example Input/Output 1:

Input:

```
6 7
1 2 20
1 6 5
6 5 2
5 4 3
4 3 2
5 2 10
3 2 2
```

Output:

```
14 12 10 7 5
```

Explanation:

The shortest distance from the city **1** to **2** is **14** (1 -> 6 -> 5 -> 4 -> 3 -> 2).

The shortest distance from the city **1** to **3** is **12** (1 -> 6 -> 5 -> 4 -> 3).

The shortest distance from the city **1** to **4** is **10** (1 -> 6 -> 5 -> 4).

The shortest distance from the city **1** to **5** is **7** (1 -> 6 -> 5).

The shortest distance from the city **1** to **6** is **5** (1 -> 6).

Example Input/Output 2:

Input:

```
5 7
1 2 10
2 3 50
```

3 5 10
4 5 60
1 5 100
1 4 30
4 3 20

Output:

10 50 30 60

Max Execution Time Limit: 500 millisecs

Ambiance

Python3 (3.x) ▼



Reset

```
1 class Link:
2     def __init__(self,source,dest,dist):
3         self.source = source
4         self.dest = dest
5         self.dist = dist
6
7 N,L = map(int,input().strip().split())
8
9 lists = []
10 for city in range(L):
11     source,dest,dist = map(int,input().strip().split())
12     lists.append(Link(source,dest,dist))
13
14 shortest = [None for i in range(N+1)]
15 relaxed = True
16 shortest[1] = 0
17
18
19 for iter in range(1,N):
20     if(relaxed):
21         relaxed = False
22         for link in lists:
23             if(shortest[link.source] == None):
24                 continue
25             if(shortest[link.dest] == None or shortest[link
26                 .source]+link.dist < shortest[link.dest]):
27                 shortest[link.dest] = shortest[link.source] +
28                     link.dist
29                 relaxed = True
30
31 else:
32     break
33
34 for city in range(2,N+1):
35     print(shortest[city],end=" ")
36
```

Code did not pass the execution



TestCase ID: 64285

Input:

```
6 7
1 2 20
1 6 5
6 5 2
5 4 3
4 3 2
5 2 10
3 2 2
```

Expected Output:

```
14 12 10 7 5
```

Your Program Output:

```
14 12 10 7
```

Save

Run

☐ Run with a custom test case (Input/Output)