

题目描述：

某通信网络中有  $N$  个网络结点，用 1 到  $N$  进行标识。网络通过一个有向无环图表示，其中图的边的值表示结点之间的消息传递时延。

现给定相连节点之间的时延列表  $\text{times}[i]=\{u,v,w\}$ ，其中  $u$  表示源结点， $v$  表示目的结点， $w$  表示  $u$  和  $v$  之间的消息传递时延。请计算给定源结点到目的结点的最小传输时延，如果目的结点不可达，返回-1。

注：

1、 $N$  的取值范围为[1,100]；

2、时延列表  $\text{times}$  的长度不超过 6000，且  $1 \leq u,v \leq N, 0 \leq w \leq 100$ ；

输入描述：

输入的第一行为两个正整数，分别表示网络结点的个数  $N$ ，以及时延列表的长度  $M$ ，用空格分隔；

接下来的  $M$  行为两个结点间的时延列表  $[u \ v \ w]$ ；

输入的最后一行为两个正整数  $u$  和  $v$ ，分别表示源结点和目的结点；

输出描述：

输出一个整数，表示源结点到目的结点的最小时延。

补充说明：

```
示例1
输入：3 3
      1 2 11
      2 3 13
      1 3 50
      1 3
输出：24
说明：1->3的时延是50，1->2->3时延是11+13=24，所以1到3的最小时延是24；
```

```

1  n, m = map(int, input().split())
2  arr = [list(map(int, input().split())) for _ in range(m)]
3  start, end = map(int, input().split())
4
5
6  def get_min_times(matrix, e_node):
7      for i in range(n):
8          min_times_path = matrix[i][e_node]
9          for j in range(n):
10             if matrix[i][j] != -1 and matrix[j][e_node] != -1:
11                 min_times_path = min(min_times_path, matrix[i][j] + matrix[j][e_node])
12             matrix[i][e_node] = min_times_path
13
14
15  def get_result():
16      matrix = []
17      for i in range(n):
18          matrix.append([])
19          for j in range(n):
20              matrix[i].append(600000)
21
22      for u, v, w in arr:
23          matrix[u - 1][v - 1] = w
24
25      for i in range(n):
26          get_min_times(matrix, end - 1)
27
28      result = -1 if matrix[start - 1][end - 1] == 600000 else matrix[start - 1][end - 1]
29      print(result)
30
31
32  get_result()

```