

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 $times[i]=\{u,v,w\}$ ，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 $times$ 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。

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 $times[i]=\{u,v,w\}$ ，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 $times$ 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。

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;

```
N, M = map(int, input().split(" "))
```

```
graph = []
```

```
for _ in range(M):
```

```
    # ints=[int(x) for x in input().split()]
```

```
    ints = list(map(int, input().split(" ")))
```

```
    graph.append(ints)
```

```
start, end = map(int, input().split(" "))
```

```
res = []
```

```
stack = [(start, 0)]
```

```
while stack:
    node, count = stack.pop()
    for temp in graph:
        if temp[0] == node:
            if temp[1] == end:
                res.append(count + temp[2])
            else:
                stack.append((temp[1], count + temp[2]))

if len(res) == 0:
    print(-1)
else:
    print(min(res))
```