Pipes

Problem Statement:

There are n stations and m pipes, the start station has id 0, and end station has id n-1. Stations are connected by water pipes, and each pipe has a capacity c, denoting the maximum amount of water it can hold at any time. The stations and pipes form a connected graph.

Mario the plumber has to redirect water to flow from the start station to the end station in a single path. Mario wants to ensure that the maximum capacity path is taken, which means that the minimum capacity of any pipe along the path is maximised. This path is guaranteed to be unique. He also needs to block off all the the pipes that have any endpoint on the path, but are not part of the path so that water does not flow anywhere else. Help him find out which pipes he needs to block off!

Input:

On the first line, 2 space separated integers, n and m (1 <= n, m <= 1000)

The next m lines contain 3 space separated integers a, b and c, where a and b are the 2 station ids that are connected by the pipe, and c is the capacity of the pipe. The pipes are labelled 0 to m-1 in the given order. (0 <= a, b <= n - 1) (1 <= c <= 500000)

Output:

In a single line, all the pipe ids that need to be blocked, sorted in increasing order. If no pipes need to be blocked, output “none” in a single line.

Sample Input 1:

```
7 10
0 1 400
1 2 150
2 3 37
3 4 40
4 5 25
4 6 50
6 1 17
0 6 5
0 2 60
0 3 50
```

Sample Output 1:

```
0 2 4 6 7 8
```

Sample Input 2:

```
4 3
0 1 5
1 2 10
2 3 15
```

Sample Output 2:

```
none
```

Explanation:

For Sample Input 1, the maximum capacity path is from station 0 -> 3 -> 4 -> 6, with a path capacity of 40. The other pipes along this path will be blocked off, which are pipes 0, 2, 4, 6, 7, 8.

For Sample Input 2, no pipes need to be blocked off.