Out-Arborescence

Description

Given a **directd** graph $G = (V, E)$ with n nodes and m edges, you are required to find a node v and some edges $I \subseteq E$ such that 1) for any node $u \in V$, there is at least a path from v to u, and 2) the total weights of I is minimized.

Input Format

The case have two integers n and m. **Note that the nodes are numbered 0... n-1**. For the following m lines, each line contain three integers s, t and C, meaning from s to t there is a edge with weight C.

Output Format

If no such a node v that can visit every nodes, output "impossible".

Otherwise, print the minimum total weights of I and the number of v. If there exists multiple satisfied v, choose the one with the minimum number.

Sample

Sample Input

```
4 4
0 1 10
0 2 10
1 3 20
2 3 30
```

Sample Output

```
40 0
```

For 100% testcases: $1 \leq n \leq 1000, 1 \leq m \leq 10000, 1 \leq C \leq 1000$.