Weighted shortest path (4) P39586_en

Write a program that, given a directed graph with positive costs at the arcs, and two vertices x and y, computes the minimum cost to go from x to y, and the number of ways of going from x to y with such minimum cost.

Input

Input consists of several cases. Every case begins with the number of vertices n and the number of arcs m. Follow m triples u, v, c, indicating that there is an arc $u \to v$ of cost c, where $u \neq v$ and $1 \leq c \leq 10^4$. Finally, we have x and y. Assume $1 \leq n \leq 10^4$, $0 \leq m \leq 5n$, and that for every pair of vertices u and v there is at most one arc of the kind $u \to v$. All numbers are integers. Vertices are numbered from 0 to $n - 1$.

The condition for c was previously $c \leq 1000$. It was updated to create new test cases.

Output

For every case, print the minimum cost to go from x to y, and the number of different paths that achieve this cost. This number will never exceed 10^9. If there is no path from x to y, state so.

Sample input 1

```
6 10
1 0 6
1 5 15
3 4 3
3 1 8
4 0 20
0 5 5
0 2 1
5 1 10
4 1 2
2 3 4
3 5

2 1
0 1 1000
1 0

3 3
0 2 100
0 1 40
1 2 60
0 2
```

Sample output 1

```
cost 16, 1 way(s)
no path from 1 to 0
cost 100, 2 way(s)
```

Problem information

Author: Salvador Roura

Generation: 2026-01-25T10:41:09.684Z