Graph

SEASON 7 – ROUND SIX

Lora has a graph with **N** vertices and **M** directed weighted edges. We denote by w_{AB} the weight of the edge from A to B if it exists. Lora now wants to change the weights of the existing edges in the following way:

- For every vertex *i* Lora chooses an integer F_i
- For every edge from vertex A to vertex B its new weight is equal to $w_{AB} + F_A – F_B$

After the change Lora wants the edge with the maximal weight in the resulting graph to be as small as possible. Help her by calculating the minimum possible value of the maximum edge in the resulting graph after an optimal change, or determine that the required value can be arbitrarily small.

Input

The first line of the input file `graph.in` contains two integers **N** and **M** – the number of vertices and the number of edges.

Each of the next M lines contains three space-separated integers **a**, **b**, **c** denoting a directed edge from vertex **a** to vertex **b** with weight **c**.

Output

On a single line in the output file `graph.out` print one number – the smallest possible weight of the edge with maximal weight after an optimal change. **If this value can be arbitrarily small, print "-inf" without the quotes.**

Constraints

$1 \leq N, M \leq 1\ 000$

$-10^9 \leq w_{ij} \leq 10^9$

Time limit: 0.6 sec

Memory limit: 256 MB

Sample test

Input (towers.in)	Output (towers.out)
3 3 1 2 1 2 3 2 3 1 3	2
2 1 1 2 -3	-inf

Clarifications

In the first sample one optimal solution is for Lora to choose $F_1 = 1$; $F_2 = 0$ and $F_3 = 0$. We get:

$w_{12} + F_1 – F_2 = 2$

$w_{23} + F_2 – F_3 = 2$

$w_{31} + F_3 – F_1 = 2$

The maximal weight is 2. There are no values for F that give a maximal weight of less than 2.

In the second sample case it is possible to get an arbitrarily small weight for the edge, since Lora may choose arbitrarily small value for F_1 and arbitrarily large value for F_2.