High-Speed Railroad (railroad)

There are N cities and M bidirectional railways connecting cities a_i and b_i with a travel time of c_i minutes. There is also a train that connects city 0 with city $N-1$, this train will always take the **only** shortest path between the two cities.

However, the train only passes through some cities and the mayors of the cities it doesn't pass through are angry.

To please the mayors, you can upgrade a **single** railway to high-speed railway, which will reduce the travel time by 1 minute for each euro spent. Of course, the travel time must be **strictly positive** after the upgrade.

You want to upgrade a railway so that the new shortest path between city 0 and city $N-1$ **always** passes to at least a city that wasn't passed by the train before. In the new railway network there may be many shortest paths, however the original one must not be one of them.

Figure 1: Italian high-speed train Frecciarossa.

What is the minimum amount of money you need to spend to achieve this?

☞ Among the attachments of this task you may find a template file `railroad.*` with a sample incomplete implementation.

Input

The first line contains the integers N and M, the number of cities and the number of railways.

The following M lines contains three integers a_i, b_i and c_i representing a bidirectional railway between a_i and b_i with a travel time of c_i minutes.

Output

You need to write a single line with an integer: the minimum amount of money you need to spend so that the new shortest path between city 0 and city $N-1$ always passes through a new city. If it is not possible to achieve this, you need to write -1.

Constraints

- $2 \leq N \leq 100\,000$.
- $1 \leq M \leq 300\,000$.
- $0 \leq a_i, b_i \leq N-1$ for each railway.
- $1 \leq c_i \leq 10^9$ for each railway.
- The shortest path between 0 and $N-1$ is unique.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (35 points) $N \leq 1000$, $M \leq 2000$.

– **Subtask 3** (30 points) $N = M$, and each city has exactly two railways.

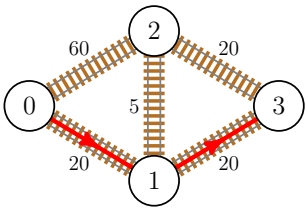
– **Subtask 4** (35 points) No additional limitations.

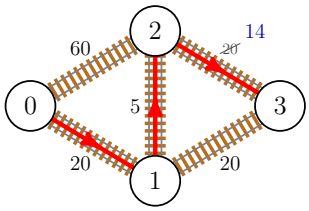
Examples

input	output
4 5 0 1 20 0 2 60 1 2 5 2 3 20 1 3 20	6
4 5 0 1 5 0 2 10 1 2 10 2 3 10 1 3 5	-1

Explanation

In the **first sample case**, it is possible to upgrade the railway between cities 2 and 3 so that the new shortest path passes through city 2.





In the **second sample case**, it is not possible to upgrade a railway so that the new shortest path passes through a new city.