International Olympiad in Informatics 2020
Practice Session
March 28, 2020
Singapore, Singapore
Session 7 Tasks

`trackingtraining`
English (US)

Tracking Training

There are n stations and m one-way tracks. The i^{th} of these tracks allows a train to travel from station a_i to station b_i, and takes c_i seconds to travel along.

There might be one track connecting x to y and another connecting y to x; for each x, there might also be (at most one) track connecting x to itself.

Suppose a train leaves the i^{th} station, takes some tracks nonstop (possibly the same track multiple times, visiting some stations multiple times, or even passing through the i^{th} station several times along the way) and then ends its journey at the i^{th} station, taking a total of t seconds.

You want to set up a camera at the i^{th} station such that it will take a photo of the train exactly as it ends its journey at time t. Unfortunately, you do not know what tracks the train will travel, and consequently you do not know what t is. Hence, you decide to set up your camera to take a photo every p_i seconds, such that no matter what t is, your camera will definitely take a photo of the train when it ends its journey. Since memory is expensive, you want p_i to be as large as possible (that is, it waits longer between photos and hence takes fewer shots).

If a train can never return to end its journey at the i^{th} station (either because there is no way to return no matter which track it takes first, or there were no outgoing tracks in the first place), we define p_i to be -1.

You want to track the trains, but you also want to train to get on track to win a gold medal at the International Olympiad in Informatics. Determine p_i for all i from 1 to n.

Input format

The first line of input contains two integers n and m, the number of stations and the number of tracks.

The next m lines describe the tracks. The i^{th} line among these contains three integers a_i, b_i and c_i, denoting that a one-way track connects station a_i to station b_i and takes c_i seconds to travel along.

Output format

Output n lines. The i^{th} line should contain the value p_i.

Subtasks

In all subtasks $1 \leq n, m$, $1 \leq a_i, b_i \leq n$, $1 \leq c_i \leq 10\,000$.

Subtask	Points	n	m	Additional Constraints
1	12	$n \leq 2$	$m \leq 20$	
2	12	$n \leq 2$	$m \leq 10\,000$	
3	32	$n \leq 100\,000$	$m \leq 100\,000$	$c_i = 1$
4	12	$n \leq 10\,000$	$m \leq 10\,000$	
5	32	$n \leq 100\,000$	$m \leq 100\,000$	

Example

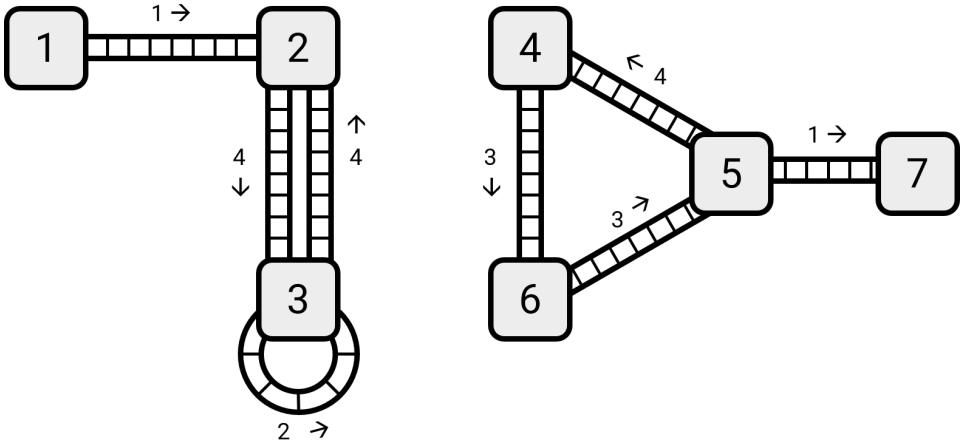
Consider the following input:

```
7 8
1 2 1
2 3 4
3 2 4
4 6 3
6 5 3
3 3 2
5 4 4
5 7 1
```

The correct output is:

```
-1
2
2
10
10
10
-1
```

In this case, we have the following stations and tracks:



Note that $p_1 = -1$, as it is impossible to return to the 1^{st} station after leaving.

Note that $p_2 = 2$, because it is possible for the train to return after 8 seconds ($2 \to 3 \to 2$), after 10 seconds ($2 \to 3 \to 3 \to 2$), after 12 seconds ($2 \to 3 \to 3 \to 3 \to 2$), and so on. The only way to ensure that regardless of which tracks are taken, we will take a photo of the train, is to take a photo either every 1 second or every 2 seconds. 2 is superior, so $p_2 = 2$.