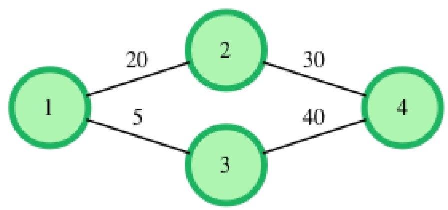
Jack goes to Rapture

Jack has just moved to a new city called Rapture. He wants to use the public public transport system. The fare rules are as follows:

1. Each pair of connected stations has a fare assigned to it regardless of direction of travel.
2. If a passenger travels from station A to station B, he only has to pay the difference between the fare from A to B and the cumulative fare that he has paid to reach station A [*fare(A,B) - total fare to reach station A*]. If the difference is negative, he can travel free of cost from A to B.

Jack is low on cash and needs your help to figure out the most cost efficient way to go from the first station to the last station. Given the number of stations g_nodes (numbered from 1 to g_nodes), and the fare between the g_edges pairs of stations that are connected, determine the lowest fare from station 1 to station g_nodes.

For example, there are $g\_nodes = 4$ stations with undirected connections at the costs indicated:



Travel from station $1 \to 2 \to 4$ costs 20 for the first segment ($1 \to 2$) then the cost differential, an additional $30 - 20 = 10$ for the remainder. The total cost is 30. Travel from station $1 \to 3 \to 4$ costs 5 for the first segment, then an additional $40 - 5 = 35$ for the remainder, a total cost of 40. The lower priced option costs 30.

Complete the program in the editor below. It should print the cost of the lowest priced route from station 1 to station g_nodes.

The program reads and supplies the following parameters:

- g_nodes: an integer that represents the number of stations in the network
- g_edges: an integer that represents the number of connections between those stations
- g_from: an array of integers where each $g\_from[i]$ is an end station for a connection
- g_to: an array of integers where each $g\_to[i]$ is the other end of a connection with station $g\_from[i]$
- g_weight: an array of integers where each $g\_weight[i]$ is the cost of traveling either direction between stations $g\_from[i]$ and $g\_to[i]$

Input Format

The first line contains two space-separated integers, g_nodes and g_edges, the number of stations and the number of connections between them.
Each of the next g_edges lines contains three space-separated integers, g_from, g_to and g_weight, the starting and ending stations that are connected and the fare between them.

Constraints

- $1 \leq g\_nodes \leq 50000$
- $1 \leq g\_edges \leq 500000$
- $1 \leq g\_weight[i] \leq 10^7$

Output Format

The minimum fare to be paid to reach station g_nodes from station 1. If the station g_nodes cannot be reached from station 1, print `NO PATH EXISTS`

Sample Input 0

```
5 5
1 2 60
3 5 70
1 4 120
4 5 150
2 3 80
```

Sample Output 0

```
80
```

Explanation 0

There are two ways to go from first station to last station.

- $1 \to 2 \to 3 \to 5$
- $1 \to 4 \to 5$

For the first path, Jack first pays 60 units of fare to go from station 1 to 2. Next, Jack has to pay $80 - 60 = 20$ units to go from 2 to 34. Now, to go from 3 to 5, Jack has to pay $70 - (60 + 20) = -10$ units, but since this is a negative value, Jack pays 0 units to go from 3 to 5. Thus the total cost of this path is $(60 + 20) = 80$ units.

For the second path, Jack pays 120 units to reach station 4 from station 1. To go from station 4 to 5, Jack has to pay $150 - 120 = 30$ units. Thus the total cost becomes $(120 + 30) = 150$ units. So, the first path is the most cost efficient, with a cost of 80.

Sample Input 1

```
5 6
1 2 30
2 3 50
3 4 70
4 5 90
1 3 70
3 5 85
```

Sample Output 1

```
85
```

Explanation 1

Travel starts at node 1 and there are two paths to node 3 that cost either 50 or 70. Taking the route from 3 through 4 to 5 brings the cost up to 90, while going directly from 3 to 5 costs only 85.