# Jack goes to Rapture

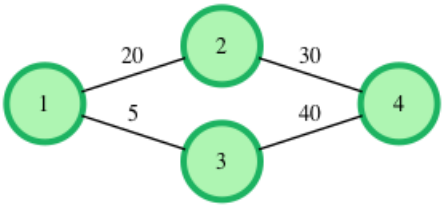
Jack has just moved to a new city called Rapture. He wants to use the public public transport system. The fare rules are as follows:

1. Each pair of connected stations has a fare assigned to it regardless of direction of travel.
2. If Jack 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 paid to reach station A), [*fare(A,B) - total fare to reach station A*]. If the difference is negative, travel is 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 fares (weights) between the $g\_edges$ pairs of stations that are connected, determine the lowest fare from station $1$ to station $g\_nodes$.

**Example**
$g\_nodes = 4$
$g\_from = [1, 1, 2, 3]$
$g\_to = [2, 3, 4, 4]$
$g\_weight = [20, 5, 30, 40]$

The graph looks like this:




Travel from station $1 \rightarrow 2 \rightarrow 4$ costs $20$ for the first segment ($1 \rightarrow 2$) then the cost differential, an additional $30 - 20 = 10$ for the remainder. The total cost is $30$.

Travel from station $1 \rightarrow 3 \rightarrow 4$ 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 $30$.

**Function Description**
Complete the *getCost* function in the editor below.

getCost has the following parameters:

- *int g_nodes:* the number of stations in the network
- *int g_from[g_edges]:* end stations of a bidirectional connection
- *int g_to[g_edges]:* $g\_from[i]$ is connected to $g\_to[i]$ at cost $g\_weight[i]$
- *int g_weight[g_edges]:* the cost of travel between associated stations

**Prints**

- *int or string:* the cost of the lowest priced route from station $1$ to station $g\_nodes$ or `NO PATH EXISTS`.

No return value is expected.

**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 connected stations and the fare between them.

**Constraints**

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

**Sample Input**

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

**Sample Output**

```
80
```

**Explanation**

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

- 1 -> 2 -> 3-> 5
- 1 -> 4 -> 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 3. 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.