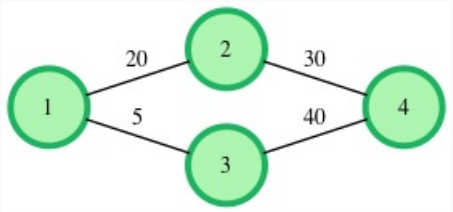
Jack has just moved to a new city called Rapture. He wants to use the public public transport system. The fare rules are as follows:

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 $1 \rightarrow 2 \rightarrow 4$ 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.

Complete the program in the editor below. It should print the cost of the lowest priced route from station 1 to station g_nodes. If there is no route, print `NO PATH EXISTS`.

Function Description
Complete the *getCost* function in the editor below. It should print the cost of the lowest priced route from station 1 to station g_nodes, or if there is no route, print `NO PATH EXISTS`. There is no expected return value from the function.

getCost has the following parameters:

- g_nodes: an integer that represents the number of stations in the network
- g_from: an array of integers that represent end stations of a bidirectional connection
- g_to: an array of integers that represent end stations of a bidirectional connection, where $g\_from[i]$ is connected to $g\_to[i]$ at cost $g\_weight[i]$
- g_weight: an array of integers that represent the cost of travel between associated stations

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`