Getting started with Competitive Programming

Week 6 Programming Problems

Problem 1 – Libraries

Find it here https://www.hackerrank.com/challenges/torque-and-development/problem

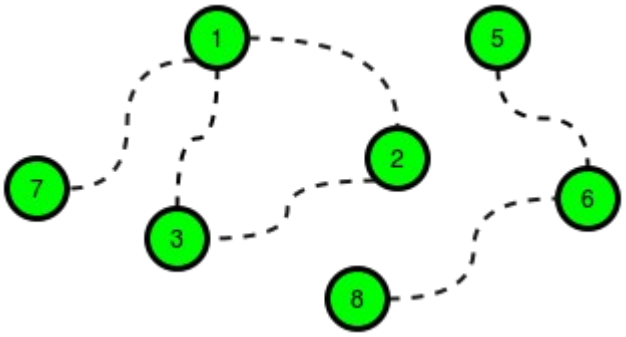
Determine the minimum cost to provide library access to all citizens of WonderLand. There are cities numbered from 1 to n . Currently there are no libraries and the cities are not connected. Bidirectional roads may be built between given city pair listed in cities.

A citizen has access to a library if:

- Their city contains a library.
- They can travel by road from their city to a city containing a library.

Example

The following figure is a sample map of WonderLand where the dotted lines denote possible roads:



The cost of building any road is cc_road = 2, and the cost to build a library in any city is c_lib = 3. Build 5 roads at a cost of 5*2 = 10 and libraries for a cost of 6. One of the available roads in the cycle 1 -> 2 -> 3 -> 1 is not necessary.

There are q queries, where each query consists of a map of WonderLand and value of c_lib and c_road. For each query, find the minimum cost to make libraries accessible to all the citizens.

Input Format

The first line contains a single integer q, that denotes the number of queries.

The subsequent lines describe each query in the following format:
- The first line contains four space-separated integers that describe the respective values of

n, m, c_lib and c_road, the number of cities, number of roads, cost of a library and cost of a road.
- Each of the next lines contains two space-separated integers, u[i] and v[i], that describe a bidirectional road that can be built to connect cities u[i] and v[i].

Constraints

- 1 <= q <= 10

- 1 <= n <= 10^5
- 0 <= m <= min(10^5, n(n-1)/2))
- 1 <= c_road, c_lib <= 10^5
- Each road connects two distinct cities.

Sample Input

```
STDIN       Function
-----       --------
2           q = 2
3 3 2 1     n = 3, cities[] size m = 3, c_lib = 2, c_road = 1
1 2         cities = [[1, 2], [3, 1], [2, 3]]
3 1
2 3
6 6 2 5     n = 6, cities[] size m = 6, c_lib = 2, c_road = 5
1 3         cities = [[1, 3], [3, 4],...]
3 4
2 4
1 2
2 3
5 6
```

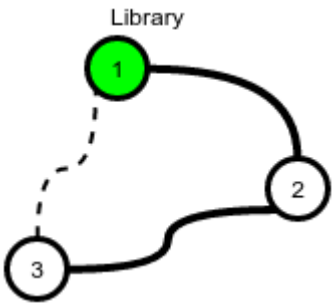
Sample Output

```
4
12
```

Explanation

Perform the following queries:

There are n = 3 cities and they can be connected by m = 3 bidirectional roads. The price of building a library is c_lib = 2 and the price for repairing a road is c_road = 1.

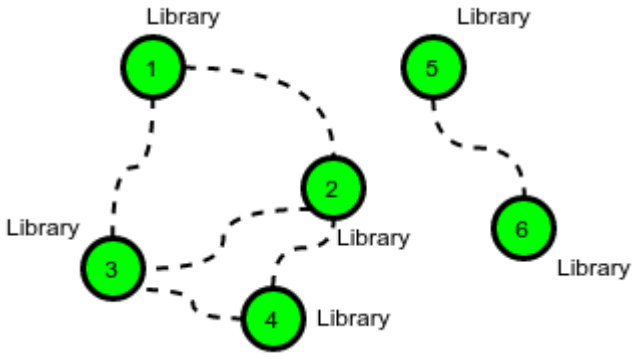


The cheapest way to make libraries accessible to all is to:

- Build a library in city 1 at a cost of x = 2.
- Build the road between cities 1 and 2 at a cost of y = 1
- Build the road between cities 2 and 3 at a cost of y = 1

This gives a total cost of 2 + 1 + 1 = 4. Note that the road between cities 3 and 1 does not need to be built because each is connected to city 2.

- In this scenario it is optimal to build a library in each city because the cost to build a library is less than the cost to build a road.



There are 6 cities, so the total cost is 6 * 2 = 12.

Problem 2 – Rapture

Find it here - https://www.hackerrank.com/challenges/jack-goes-to-rapture/copy-from/242372406

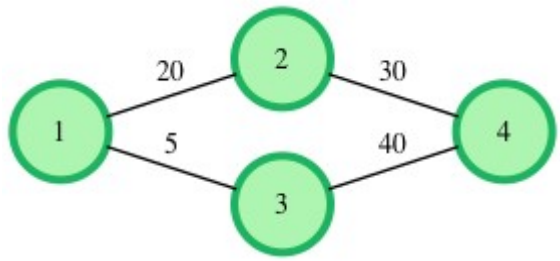
Jack has just moved to a new city called Rapture. He wants to use the public transport system. The fare rules are as follows:

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

The graph looks like this:



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

Travel from station 1->3->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.

Output Format

- *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 lines g_edges contains three space-separated integers, and g_from, g_to and g_weight, the connected stations and the fare between them.

Constraints

- 1 <= g_nodes <= 50000
- 1 <= g_edges <= 500000
- 1 <= g_weight[i] <= 10^7