UNITED INTERNATIONAL UNIVERSITY
Department of Computer Science and Engineering (CSE)

Course Title: Data Structure & Algorithm Lab II **Course Code: CSE2218**

Trimester & Year: Summer 2023 **Section: B** **Credit Hours: 1.0** **AZ**

ASSIGNMENT 02: Greedy Approach

Q1: Kruskal Algorithm Implementation

Implement Kruskal's algorithm using the following pseudocode as discussed in the Class Lecture and also show its simulation (**Deposit the simulation as a separate pdf named *1.pdf***)

MST-KRUSKAL(G, w)

$A = \emptyset$

for each vertex $v \in G.V$

    MAKE-SET(v)

sort the edges of $G.E$ into nondecreasing order by weight w

for each edge $(u, v) \in G.E$, taken in nondecreasing order by weight

    if FIND-SET$(u) \neq$ FIND-SET(v)

        $A = A \cup \{(u, v)\}$

        UNION(u, v)

return A

UNITED INTERNATIONAL UNIVERSITY
Department of Computer Science and Engineering (CSE)

Q2 Course Selection Problem

X-land has a very famous university. The university offers N courses. Each course runs for some consecutive range of days. You are given starting and ending days of the i^{th} course by $start_i$ and end_i, respectively.
Sam wanted to enrol himself in the university. But he is not sure about the exact time for which he wants to study. Though he has Q such tentative plans in his mind. Each plan consists of a start date $plan\_start_j$ and an end date $plan\_end_j$.

Sam wants your help in finding out the maximum number of courses he can complete during each of his plans. Note that at a time, Sam cannot handle multiple courses, i.e. he can attend at most one course during a day. Also, a course will be considered completed only if Sam attends all the classes of the course.

Input

- There is a single test case.
- The first line of the input contains two space-separated integers N and Q denoting the number of courses the university offers and the number of plans Sam has in mind, respectively.
- The i^{th} of the next N lines contains two space-separated integers $start_i$ and end_i denoting the starting and the ending day of the i^{th} course.
- The j^{th} of the next Q lines contains two space-separated integers $plan\_start_j$ and $plan\_end_j$, denoting the start and the end day of Sam's plan.

Output

Output **Q** lines - each containing an integer corresponding to the maximum number of the courses Sam can complete in the corresponding planned visit.

Example

Sample Input	Sample output
3 3 1 3 5 6 2 4 1 6 1 3 2 3	2 1 0

UNITED INTERNATIONAL UNIVERSITY
Department of Computer Science and Engineering (CSE)

Q3 Burst the Balloons

There are some spherical balloons taped onto a flat wall that represents the XY-plane. The balloons are represented as a 2D integer array of points where `points[i] = [xstart, xend]` denotes a balloon whose horizontal diameter stretches between x_{start} and x_{end}. You do not know the exact y-coordinates of the balloons.

Arrows can be shot up directly vertically (in the positive y-direction) from different points along the x-axis. A balloon with x_{start} and x_{end} is burst by an arrow shot at x if $x_{start} <= x <= x_{end}$. There is no limit to the number of arrows that can be shot. A shot arrow keeps traveling up infinitely, bursting any balloons in its path.

Given the array points, return the *minimum number of arrows* that must be shot to burst all balloons.

Sample Input	Sample Output	Explanation
4 10 16 2 8 1 6 7 12	2	The balloons can be burst by 2 arrows: - Shoot an arrow at x = 6, bursting the balloons [2,8] and [1,6]. - Shoot an arrow at x = 11, bursting the balloons [10,16] and [7,12].
4 1 2 3 4 5 6 7 8	4	One arrow needs to be shot for each balloon for a total of 4 arrows.

Q4 WANDA in Action

Problem

In 2021, The Avenger land became infected by a deadly virus named “Thanos-v2.0”. The Avenger land has many cities and some cities are connected to other cities. In order to prevent virus from spreading Wanda plans on destroying the connection between all the cities. Wanda has got a power called **wanda-suprema**. Using this power she can destroy any city, which results in destruction of all connections from this city. For destroying one city, Wanda requires **one unit of** wanda-suprema power. Wanda’s final aim is to isolate all the cities. In order to do so, Wanda follows a simple approach, he keeps on destroying the city with most number of connections in it at that moment.

Since Wanda is a high-tempered lady who is not very sharp in calculation, please help her in finding out the units of **wanda-suprema** power required by her to achieve the aim. There cannot be multiple connections between two cities i.e. two cities can have only one road connected to them. **Note:** If there are multiple cities with highest connections, destroy the city with the lowest index.

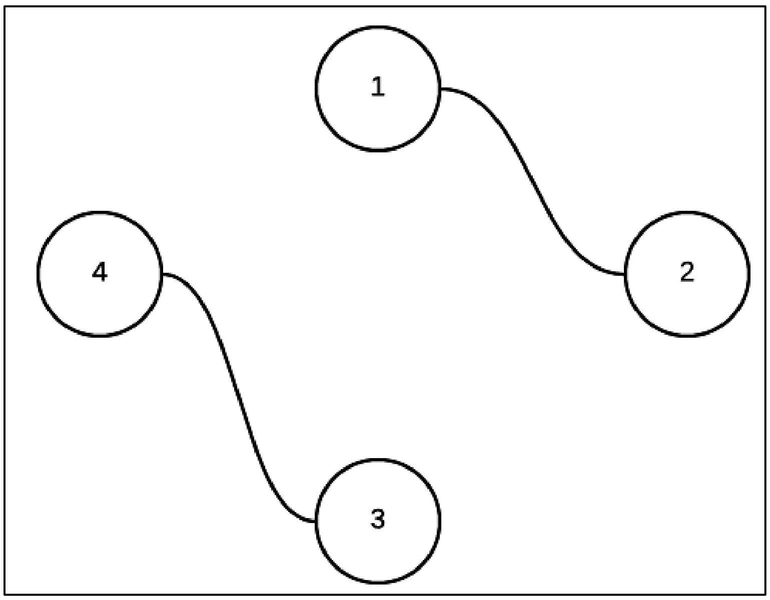
Input Format:

- The first line will contain 2 space separated integers ***N*** and ***M***, where ***N*** is the number of cities in Wanda's town and ***M*** is the number of connections that the cities have.
- Next ***M*** lines follow, each line consists of two integers ***A*** and ***B***, specifying that city ***A*** has a connection with city ***B*** and ***B*** has a connection with the city ***A***.

Output Format:

For each test case, output the **minimum energy** that Wanda requires to achieve his aim.

Image for sample Test case:



Sample Input	Sample Output	Explanation
4 2 1 2 3 4	2	To achieve Wanda's aim, she can either destroy the cities numbered: ➢ 1 and 3, or ➢ 2 and 3, or ➢ 1 and 4, or ➢ 2 and 4

UNITED INTERNATIONAL UNIVERSITY
Department of Computer Science and Engineering (CSE)

Q5: 'JALIL AND CO' FOOD DELIVERY SERVICE

In Narail, Samrat Jalil has started new delivery company named "Jalil and Co" has started. Timely delivery is very important for them. They want to tell the customers an exact timing of their arrival so customers don't have to suffer. However, there are some challenges in their business.

There are several junctions in Narail connected by roads. There is at most one road between any pair of junctions. There is no road connecting a junction to itself. The travel time for a road is the same in both directions. At every junction there is a single traffic light. These traffic lights are a bit peculiar. Starting from time 0, each light flashes green once every T time units, where the value of **T** is different for each junction.

A delivery vehicle that is at a junction can start moving along a road only when the light at the current junction flashes green. If a vehicle arrives at a junction between green flashes, it must wait for the next green flash before continuing in any direction. If it arrives at a junction at exactly the same time that the light flashes green, it can immediately proceed along any road originating from that junction. You are given a city map that shows travel times for all roads.

For each junction **i**, you are given $\mathbf{T_i}$, the time period between green flashes of the light at that junction. Your task is to find the minimum time taken from a given source junction to a given destination junction for a vehicle when the traffic starts.

Input

- There are **N** junctions and **M** roads. The junctions are identified by integers 1 through **N**.
- The first line of input contains two integers: the source junction and the destination junction.
- The second line contains two integers: **N** and **M**.
- The third line contains N integers, $\mathbf{T_1, T_2, \ldots T_N}$, describing the time periods at which the traffic lights flash green. The light at junction i flashes green at times $\mathbf{0, T_i, 2T_i, 3T_i, \ldots}$
- The next M lines contain information about the M roads. Each line has three integers $\mathbf{i, j, l_{ij}}$, where: **i** and **j** are the junctions connected by this road
- $\mathbf{l_{ij}}$ is the time required to move from junction i to junction j using this road

Output

A single line consisting of a single integer, the time taken by a minimum-time path from source to destination.

Example

Sample Input	Sample Output	Explanation
1 4 4 5 4 3 2 5 1 2 4 1 3 8 2 3 6 2 4 10 3 4 7	15	Four Possible Options are there to reach the destination 4: 1. 1 to 2 to 4 takes time 4 + 2 (wait till 6) + 10 = 16. 2. 1 to 3 to 4 takes time 8 + 0 (no wait) + 7 = 15. 3. 1 to 2 to 3 to 4 takes time 4 + 2 (wait till 6) + 6 + 0 (no wait) + 7 = 19. 4. 1 to 3 to 2 to 4 takes time 8 + 0 (no wait) + 6 + 1 (wait till 15) + 10 = 25