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

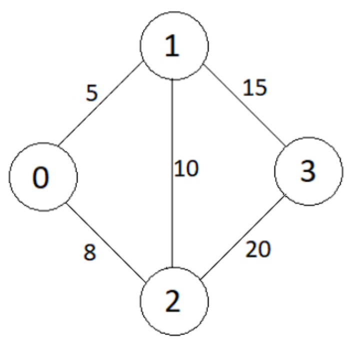
Course Title: Data Structure & Algorithm Lab II **Course Code:** CSE 2218
Trimester & Year: Fall 2023 **Section:** D **Credit Hours:** 1.0 (MdMH)

ASSIGNMENT 03: DSU & MST

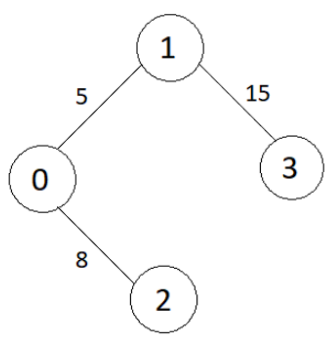
Q1: Prim's MST

You are given an undirected connected weighted graph having **'N'** nodes numbered from '1' to 'N'. A matrix **'E'** of size **M x 2** is given which represents the **'M'** edges such that there is an edge directed from node **E[i][0]** to node **E[i][1]**. You are supposed to return the minimum spanning tree where you need to return weight for each edge in the MST.

For example:



The MST (Minimum Spanning Tree) for the above graph is:



Input Format:

The first line Contains an integer **'T'** representing the number of the test case. Then the test cases are as follows.

The first line of each test case argument contains a given integer **'N'** representing the number of nodes in the graph.

The second line of each test case contains a given integer **'M'** representing the number of edges in the graph.

The next 'M' lines in each test case contain a matrix 'E' of size M x 2 which represents the 'M' edges such that there is an edge directed from node E[i][0] to node E[i][1].

Output Format:

For each test case, print the minimum spanning tree in the form of edges and their weights which are included in the MST.

Sample Input 1:

1
5 14
1 2 2
1 4 6
2 1 2
2 3 3
2 4 8
2 5 5
3 2 3
3 5 7
4 1 6
4 2 8
4 5 9
5 2 5
5 3 7
5 4 9

Sample Output 1:

1 2 2
1 4 6
2 3 3
2 5 5

Explanation of Input 1 :

The Minimum spanning tree for the given graph will contain the edges: (1,2) with weight 2, (1,4) with weight 6, (2,3) with weight 3 and (2,5) with weight 5.

Sample Input 2 :

1

5 15

1 2 21

1 4 16

2 1 12

2 3 13

2 4 18

2 5 15

3 2 13

3 5 17

4 1 16

4 2 18

4 5 19

5 1 18

5 2 15

5 3 17

5 4 19

Sample Output 2 :

1 2 12

1 4 16

2 3 13

2 5 15

Explanation of Input 2 :

The Minimum spanning tree for the given graph will contain the edges: (1,2) with weight 12, (1,4) with weight 16, (2,3) with weight 13 and (2,5) with weight 15.

Tip:

Create a function like the following that will find the MST and return it:

```
vector<pair<pair<int, int>, int>>  calculatePrimsMST(int n, int m, vector<pair<pair<int, int>, int>> &g)
{
   // Write your code here.
}
```

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

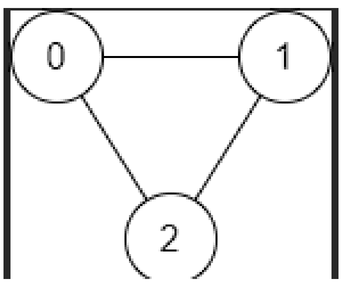
Q2: Find if Path Exist in Graph

There is a bi-directional graph with **n vertices**, where each vertex is labeled from ***0 to n - 1*** (inclusive). The edges in the graph are represented as a 2D integer array **edge**, where each **edges[i] = [u_i, v_i]** denotes a bi-directional edge between vertex, u_i and vertex, v_i. Every vertex pair is connected by at most one edge, and no vertex has an edge to itself.

You want to determine if there is a valid path that exists from vertex source to vertex destination.

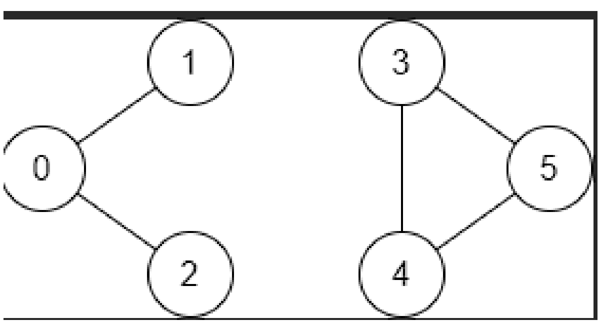
Given edges and the integers n, source, and destination, return true if there is a valid path from source to destination, or false otherwise.

Example 1:



Input: n = 3, edges = [[0,1],[1,2],[2,0]], source = 0, destination = 2
Output: true
Explanation: There are two paths from vertex 0 to vertex 2:
- 0 → 1 → 2
- 0 → 2

Example 2:



Input: n = 6, edges = [[0,1],[0,2],[3,5],[5,4],[4,3]], source = 0, destination = 5
Output: false
Explanation: There is no path from vertex 0 to vertex 5.

Constraints:

1 <= n <= 2 * 105

0 <= edges.length <= 2 * 105

edges[i].length == 2

0 <= ui, vi <= n - 1

ui != vi

0 <= source, destination <= n - 1

There are no duplicate edges.

There are no self edges