**Assignment No.: 7**

**Title:** Minimum Spanning Tree

**Course Outcome:**

**CO1(**C214447.1**):** Analyze algorithms and to determine algorithm correctness and time efficiency class.

**CO2(**C214447.2**):** Implement abstract data type (ADT) and data structures for given application.

**CO3(**C214447.3**):**Design algorithms based on techniques like brute -force, divide and conquer, greedy, etc.).

**CO5(**C214447.5**):** Analyze of algorithms with respect to time and space complexity.

**Date of Completion:**

**Assessment Grade / Marks:**

**Assessor’s Sign with Date:**

**Assignment No: 7**

**Title:** Minimum Spanning Tree

**Aim:** Represent graph using adjacency matrix and find minimum spanning tree.

**Objective:** Represent graph using adjacency matrix and find minimum spanning tree.

## Problem Statement:

Represent a graph of your college campus using adjacency list /adjacency matrix. Nodes should represent the various departments/institutes and links should represent the distance between them.

Find minimum spanning tree

1. Using Kruskal’s algorithm.
2. Using Prim’s algorithm.

**Course Outcome:** CO Number: Applicable CO : Blooms Taxonomy Category

**Requirements:** (Components / Digital Kits / Platform / Software / Hardware)

Platform :- Online GDB Compiler

## Theory / Procedure / Diagrams / Circuits:

**Kruskal’s Algorithm :-**

Kruskal's Algorithm is used to find the minimum spanning tree for a connected weighted graph.

The main target of the algorithm is to find the subset of edges by using which, we can traverse every vertex of the graph. Kruskal's algorithm follows greedy approach which finds an optimum solution at every stage instead of focusing on a global optimum.

It is used for finding the Minimum Spanning Tree (MST) of a given graph.

To apply Kruskal’s algorithm, the given graph must be weighted, connected and undirected.

## Prim’s Algorithm :-

Prim's algorithm is a minimum spanning tree algorithm that takes a graph as input and finds the subset of the edges of that graph which form a tree that includes every vertex has the minimum sum of weights among all the trees that can be formed from the graph. It falls under a class of algorithms called greedy algorithms that find the local optimum in the hopes of finding a global optimum.

We start from one vertex and keep adding edges with the lowest weight until we reach our goal.

**Algorithm / Methods / Steps:** (if applicable):

# Kruskal’s Algorithm (Algo):- STEP 1: START

**STEP 1:** Sort all the edges from low weight to high weight.

**STEP 1:** Take the edge with the lowest weight and use it to connect the vertices of graph.

**STEP 1:** If adding an edge creates a cycle, then reject that edge and go for the next least weight edge.

**STEP 1:** Keep adding edges until all the vertices are connected and a Minimum Spanning Tree (MST) is obtained.

**STEP 1:** STOP

# Prim’s Algorithm (Algo):- Step 1: START

**Step 2:** Keep a track of all the vertices that have been visited and added to the spanning

tree.

**Step 3:** Initially the spanning tree is empty.

**Step 4:** Choose a random vertex, and add it to the spanning tree. This becomes the root node.

**Step 5:** Add a new vertex, say x, such that

x is not in the already built spanning tree.

x is connected to the built spanning tree using minimum weight edge. (Thus, x can be adjacent to any of the nodes that have already been added in the spanning

tree).

Adding x to the spanning tree should not form cycles.

**Step 6:** Repeat the Step 4, till all the vertices of the graph are added to the spanning tree.

**Step 7:** Print the total cost of the spanning tree.

**Step 8:** STOP

**Input**: (Test Cases / Data sets / Database Links):

Adjacency Matrix(Graph):-

{0,90,60,0,0,0},

{90,0,60,300,90,0},

{60,60,0,100,80,0},

{0,300,100,0,200,300},

{0,90,80,200,0,70},

{0,0,0,300,70,0} };

**Output:** (Results / Visualization):

=========================================================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Prim's Algorithm\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

=========================================================================

Department -----> Distance

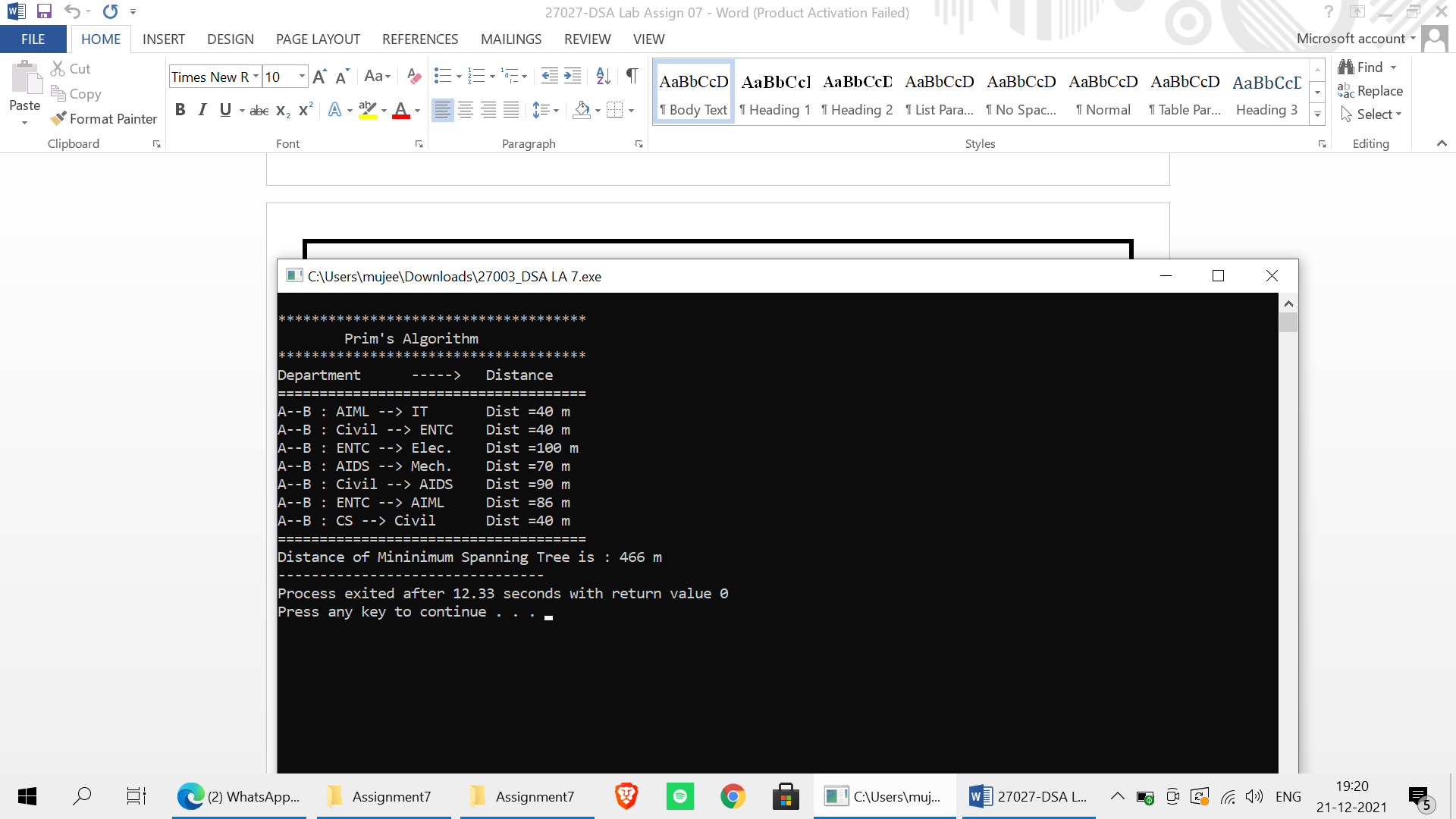
========================================================================= A--B : ENTC --> IT Dist =60 m

A--B : Comp. --> ENTC Dist =60 m

A--B : ENTC --> Elec. Dist =100 m A--B : ENTC --> Mech. Dist =80 m A--B : Mech. --> AIDS Dist =70 m

=========================================================================

Distance of Minimum Spanning Tree is : 370 m



**Inference:** Hence we have studied the represention of graph using adjacency matrix and find minimum spanning tree using prim’s algorithm.