**Department of Computer Science and Engineering**

|  |  |
| --- | --- |
| **Course Code: CSE221** | **Credits: 1.5** |
| **Course Name: Algorithms** | **Semester: Fall’18** |

**Lab 05  
Minimum Spanning Tree**

**(Prim’s and Kruskal’s algorithm)**

1. **Topic Overview:**

The idea of Minimum Spanning Tree (MST) or Minimum Weight Spanning Tree is to find a subset of edges of a connected graph that connects all the vertices of the graph so that total weight of all the selected edges are minimized and no cycle is present in the path. Minimum Spanning Tree has a great impact on solving real life problems of different dimensions. The key point is to use MST for optimization problems where a path of lesser resistance needs to be found to reach desired conditions.

Prim’s and Kruskal’s algorithm are two different greedy approach to find MST in a connected graph. We follow greedy approach so that we can use the heuristics connected to the problem which aid us to come up with a solution without roaming around the whole graph. It reduces the computation for those parts of the graph which has less possibilities of being a part of the solution to the problem defined.

1. **Lesson Fit:**

To solve this problem, the students must have a basic idea on the following concepts:

* 1. Graph Representation and Computation
  2. Greedy Algorithm
  3. Optimization Task

1. **Learning Outcome:**

After this lab, the students will be able to:

* 1. Find minimum weighted path from a connected graph
  2. Represent real life problem in graphs
  3. Optimize the calculation for finding solution

1. **Anticipated Challenges and Possible Solutions**
   1. Solving different optimization problem using the concept of MST is often challenging for the students as they fail to identify how to map the problem in graph representation and which of the greedy approach they should use in different scenario.

**Solutions:**

* + 1. The students need to map the optimization problem in terms of vertices and edges and find out the conditions of the scenario which goes with Prim’s or Kruskal’s approach so that they get the optimal solution.

1. **Acceptance and Evaluation**

Students will show their progress as they complete each problem. They will be marked according to their class performance. There may be students who might not be able to finish all the tasks, they will submit them later and give a viva to get their performance mark. The mark distribution for the lab will be as follows:

Code: 05

Viva: 05

1. **Activity Detail**
   1. **Hour: 1  
      Explanation:**The lab instructor will explain how to represent a connected graph in a matrix and traversing the graph using the matrix. Instructor will explain the Prim’s algorithm in a connected graph and will let the students understand the approach thoroughly.
   2. **Hour: 2**

**Implementation:**

After explanation, the students will implement Prim’s algorithm to solve a given problem.

**Problem Task:**

* + 1. Page 4
  1. **Hour: 3**

**Explanation:**Instructor will explain the Kruskal’s algorithm in a connected graph and will let the students understand the approach thoroughly.

**Implementation:**

After explanation, the students will implement Kruskal’s algorithm to solve a given problem.

**Problem Task:**

* + 1. Page 4

1. **Home tasks**
   1. Unfinished tasks

**Lab 7 Activity List**

**Problem Description:**

Andrew, an Australian, has recently come to Bangladesh. He wants to visit 6 cities of the country and asked you to accompany him throughout the visit. As you will guide him, you need to decide how you can complete the trip without travelling extra distance. So you need to design the trip. You don’t have to end your tour from where you will start. For your assistance, distances of 6 cities of Bangladesh are given in the following table. Find out your tour path based on the optimization knowledge you have.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Dhaka | Chittagong | Barisal | Sylhet | Rajshahi | Khulna |
| Dhaka | 0 | 264 | 277 | 346 | 270 | 335 |
| Chittagong | 264 | 0 | 541 | 425 | 534 | 599 |
| Barisal | 277 | 541 | 0 | 623 | 401 | 322 |
| Sylhet | 346 | 425 | 623 | 0 | 616 | 694 |
| Rajshahi | 270 | 534 | 401 | 616 | 0 | 295 |
| Khulna | 335 | 599 | 322 | 694 | 295 | 0 |

**Input:**A line with the name of n cities, followed by n times: • A line contains n numbers representing the distances of that city from every other city

**Output:**Output shows the list direct paths between the pair of cities which optimizes the tour.

**Sample Input:**Newyork Washington California  
0 3750 3923   
3750 0 1227   
3923 1227 0

**Sample Output:**(Newyork – Washington) , (Washington - California)

**You have to implement both Prim’s and Kruskal’s algorithm as two different methods in a class named “MST”.** You are encouraged to use Java’s built-in class HashSet (<https://docs.oracle.com/javase/7/docs/api/java/util/HashSet.html>) for Kruskal.