**REPORT**

**Project 2: Graph Algorithms**

**Single-source shortest path algorithm and Minimum Spanning Tree (MST)**

**Submitted By**

Abdullah Al Raqibul Islam (ID# 801151189)

**Code Repository Overview**

I have used C++ as the programming language for this project. The project structure looks like this

//fig

**Problem 1: Single Source Shortest Path**

Short description:

Data-structure:

Runtime analysis:

Sample I/O specification:

**Problem 2: Minimum Spanning Tree (MST)**

**Short description:** For this problem, I have implemented Kruskal’s algorithm to find minimum spanning tree of an undirected graph. This algorithm will only work on undirected graph. The working principal of Kruskal’s algorithm is to choose the edge with minimum weight (greedy approach) that connects any two trees in the forest. Eventually it will become a single tree (only if it is a connected graph) which connects all the nodes of the graph.

**Data-structure:** I have used disjoint set data-structure to track the set-id of a node. Initially every node assigned to his own set-id. When we found an edge, which connects two nodes from two separate sets, we make a connection between them by applying Union operation. I applied two technique called **path-compression** in find and **union-by-rank** to reduce the time complexity of each operation done on disjoint set data-structure. This two techniques actually complement each other and make the amortized time complexity even smaller than O(log n).

To store edges, I used vector from C++ Standard Template Library (STL). To sort the edge list by weight, I use sort function provided by C++ standard library which requires O(n log(n)) in the worst case scenario.

**Runtime analysis:**

**Sample I/O specification:**

**Test platform:**

* Processor: Intel(R) Xeon(R) CPU E5-2620 2.00GHz (12 Core)
* Linux version: 5.0.0-27-generic
* g++ version: (Ubuntu 7.4.0-1ubuntu1~18.04.1) 7.4.0