**ALGORITHM DESIGN AND ANALYSIS**



**PROJECT SYNOPSIS**

**COMPARING KRUSKALS AND PRIM’S ALGORITHM**

**IN MST**

**SUBMITEED BY: NOUSHAD AHMED (2K19/CO/266)**

**NEERAJ GUPTA (2K19/CO243)**

**GUIDED BY: SHIKHA MAM**

**ABSTRACT**

This project will contain an analytical comparison between two algorithms which are commonly used to solve minimum spanning tree (MST) problem. Those two algorithms are none other than Kruskal and Prim. Kruskal is a greedy algorithm that works by sorting all the edges of the graph in non-decreasing order of the graph weight. Next step is to pick the smallest edge and check if it forms a cycle with the spanning tree that is already arranged so far. If cycle is not formed then include this edge, otherwise leave it. Prim algorithm (also known as Jarnik's algorithm) is another greedy algorithm which works by joining two subsets, one containing vertices already included in the MST, the other set contain the rest of the vertices which are not included, and we try to connect both subsets with weight edge, hence the minimum spanning tree.

**OBJECTIVE**

To analyse prim’s and kruskals algorithm and go through time complexities of these algorithm and find which algorithm is efficient under different condition.

**INTRODUCTION**

In this Project we selected the topic as a title Comparing Kruskal’s and Prim’s algorithm in MST (Minimum Spanning Tree)

In this paper we will analyze two algorithms which one is better between Kruskal’s and Prim’s algorithm to solve MST (Minimum Spanning Tree) problem, and we will use c++ as the programming language to solve the problem.

In this project we will discuss about MST problem using methods; Spanning Tree is a subgraph of undirected graph that is a tree including all vertices of the graph, with a minimum number of edges. Minimum Spanning Tree is same as spanning tree, but with minimum total edge weights as well. In this problem we will comparing two methods which are Kruskal’s algorithm and Prim’s algorithm which one is faster and efficient from its time complexity.

**TECHNICAL SPECIFICATIONS**

Brief Algorithm Kruskal's algorithm is a minimum-spanning-tree algorithm which finds an edge of the least possible weight that connects any two trees in the forest. It is a greedy algorithm in graph theory as it finds a minimum spanning tree for a connected weighted graph adding increasing cost arcs at each step.This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. If the graph is not connected, then it finds a minimum spanning forest(a minimum spanning tree for each connected component).

Prim’s algorithm is a greedy algorithm that finds a minimum spanning tree for a weighted undirected graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. The algorithm operates by building this tree one vertex at a time, from an arbitrary starting vertex, at each step adding the cheapest possible connection from the tree to another vertex.

**Kruskal’s Algorithm**  
In this section we will explain what are the function that used in this algorithm. Here are the functions.

* **Def\_init\_(self,vertices)**

This function has default dictionary to store the graph.

* **Def addEdge(u, v, w)**

This function will be use for adding the edge to the graph

* **Def find (self, parent, i)**

This function will find set of an element i.

* **Def union (self, parent, rank, x, y)**

This function will union one graph to another.

* **Def KruskalMST (self)**

This is the main function to Kruskal algorithm itself.

**Prim’sAlgorithm**

In this section we will explain what are the function that used in this algorithm. Here are the functions.

* **Def \_init\_ (self, graph)**

This function will store the graph.

* **Def printMST (self, parent)**

This function to print the constructed MST stored in parent.

* **Def minKey (self, key, mstSet)**

This function is to find the minimum distance value which the vertex not included in the set.

* **Def primMST(self)**

This is the main function for the prim’s algorithm itself.

**Time Complexity**

Kruskal’s algorithm: O(E log V)

Prim’s algorithm : O(VlogV + ElogV) = O(ElogV)

**FUTURE SCOPE**

Applications where Kruskal’s algorithm is generally used:

1. Landing cables

2. TV Network

3. Tour Operations

4. LAN Networks

5. A network of pipes for drinking water or natural gas.

6. An electric grid

7. Single-link Cluster

Applications where Prim’s algorithm is generally used:

1. All the applications stated in the Kruskal’s algorithm’s applications can be resolved using Prim’s algorithm (use in case of a dense graph).

2. Network for roads and Rail tracks connecting all the cities.

3. Irrigation channels and placing microwave towers

4. Designing a fiber-optic grid or ICs.

5. Travelling Salesman Problem.

6. Cluster analysis.

7. Pathfinding algorithms used in AI(Artificial Intelligence).

8. Game Development

9. Cognitive Science

**REFERENCE**

Denver, H. J.-U. (1998). Greedy Algorithms for Minimum Spanning Tree. Spanning Tree. Knoxvile, K. C.-T. (2017). minimum spanning trees. Retrieved from http://web.eecs.utk.edu: http://web.eecs.utk.edu/~cphillip/cs594\_spring2017/presentations/minimum-spanning-trees.pdfThomas H. Cormen, C. E. (2001). Introduction to Algorithm. London, England: The MIT Press