CSCI_6212_11 - Design and Analysis of Algorithms - Project 2

1. Problem Statement: We are required to analyse the below problem statement:
Input: Given a weighted connected graph G = (V, E), where w[i,j] or w(e) is the weight of edge between vertices i and j.
Objective: Find a minimum-weight spanning tree of G

2. Theoretical Analysis

A minimum spanning tree of a graph refers to a selected set of edges that connects all the vertices of the graph while minimizing the total weight of the connecting edges. Kruskal's algorithm is the best approach to finding a minimum spanning tree of any graph since it uses the greedy approach for selecting the edges from the graph. The algorithm works as below:

- **Algorithm:** KruskalsMST (G, W[1:n,1:n], T)

```
Sort m edges in the ascending order of their weights: e[1], e[2], .. e[m].
Initialize a counter k = 1
Initialize an empty tree T
While (number of edges in Tree < n-1) {
    If adding the edge e[j] does not create a cycle, add edge e[j] to tree T
    k++
}
```

- Consider "V" to be the number of vertices and "E" to be the number of edges of any connected graph G with the weights of all the edges. The Kruskal's algorithm for finding minimum spanning tree of graph G is based on the following operations which can be achieved by using disjoint-set or union-find data structure since they facilitate these operations:
  a) Sorting the edges by ascending of their weights which takes O(E log V) time.
  b) Checking whether adding an edge creates a cycle by using **Find(x)** operation which finds the set to which the element x belongs (i.e., if the vertices are connected). This takes O(E log V) time to achieve since it is executing once per edge.
  c) Updating the edges for the connected vertices by using **Union (x, y)** operation per edge which merges the sets containing elements x and y. This requires O(V log V) time for V-1 edges.
- Therefore, the total time complexity of the algorithm is O(E log V + V log V) which on further simplification leads to **O(E log V) as the time complexity.** Since minimal weight is the primary objective for finding the MST, Kruskal's algorithm uses the greedy method by consistently choosing the smallest edge that does not create a cycle, resulting in an MST. It is one of the best examples of a greedy method for finding the Minimum Spanning Tree because of its simplicity, efficiency, and optimality to handle different types of graphs.
- **Example Test Case:** Consider a graph G with 4 vertices and 6 edges where weights of each edge between vertices is represented by disjoint set (v_init, v_end, weight).
  - If (0, 1, 2) (0, 3, 7) (0, 2, 5) (1, 3, 3) (1, 2, 1) and (2, 3, 4) are the weights of 6 edges in the graph, the algorithm chooses the edges with least weights to form a minimum spanning acyclic tree of the graph connecting all the vertices of the graph.
  - Hence, using greedy approach, Kruskal's algorithm forms an MST by selecting the edges (1, 2, 1), (0, 1, 2) and (1, 3, 3) with least weights of 1, 2 and 3 respectively adding them to the spanning tree after checking if it already creates a cycle or not.

3. Experimental Analysis

3.1 Program Listing

- GitHub Link for the code: Namana_G21372717_Algorithms_Project_2_Code

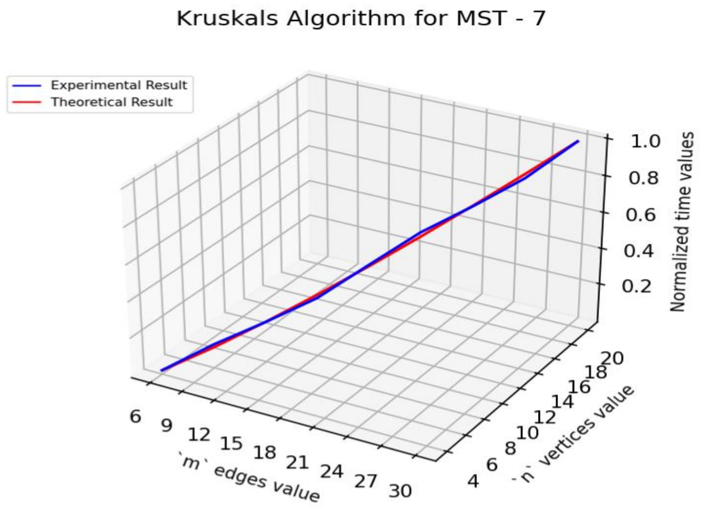
3.2 Data Normalization Notes

Average of Experimental results is: **1152210.56**
Average of Theoretical result is: **66.28**
Scaling constant = Average of Experimental result / Average of Theoretical result
Therefore, Scaling Constant for normalizing the theoretical results = **17383.98552**

3.3 Output Numerical Data

Vertices value (v)	Edge Value (e)	Experimental Result in ns	Theoretical Result (e log v)	Scaling Constant	Scaled Theoretical Result
4	6	302277.00	12.00		208607.826
6	9	488636.00	23.27		404438.431
8	12	795363.00	36.00		625823.476
10	15	962233.00	49.83		866243.998
12	18	1296068.00	57.06		991930.213
14	21	1468066.00	79.96		139023.482
16	24	1594672.00	96.00		166886.261
18	27	1623848.00	112.78		196056.886
20	30	1954732.00	129.66		225407.562
		1152210.56	**66.28**	**17383.98552**	

3.4 Graph



3.5 Graph Observations

The plot of the experimental results **increases constantly** along with the increase in theoretical result at all the points on the graph for respective increase in values of edges and vertices. As the number of edges increases, the impact on the time complexity is more significant than the log factor depending on the number of vertices.

4. Conclusions

The line plots of the experimental and theoretical analysis are intersecting and seem to be convergent. Hence the Big O notation for finding minimum spanning tree using Kruskal's greedy method algorithm is **O(E log V)** where "E" is the number of edges and "V" is the number of vertices of the graph.