Project 2

Samank Gupta - https://github.com/samankgupta/GW_DAA_Project-2

1 Problem Statement

Option 7: Kruskal's Algorithm for Minimum Spanning Tree

2 Theoretical Analysis

Kruskal's algorithm follows a greedy approach by selecting the minimum weighted edge that does not form a cycle. This process is repeated until we have selected n-1 edges.

1. **Sort the Edges:** First, all edges are sorted in increasing order of weight.
2. **Initialize Disjoint Sets:** Use the Disjoint Set (Union-Find) data structure to track the sets of connected vertices. It helps in checking whether adding an edge forms a cycle by determining if two vertices are already part of the same set through the **Find** operation. Additionally, when an edge is added, the **Union** operation merges the sets of the two connected vertices, ensuring that they are part of a single connected component.
3. **Pick Edges:** Iteratively add the smallest available edge to the MST, provided it does not form a cycle.
4. **Stop**: The algorithm stops once the MST has n-1 edges.

We take **m** as the number of edges and **n** as the number of nodes. The first step of sorting the m edges by weight requires **O(m log m)** time. As $m \leq n^2$, the sorting time can be bounded as **O(m log n)**. Find and Union operation runs would contribute **O(n log n)** to the overall complexity. Thus, we require **O(m log n + n log n)** time. Since m is larger than n, we can write this as **O(m log n)** time.

3 Experimental Analysis

3.1 Program Listing

I ran a JAVA program with different values of edges and nodes and calculated the time in nanoseconds. I took values of m (edges) as 5, 10, 25, 50, 100, 200 and n (nodes) as 10, 15, 30, 55, 105 205.

3.2 Data Normalization Notes

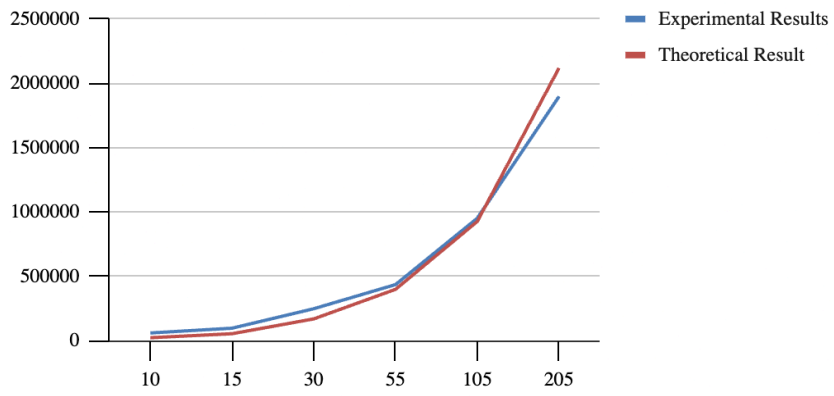
I took the average of Experimental and Theoretical results, and then calculated a scaling

constant by dividing the experimental average with theoretical average. I used this scaling constant to an adjusted theoretical result column.

3.3 Output Numerical Data

m (edges)	n (nodes)	Experimental Result, in ns	Theoretical Result	Scaling Constant	Adjusted Theoretical Result
5	10	59875	16.60964047		22914.22262
10	15	96958	39.06890596		53898.43387
25	30	248333	122.6722649		169235.4264
50	55	435833	289.0679857		398790.5811
100	105	951334	671.4245518		926279.6312
200	205	1897667	1535.89602		2118881.705
		615000	445.7898948	1379.573667	

3.4 Graph



3.5 Graph Observations

Looking at the graph, we observe that the experimental and theoretical values are very close to each other for different values of m and n.

4 Conclusions

After observing the theoretical and experimental results, we conclude that Kruskal's algorithm will have a time complexity of **O(m log n)**.

As we can see in the table and the graph, our experimental values coincide with our theoretical values, proving that our hypothesis was correct.