Assignment 3: Optimization of a City Transportation Network (MST Project)

Astana IT University
Course: Design and Analysis of Algorithms
Assignment 3
Student: Bakdaulet Rinatuly
Group: SE-2419
Date: October 2025

1. Objective

The objective of this assignment is to design and analyze two Minimum Spanning Tree (MST) algorithms – Prim's and Kruskal's – to optimize a city's transportation network by connecting all districts with minimal total construction cost. Each district is represented as a vertex, and roads are represented as weighted edges corresponding to construction costs.

2. Implementation Details

The project was implemented in Java using an object-oriented design. The architecture included classes Graph, Edge, PrimMST, and KruskalMST. Each algorithm computes the MST while tracking execution time, operation counts, and comparisons. Input data was provided in JSON format and output results were stored in both JSON and CSV for analysis. Automated JUnit tests were used to validate correctness and performance consistency.

3. Input Data

Dataset	Vertices (V)	Edges (E)	Type
small-1	5	7	Small
medium-1	12	15	Medium
large-1	25	30	Large

4. Experimental Results

The results below summarize the performance of both algorithms based on execution time, operation count, and MST total cost. Data were collected from the generated CSV file.

Graph ID	V	E	Prim Cost	Kruskal Cost	Prim Time (ms)	Kruskal Time (ms)	Ops Comparison
small-1	5	7	14	14	1	1	Prim=28, Kruskal=7

medium-1	12	15	43	43	2	3	Prim=112, Kruskal=30
large-1	25	30	92	92	6	3	Prim=275, Kruskal=60

5. Discussion and Comparison

Both algorithms produced identical MST total costs, confirming correctness. However, their performance varies depending on graph density and structure.

Prim’s algorithm is vertex-based and tends to perform better on dense graphs where the number of edges is large. It incrementally adds the smallest edge connecting an unvisited vertex, making it efficient for adjacency list representations.

Kruskal’s algorithm is edge-based and performs better on sparse graphs because it avoids redundant vertex comparisons. Its efficiency relies on the Union–Find (Disjoint Set Union) structure to detect cycles.

In practice, both algorithms demonstrated linear scalability with input size, with Kruskal slightly faster for sparse graphs and Prim more efficient for dense networks.

6. Conclusion

This experiment confirmed that both Prim’s and Kruskal’s algorithms produce the same optimal MST cost, ensuring network connectivity with minimal total construction cost. Kruskal’s algorithm is preferable for sparse networks, while Prim’s is more efficient for dense networks due to its adjacency-based updates. Overall, both approaches proved to be reliable and effective for city transportation optimization.

7. References

1. Cormen, T. H., et al. Introduction to Algorithms, MIT Press.
2. GeeksForGeeks – “Prim’s and Kruskal’s MST Algorithms”.
3. Baeldung – “Minimum Spanning Tree in Java”.
4. AITU Course Materials, Design and Analysis of Algorithms, 2025.