Assignment 3 — Optimization of a City Transportation Network (Minimum Spanning Tree)

Student: Farkhad Imanbayev

Group: SE-2438

Objective

The goal of this assignment was to **Prim's** and **Kruskal's** algorithms to optimize a city's transportation network by finding the **Minimum Spanning Tree (MST)** — the minimum-cost set of roads that connects all city districts.

Input Data

Two example graphs:

- **Graph 1:** 5 vertices, 7 edges
- **Graph 2:** 4 vertices, 5 edges

Edge weights represent road construction costs.

Results

Graph	Vertices	Edges	Prim Cost	Kruskal Cost	Prim Time (ms)	Kruskal Time (ms)	Prim OC	Kruskal OC
1	5	7	16	16	0.1591	0.191	29	5
2	4	5	6	6	0.0306	0.0609	15	3

Both algorithms produced the same MST total cost for both graphs, confirming correctness.

Comparison Between Prim's and Kruskal's Algorithms

Theory

- **Prim's Algorithm:** builds the MST starting from a chosen vertex and repeatedly adds the smallest edge connecting the visited vertices to an unvisited vertex.

Efficient for **dense graphs**, especially when implemented with adjacency lists and a priority queue.

- **Kruskal's Algorithm:** sorts all edges by weight and joins components using the **union–find** data structure.
  Efficient for **sparse graphs** because it avoids checking all vertex connections.

In Practice

- In both test graphs, **Prim's and Kruskal's algorithms produced identical MST total costs** (16 and 6).
- However, **Kruskal's algorithm required significantly fewer operations** —
  5 vs 29 operations on Graph 1, and 3 vs 15 on Graph 2.
- Despite this, **Prim's algorithm showed slightly better execution time** on small graphs (0.159 ms vs 0.191 ms in Graph 1).
- This shows that in small datasets, time differences are negligible, but operation efficiency trends are visible.

Conclusions

Both algorithms, **Prim** and **Kruskal**, made the same correct result. They found the Minimum Spanning Tree with the same total cost. This means both programs work well. **Kruskal** used fewer steps and was faster for small and simple graphs. **Prim** worked a little slower but was stable. **Kruskal** is better for graphs with few roads, and **Prim** is better for graphs with many roads. In general, both algorithms are good for building the best and cheapest road network for the city.

References

1. Astana IT University — *Design and Analysis of Algorithms (DAA)* course materials, 2025.