



ASSIGNMENT 4 - MINIMUM SPANNING TREE (MST) ALGORITHMS

Due Date: Friday November 28th, 2025, at 5:00 pm
(100 points)

1 Goals

- Implement and deeply analyze both Prim's and Kruskal's algorithms for the Minimum Spanning Tree (MST) problem.
- Collect and compare empirical results across multiple graph types.
- Explore state-of-the-art MST research and advanced applications.
- Develop skills in critical evaluation and academic communication.

2 Assignment Components

2.1 MST Algorithm Implementations (30 points)

2.1.1 Prim's Algorithm:

- Implement Prim's MST algorithm for a weighted, undirected graph using a heap data structure as an abstract data type.

2.1.2 Kruskal's Algorithm:

- Implement Kruskal's MST algorithm, making use of an efficient union-find (disjoint-set) data structure.
- Implement Prim's MST algorithm for a weighted, undirected graph using a heap data structure as an abstract data type.

An input graph file will be available. The format of the input file is the following:

- The first line of the input file contains an integer indicating number of vertices of the input graph.
- Each of the remaining lines contains a triple "*i j w*" indicating an edge between vertex *i* and vertex *j* with cost *w*.
- Vertex **1** can be considered as the root.

The output of your program should be the following:

- The input graph in adjacency list representation format listing each edge with its weight.
- The edges (with their weights) of the minimum spanning tree, in the order produced by each algorithm.

2.2 Empirical Analysis and Comparison (30 points)

- Generate results for at least three types of graphs:
 1. Sparse (few edges relative to vertices)

2. Dense (many edges relative to vertices)
 3. Large graphs (≥ 500 vertices)
- Measure and report for both algorithms:
 - Execution time
 - Memory usage
 - Order/selection of MST edges
 - Present data in tables and graphs.
 - Discuss observed differences and theoretical expectations for each scenario.

2.3 Research Component (30 points)

- Summarize at least two recent advances in MST algorithms (e.g., incremental MST, clustering with MST, temporal/dynamic MST, special data structures like AM-trees).
- Include context, method, and how they compare to classical approaches.
- Suggest one open research question or advanced application based on the papers you reviewed.
- Research Reference Suggestions:
 - Review comparative studies and surveys on modern MST algorithms, including non-greedy and dynamic approaches.
 - Examine applications to clustering, resource-constrained graphs, and temporal network analysis.

2.4 Reflection and Critical Discussion (10 points)

- Discuss when Prim's or Kruskal's algorithm is preferable, based on your experiments.
- Comment on practical tradeoffs (e.g., implementation complexity, data structure overheads, parallelization potential).

3 Hand In

- Use IntelliJ IDEA to create a Java project for your implementations and name it: group#_cs8050_assignment4.
- ZIP your entire project folder, including the project code and a written report containing your results and research summaries.
- Submit the ZIP file through Canvas.