

CSE 208: Data Structures and Algorithms II

Offline Assignment: Minimum Spanning Tree (MST)

Department of Computer Science and Engineering

Term: July 2025

1 Introduction

A remote archipelago consists of N islands ($3 \leq N \leq 1000$). To provide electricity to all islands, the government plans to build underwater power cables. There are M ($N \leq M \leq 10000$) potential routes between islands where cables can be laid, each with a specific construction cost.

Your task is to find the minimum total cost required to connect all islands so that every island can receive power from a central generator located at a designated root island.

2 Task

Implement the following two algorithms to find the Minimum Spanning Tree (MST):

1. **Prim's Algorithm:** Use a Priority Queue-based implementation with a time complexity of $O(E \log V)$.
2. **Kruskal's Algorithm:** Use a Disjoint Set Union (DSU) with path compression and union by rank/size, with a time complexity of $O(E \log E)$.

3 Input Format

Input will be read from standard input (stdin). You can use file redirection (like `./program.exe < input.txt`) to read from a file.

- The first line contains two integers N (number of vertices) and M (number of edges).
- Each of the following M lines contains three values: u (start node), v (end node), and w (weight/cost).
- Vertices are 0-indexed ($0 \leq u, v < N$).
- The last line of the input file contains a single integer representing the **root node** for Prim's algorithm.

4 Output Format

Output the results for both algorithms clearly.

- **Total Weight:** The sum of the weights of the edges in the MST.
- **Edge List:** Each edge should be printed as two space-separated integers u and v on a new line.

5 Sample I/O

See more examples in the attached `sampleio.zip` file.

Sample Input

```
5 7
0 1 2
0 3 6
1 2 3
1 3 8
1 4 5
2 4 7
3 4 9
0
```

Sample Output (Prim's Algorithm)

```
Total weight 16
Root node 0
0 1
1 2
1 4
0 3
```

Sample Output (Kruskal's Algorithm)

```
Total weight 16
0 1
1 2
1 4
0 3
```

6 Submission Guidelines

1. Create a folder named with your 7-digit Student ID.
2. Include your source files (e.g., `.cpp`, `.java`, or `.py`).
3. Zip the folder into a single `.zip` file.
4. Plagiarism will result in a 100% mark deduction. Ensure the work is your own.