

# Dynamic Minimum Spanning Tree

Hyoungshick Kim

November 20, 2023

## 1 Objective

Design and implement an algorithm to dynamically maintain the minimum spanning tree (MST) of an initially empty undirected graph, as edges are inserted, deleted, or their weights are changed over time.

## 2 Assumptions and Requirements

1. The graph is initially empty and undirected. Nodes are numbered from 1 to  $N$ , where  $N$  is the total number of nodes in the graph, specified in the first line of the input file 'mst.in.'  $N$  is less than or equal to 500.
2. The total number of `insertEdge`, `findMST`, `deleteEdge`, and `changeWeight` operations is less than or equal to 10,000.
3. Edge weights are positive integers.
4. If the graph is connected, `findMST` should print the total weight sum of the MST to the output file 'mst.out.' If the graph is disconnected, it should print 'Disconnected.'
5. If an operation is not valid (e.g., inserting an existing edge, deleting a non-existent edge), it should be ignored.

## 3 Example Input and Output

**Example Input (mst.in):**

```
5
findMST
insertEdge 1 2 4
insertEdge 1 3 3
insertEdge 2 3 2
insertEdge 3 4 1
insertEdge 4 5 5
findMST
```

```
deleteEdge 2 3
changeWeight 1 3 6
findMST
insertEdge 2 3 4
findMST
changeWeight 4 5 1
findMST
```

**Expected Output (mst.out):**

```
Disconnected
11
16
14
10
```

## 4 Grading Criteria

Your work will be assessed on:

1. Correctness and efficiency of your algorithm in maintaining the MST.
2. The total execution time taken by the experiments.
3. Clarity of your documentation, including approach explanation and time complexity analysis.

## 5 Submission Guidelines

- Your code should be written in ANSI C. The GNU compiler (i.e., `gcc`) on Ubuntu Linux will be used for compilation.
- Test your code extensively. A 16GB RAM environment will be employed during the testing phase. Ensure its correctness with diverse input scenarios.
- Do not use pre-defined algorithms (e.g., `qsort()`) or data structures, excluding arrays and strings. You should create the priority queue from scratch. However, basic string functions such as `strcmp`, `strcpy`, and `strlen` and I/O functions are permissible.
- Upload your C source files, a detailed document explaining your code, and a performance analysis of your algorithm to iCampus. Documentation can be written in either Korean or English.
- Ensure originality. Plagiarism checks will be conducted.