

CS 211 Data Structures and Algorithms Lab
Autumn 2022

Assignment no.	9
Objective	To implement Kruskal's algorithm to find a minimum spanning tree (MST)
Total marks	10
Due date (without penalty)	28th October (Friday) 11:59 pm
Cut-off date (with penalty - 5%)	31th October (Monday) 11:59 pm
Penalty for violating naming convention(s)	5%

The objective of this assignment is to implement Kruskal's algorithm to find a minimum spanning tree (MST) of the given connected edge-weighted undirected graph.

Command-line argument:

Your program should receive a file (input file) as a command line argument.

Input:

Your program should accept an input file as a command-line argument. A typical execution of your program will be ***./a.out input.graph***

- The input file represents a connected undirected graph with integer weights on edges.
- Every node (vertex) in the graph is labeled uniquely with a non-negative integer.
- Every line in the input file is of the form $x\ y\ w$, which represents an edge between node x and node y , where the weight of the edge is w .
- No edge is repeated in the input file.
- Since edge-weights are unique and you are using Kruskal's algorithm, it is guaranteed that the output is unique.

Task:

Implement *Kruskal's algorithm* to find a minimum spanning tree of the given connected, weighted, and undirected graph. It is recommended that you use disjoint-set forest data structure with union-by-rank and path-compression heuristics. But a simpler implementation of the algorithm will also be accepted with full credits.

Output:

Your program should create a file named '**mst.txt**'.

- Every line in the output file should be of the form $x\ y\ w$, which represents an edge xy with weight w in the minimum spanning tree.
- The edges should be present in the file in the non-decreasing order of their weights.

Note

[*Note#1* : since the graph is undirected, lines ' $x\ y\ w$ ' and ' $y\ x\ w$ ' are equivalent and both will be considered correct in the output file.]

Submission:

- The program you submit should output '**mst.txt**' when run.
- The main file of your program should be named as $\langle \text{roll no} \rangle.\langle \text{extension} \rangle$, where roll no. specifies your roll no. and the extension depends on the language you choose (Use C or C++ for this assignment). Ex: 210010001.c.
- Do the stress test of your program well before submission.
 - (i) You may use the attached sample input files for testing; the corresponding output files are also attached;
 - (ii) We have some hidden inputs with us to test your program. *The mark you obtain is purely based on whether your program correctly gives outputs for the hidden inputs.*
- If your program has only a single source file, please submit the file as it is. If your program has multiple source files, please submit your code as a zip file where the name of the zip file should be your roll number. It is important that you follow the **input/output conventions** exactly (including the naming scheme) as we may be doing an automated evaluation. ***There will be a penalty of 5% (on the mark you deserve otherwise) if you do not follow the naming conventions exactly.***
- Follow some coding style uniformly. Provide proper comments in your code.
- *Submit only through moodle. Submit well in advance.* Any hiccups in the moodle at the last minute is never acceptable as an excuse for late submission. Submissions through email or any other means will be ignored.
- Acknowledge the people (other than the instructor and TA) who helped you to solve this assignment. The details of the help you received and the names of the people who helped you (including internet sources, if applicable) should come in the beginning of the main file as a comment. *Copying others' programs is a serious offense and a deserving penalty will be imposed if found.*

Evaluation:

- To consider for the first evaluation without penalty, you have to submit your program by the due date. If you submit after the **due date** but on or before the **cut-off date**, ***there will be a penalty of 5% on the marks you deserve otherwise.***
- If you do not submit by the **cut-off date**, your program will not be considered for the first evaluation.
- We will do the second evaluation later. This is for those who want to improve their marks obtained in the first evaluation or who do not submit for the first evaluation. There will be

a penalty of 20% for those who are submitting for the second evaluation. The details of the second evaluation will be shared later.