

CS 211 Data Structures and Algorithms Lab
Aug--Nov, 2019
Assignment 7
Total Marks: 10
Due on 26th October

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.

Inputs

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

The input file represents a connected undirected graph with integer weights on edges. Every node (vertex) in the graph is labelled 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 that, 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 and Evaluation

- The program you submit should output `mst.txt` when run.
- The main file of your program should be named as `<roll no>.<extension>`, where roll no. specifies your roll no. and the extension depends on the language you choose (Usage of C/C++ is mandatory for this assignment). Ex: `180040001.c`
- Test well before submission. You may use the attached sample input files for testing. The corresponding output files are also attached. 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 10% (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/internet 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 and allowing others to copy your program are serious offences and deserving penalty will be imposed if found.
- To consider for first evaluation without penalty, you have to submit your program by **26th October (11:59 pm)**. If you submit after **26th October** but on or before **2nd November (11:59 pm)**, there will be a penalty of 10% on the marks you deserve otherwise.
- If you do not submit by **2nd November**, your program will not be considered for the first evaluation.
- We will do the first evaluation after **2nd November**. The marks you obtain will be proportional to the number of correct lines in the output file. There will be a second evaluation for the assignment for the needy. The details will be shared after the first evaluation.