Project Statement

(CS-2009 Design and Analysis of Algorithms – Fall-2023)
Due Date and Time: Monday December 4, 2023 (11:59 pm)

Instructions:

1. Late submission will not be accepted.
2. Project can be done in a group. Max. group size is 2 members.
3. Submit all codes files in a zip folder.
4. Submit report in pdf format.
5. File naming format for report should be "Member1-Roll#_Member2-Roll" e.g. 200123_205478.pdf.
6. File naming format for code files should be "ProblemName_Member1-Roll#_Member2-Roll#.cpp" e.g. Dijkstra_200123_205478.cpp , Cycle_200123_205478.cpp etc.
7. There will be no credit if the given requirements are changed.
8. Your solution will be evaluated in comparison with the best solution.
9. There will be negative marking for not following the submission instructions on GCR.
10. Plagiarism may result in zero marks in the whole project regardless of the percentage plagiarized.

Problem Description:

Provide efficient implementation of the algorithms for the problems given below. Test your algorithms on this dataset https://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html . You are also required to provide a detailed analysis of your algorithms.

1. Normalize the dataset if required to avoid maximum size errors and generate graph that would be passed to algorithms.
2. Find single source shortest path using Dijkstra and Bellman Ford algorithm.
3. Find minimum spanning tree using Prims and Kruskal's algorithm.
4. Detect cycle in a graph if exists any.
5. Sort the fourth column using Merge sort, quick sort and heap sort and store result in a separate file.
6. Analyse the complexity of the algorithms and justify in the report.

Requirements:

Implementation:

• Use any programming language to implement the algorithms

• Provide complete trace of your algorithm. For example, in case of using stack/queue, write complete trace like node insertion and removal for each step in a separate file.

• Make sure to use efficient implementation of the queue as discussed in the class for finding

the shortest path.

• Store shortest paths in a separate file along with the printing on the screen

• Store result of minimum spanning tree on a separate file. Also show MST on the screen.

• Store execution time of all the implementations.

• You can use any library for the visualization of your results.

Report:

Your report should contain:

1. Complete algorithms with time complexity analysis for best/worst/average case.
2. Comparison of algorithms using plots(graphs) of your results. For example, it should contain analysis and comparison of execution time of Dijkstra algorithm in comparison to bellman ford and prims in comparison to Kruskal's algorithm and comparison of merge, quick and heap sort.
3. Discuss implementation structure (stack/queue, type of stack/queue), effect on selection of a particular stack/queue on the time complexity of the algorithm.

Marks Distribution:

▪ Normalization and Graph Generation [10]

▪ Single Source Shortest Path (Dijkstra, Bellman Ford) [15+15]

▪ Minimum Spanning Tree (Prims, Kruskal's) [15 + 15]

▪ Report [30]

▪ Cycle detection in a graph [20]

▪ Sorting Algorithms [10+10+10]

➢ For each of the above algorithms correct implementation with correct results (shortest path/MST/cycle) including traces of the stack/queue contains 60% marks. Write results of all these algorithms in a separate file for demonstration.

➢ Analysis part contains 40% marks.