Design Document

Mohammad Immam

COP 3530

Instructor: Cheryl Resch

For assignment 2-b we expanded 2-a's graph implementation to execute Dijkstra's algorithm in order to find the shortest distance to all vertices from the source.

What graph implementation did you use and why?

I have used adjacency list implementation of graph using C++ vector, where each vertex is aware of other vertices it is connected to. This implementation was decided and approached earlier in assignment 2-a keeping Dijkstra in mind. Doing Dijkstra was slightly easier since it wasn't much work to get a list of vertices from each vertex to look for updates. Also, since the inputs to the graphs were relatively sparse, using adjacency list seemed like a better approach.

What is the computational complexity of printDijkstra, printMST, insertEdge, getWeight?

printDijkstra: Dijkstra's algorithm is a greedy algorithm, selecting local optimum at each level to come up with a global optimum solution. The runtime complexity of Dijkstra's algorithm is O(|E|>|V|) average as it iterates through all edges that goes out from v vertices.

printMST: printMst first sorts the edges in order of their weight. Since I used selection sort, that part is O(v^2). For all edges, generateMST checks for selected edges which is also (v^2). Therefore, the runtime complexity of generate MST is O(v^2).

insertEdge: This uses a hash-map to find its to and from and inserts the edge to the appropriate location. Assuming the C++ map function has operation O (1), insertEdge's runtime complexity is O(1).

getWeight: Since a hash-map was used for keeping the nodes and getWeight collects a data finding the appropriate vertex, assuming C++ hash-map has a runtime complexity of O (1), getWeight should also be O (1).

What was the hardest part of this assignment?

With professor Resch's awesome explanation and algorithm discussion, the assignment felt relatively simple. The hardest part of the assignment personally was to collect and sort all edges in order to perform generateMst with the adjacency list implementation of my graph.

What did you learn from this assignment?

This simplified version of generating a graph structure gave us the idea of building, using and maintaining a graph for any operations on data in a simple, fast and better way. This assignment also reminded me the importance of faster sorts and usage of different data structure according to needs and wants. The assignment was made simple yet crucial and significant for the course to reinforce almost everything that was taught in since the beginning.