Dayoung Lee

DL29923

**EE 360C Programming Assignment 2 Report**

1. Explain the tradeoff between using adjacency matrix representation of graph and adjacency list representation of graph. Give a Big-O analysis of the memory space efficiency of your graph representation in terms of nodes N and edges M.

If we have a sparse graph, it would be better to use an adjacency list because it uses less memory. If we want to know if an edge exists between two nodes, *u* and *v*, we might have to traverse *u*’s entire list. If we have a dense graph, it would be better to use an adjacency matrix because it uses about the same amount of memory compared to adjacency lists, and finding whether an edge exists is faster because we just check A[u][v], which takes O(1).

For adjacency list representation, we will have N adjacency lists. For an undirected graph, there are 2\*M edges represented in the lists. Therefore we have a total of O(N+2\*M), or O(N+M) memory used.

Because we use a NxN matrix to represent our edges, an adjacency matrix uses O(N­2­) memory.

1. Give a Big-O analysis of the runtime complexity and memory complexity of the algorithm you used to find the time optimal route.
2. Write pseudo-code for the algorithm you have implemented to find the capacity optimal route and prove the correctness of your algorithm.
3. Is Dijkstra’s algorithm able to solve graph problem with negative weighted edges? Explain why?