Elba Chimilio
COP3530 – Section 7303
Project 3: Dijkstra’s Algorithm

Project Commentary

It must contain commentary in which you discuss your graph implementation, why you chose it, and the computational complexity of the operations. Discuss the computational complexity of Dijkstra's Algorithm. Discuss any additional data structures used to implement Dijkstra's algorithm (e.g. priority queue). Discuss what you learned from this assignment and what you would do differently if you had to do it over.

- void insertVertex(int vertex) | **Computational Complexity:** O(1)
- void insertEdge(int from, int to, int weight) | **Computational Complexity:** O(n), where n is the number of vertices in the graph
- bool isEdge(int from, int to) | **Computational Complexity:** O(n), where n is the number of vertices in the graph
- int getWeight(int from, int to) | **Computational Complexity:** O(n), where n is the number of vertices in the graph
- vector<int> getAdjacent(int vertex) | **Computational Complexity:** O(nlog(n)), where n is the number of vertices in the graph which executes n times
- void printDijkstra(int source) | **Computational Complexity:** O(mlog(n)), where n is the number of vertices in the graph and m is number of edges in in the graph
- void printGraph() | **Computational Complexity:** O(n*mlog(m)), where n is MAX_VERTICES and m is the number of edges per vertex
- bool Graphs_P3::isVertex(int vertex) | **Computational Complexity:** O(1)