**Assignment 3**

The edges text file is read and then put into a list where each index represents a vertex and has a list of all the edges in it. Because the list contains all the edges and is bidirectional, it will have 2E edges, the space complexity will be O(E). Going through every edge to make the list will have time complexity of O(E).

All the vertices that are connected to the first vertex are checked if they are already in the discovered min heap and this takes constant time because a vertex array is kept full of each of the vertices indexes in the min heap array, so if the element is greater than 0 then it is in the heap. If the element is less than 0, -2 means it is not discovered and -1 means it has been finalised. The space complexity of this vertex will be O(V). If the vertex is in discovered but not finalised, it checks if the new path is smaller than the current one and updates it if needed. The Dijkstra’s algorithm will go through every edge in the worst case, making the time complexity O(E). Since a min heap is being used for discovered, extracting the minimum and maintaining the properties of a min heap as the height of the tree will at most be O(logV) swaps. The smallest element from discovered is then moved to final and the process continues till all the vertices are finalised. To recover the path, an array is used where each index represents a vertex and the element is the vertex that is connected to it to make the shortest path. To recover the path, the vertex parent array is tracked back till it reaches the initial vertex. In the worst case this would be O(E) and the space complexity is O(V). So the overall time complexity of the algorithm is ElogV+2E but since ElogV is bigger E, the time complexity is O(ElogV) and the space complexity is 2V+E making it O(V+E).

For the second task the Dijkstra’s algorithm is run for both initial and final vertices meaning the time complexity would be 2ElogV meaning it’s O(ElogV). Then it goes through each customer seeing what the minimum distance is to find the lucky customer, making the time complexity O(C) and making that list is also O(C). This makes the time complexity O(ElogV+C) but since C can be at max V and assuming the graph is connected, this implies C<E meaning it can be disregarded making the time complexity O(ElogV). Then to print the path, task 1 is used meaning the time complexity is 3ElogV making the time complexity O(ElogV).

Josh Kannemeyer

27845508