[TRANSPORTATION MANAGER]
RETROSPECTIVE

<Chee Ng>

CSC316: Data Structures for Computer Scientists

<cwng@ncsu.edu>

North Carolina State University

Department of Computer Science

<Date: July 23, 2018>

CHALLENGES

In my project 3, the two major challenges are figure it out the bug on Minimum Heap tree and implement Kruskal Algorithm. For minimum Heap Tree, I used print line to catch the incorrect removal. I got my min Heap code from TYPOS but there is a minor issue with it on the bubble down. For the Kruskal algorithm (which is the minimum spanning tree for highway), it took me a day to assemble everything. Especially the Uptree, I was having trouble to implement that in the beginning. In fact, I fixed one line of code that gives me green ball. This project, I got fair enough time to work on project since most of the implementation are not too difficult. I don’t have any issue with input/output or Junit test cases.

DATA STRUCTURES & ALGORITHMS

Did you use the same data structures and/or algorithm(s) you selected in your proposal? Why or why not? (2-3 sentences)

In my proposal, I decided to use graph ADT and list ADT to store the graph for reading the input. The graph wills stored as ArrayList that will contain two vertices and an object of weighted edge. I'm not sure about the time efficient since I have not encounter the time-out issue on Jenkins; however, I know ArrayList is more efficient because it has constant O(1) to access any element on the list. With List ADT, I also can use mergeSort to the sort Minimum highway output. My original plan was to build sortedArrayList but I think merge is actually faster. This project relies on get method very frequent, and LinkedList data structure would not be a good strategy. Since Kruskal algorithm and Minimum Priority Queue are required for this project, I will skip the explanation for this.

IMPROVING EFFICIENCY

Describe at least one way you could further improve the efficiency of your project software, based on topics we have discussed in class. (2-3 sentences)

For this project, I still think ArrayList is the best option because it works pretty well as a graph. For the sorting part, my original plan was to build sortedArrayList so I don't have to sort after created a new list. However, when I compute the time efficiency. I found out that merge sort is actually faster. I am not sure if LinkedList data structure would not be a good strategy, but I think it could help to implement as Adjacency List.

MOVING FORWARD

Briefly describe one thing you will change or improve upon as you work on the next project. (2-3 sentences)

First, I personally would like to thank Adam Gaweda provides his data structure examples TYPOS. It gives me an opportunity to learn how to implement data structure and saves some time for research; for instance, the Graph, Heap and uptree structure. I also learned so much for the print line on console output to view better in Jenkins test. I fixed my bug with this technique. It turns out I have only one line of codes needs to be change. This is my first project that got a greenball in this class. I am very grateful. For project 4, I am going to start it tonight because of time constraint and I also hope I can do well in project 4 like project 3. I am just going to use the same technique to debug project 4.