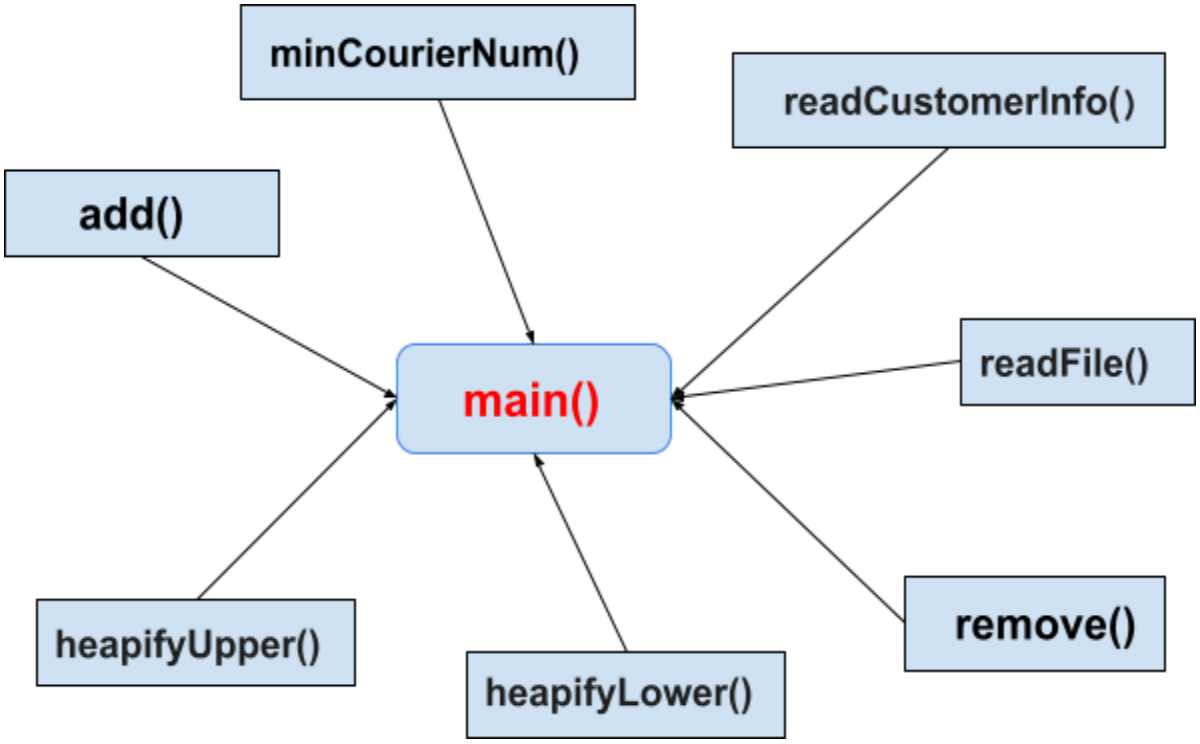
Abdallah Ghordlo
Hussein Abdikarim

CMPE 223 - SEC 05
Assignment/ HW-4

PA REPORT- 4

Problem Statement and Code Design

This assignment is made up of one part, that deals with modeling for an online delivery company. This class takes in a txt file and adds that to a String array line by line, then it moves that information from the txt file into a customer array where it uses these values to determine the average couriers from it with the help of `minCourierNum()`. Our code includes several sub-parts to make it modular. These sub modules are basically defined using a structure chart below.



Implementation, Functionality

How does it work?

By calling various methods, main() first reads the txt file, analyzes every line in that file and saves it in a String array, then it creates a heap and a customer array with a size that depends on the number of customers in the txt-file. After that, it assigns the txt values into the customer array and prints them based on calculations within the method.

1. **minCourierNum():**This function returns the average time the deliveries take with the given number of couriers.The function is first used for getting the minimum amount of couriers necessary for the average time to be equal or less than the average time desired by the user. After getting the minimum number of couriers,the function is called again, but this time it's called with the printStatements boolean as true, which will make the function print lines that outline the process.
2. **readCustomerInfo():**This function takes in a String as input and and returns the customer from the lines of the String array.
3. **readFile():**This function takes in the file name, reads it, and returns a String Array containing the lines of the txt-file.
4. **add():** This function takes a node as input and adds it to the last of the heap and heapifyUpper() is called.
5. **remove():**This function removes the root of the heap and replaces it with the last value of the heap and heapifyLower() is called.
6. **heapifyUpper():**This function takes in an int as index and compares it with the parent if it is greater than the parent, we move it up and make it the parent.
7. **heapifyLower():**This function takes in an int as index and compares the parent with its children, if they are greater than the parent, we exchange the parent with the greater child.

FINAL ASSESSMENTS

This assignment was the hardest assignment we had in terms of figuring out an approach for solving the question. The functions took a lot of trial and error before finally being able to produce the desired output. Nonetheless, it was quite an intriguing assignment, maybe more than the others we would say. We had fun doing this assignment as a group and learned a lot from each other. At this point, we pretty much got acquainted with reading txt-files. That being said, we are very happy with the journey we took in this assignment as a team and look forward to more challenges in the future.