Data Structure Final Project Report

109006243 姚林飛

1 Program Structure

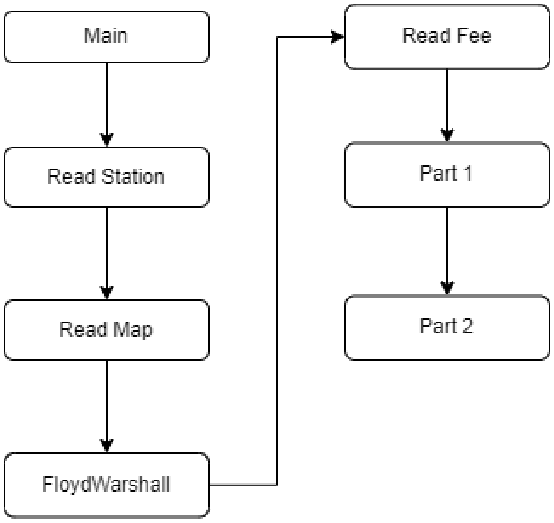


Figure 1: Program flow Chart

1.2 Detailed Description

Figure 1 is the flow chart of main. Firstly, read the input from the text file and execute Floyd Warshall All-Pair Shortest Path algorithm after Read Map to calculate the shortest path. We then process the user input in Part 1, then Part 2 will do additional processing based on the result of Part 1.

2 Part 1

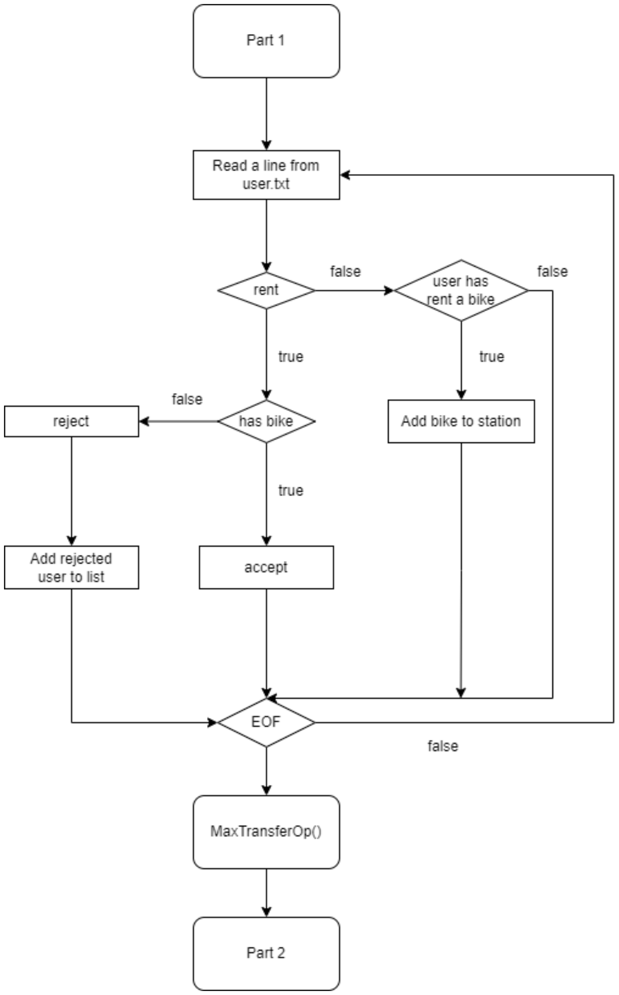


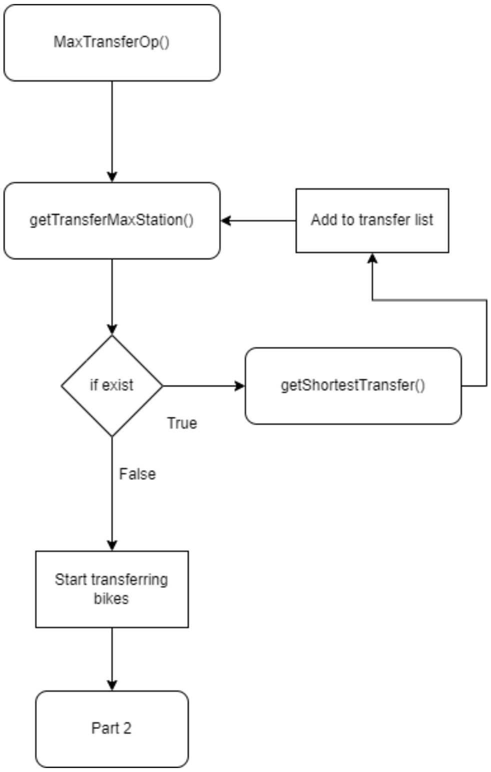
Figure 2: Part 1 Flow chart

Figure 2 is the flow chart of part 1. When we store the bikeID for each type of bike at each station, we will put those ID in the minheap, because when the user rent the bike, we need to always give them the smallest ID and by utilizing minheap, we can keep the operation pop and insert time complexity at O(log n). In addition, there is an array call user to store the status, such as the time of rent and return, the station in and out, and the type of bike rented. Because userID has a maximum of five digits, we can use a fixed array size, and access the ID of user directly.

If there is a situation where there is no type of that bike available, we will output rejected, then add the rejected user to the list for optimization for in Part 2 (We also store the possible revenue IF THAT BIKE EXISTS and the user able to rent it). Also, when returning the bike, we insert the bikeID to the return station.

(OpMaxTransfer() will be later explained in Part2)

3 Part 2



In Figure 2 above, we can see the function OpMaxTransfer(), this function actually calculate the maximum transfer of each type of bike of each station that when we transfer all of them, all the accept request in Part 1 will be still valid. The function OpMaxTransfer() calls the getTransferMaxStation(), this function will return the highest revenue of that station (Calculate earlier in part 1), then getShortestTransfer() find the shortest transfer station from other station to this station, adding to transfer list.

Proceeding to part 2.

Now, we have the transfer list already, to be efficient, we start to transfer at time zero and start the normal rent, return operation. If we got a Reject status, we compare whether if we wait for the transferring bike (if exists) or whether if we give them a discount.

Open_basic3 Part1 is 1680165, when use all the operation combined in Part 2, the result is 1757715.