Data Structure Project 3

Long Nguyen

May 2019

1. Project:

- The project contains a main.cpp, main.o, hashmap,cpp, hashmap.o, and 3 hotels csv files.
- The main function we implemented in the project is creating 2 hash tables to store the data from the csv files. This will make the time complexity of searching and inserting faster.

2. Steps:

- We create 2 hash maps:
  - 1 to store the values with the key as “hotel,city”.
  - The other to store the values with the key as “city”.
- We also store the values into a linked list named: “hotels”. The purpose is to keep track the storage of values at all time (add, delete). We will later use this list to create the priority queue where we alphabetically sort all the values and write into a new .txt file.
- The program asks for command which will only be valid if it contains one of the five word (find, add, delete, dump, allincity). We make sure of this by using a function which will return the position of the command word (which should always be 0 as the start of commandline).
- Depends on the command word, we assign each to a different case to use the switch(case) function.

- Case 1: if there's a "find" command, use the search function of the hashmap.
- Case 2: if there's a "add" command, first check if the given string is valid by counting the number of commas (should be 5). Then, check if the word is already in the hash table. If yes, return warning. If no, use the insert function and print out "insertion succeeded".
- Case 3: if there's a "delete" command. Again, first check if the value already exists. If yes, return warning. If no, use the remove function.
- Case 4: if there's a "dump" command, first create a min priority queue. Import the data from hotels list to pqueue. Then, create a new file and write into it the values from pqueue.
- Case 5: if there's a "allincity" command, use the searchall function. It will print out all the hotels if they exist. If not, it will print out "no record found…".