Data Structures Project 3

Jakub Podmokly
Nhi Pham Le Yen
Lab Section: Wednesday 1:15pm - 3:15pm

May 2019

1 Project

In this project, we develop a hotel-finder application termed **hotelFinder**. Our program includes 3 classes:

- **class Hotel:** stores hotel name, city name, stars, price, country name, address.
- **class HashHotel:** stores the key and the value, and has the methods getKey(), getValue(), and setValue().
- **class MapHotel:** stores the array hotelArr, size, capacity, and has the methods insertHotel(), searchHotel() and deleteHotel().
- **class MapCity:** stores list buckets, size, capacity, and has the methods insertHotel(), searchCity(), and deleteHotel().
- **HotelFinder:** handle file input, and input from user with commands including *find, add, delete, allinCity, dump, quit.*

2 Data Structures

The data structures we use in the project are hashing and priority. We deal with collision by linear probing and separate chaining, and we handle the command *dump f* with priority queue to sort the entire content alphabetically and save data to file f.

- We use linear probing in the class MapHotel. When we insert a new element, which is a new hotel, we use the hash function to find the index. If the position at the index is not empty, we continue to move until we find the very first empty slot and put the element here. It is also the same for delete and search a hotel. We also need to first find the index by using the hash function and scan linearly one by one to find the element that matches the key.

- We use separate chaining in the class MapCity. When we insert a new hotel, we find the index of the bucket using hash function, and put it in the list at this index. Hence, for function search and delete, we also first find the index of the bucket, and scan the list at this index to find the elements that have the matching and print (if search) or remove them from the list (if delete).