Assignment 3
Liyan Ibrahim, Wajahat Mirza
9/12/2019

Program Description:

HotelFinder is a program that stores, retrieves, deletes and manages the records of hotels in cities from a text file and allows users to perform the following functionalities:

1. Find a Hotel: which will display all the different features of the required hotel
2. Add a Hotel: which will allow the user to add a hotel to the records
3. Delete a Hotel: which will allow the user to delete a hotel record from the entirety of the program
4. Dump into a File: which will allow the user to add all the records of the hotels into a certain file alphabetically
5. allinCity of a city: allows the user to find all the hotels and the corresponding details for a certain city
6. Quit: terminates the program

HotelFinder.cpp

Our program sustains 2 data structures in order to perform all the functionalities: A HashTable and a Trie. A trie is used to make the functionalities it accommodates (such as allinCity) more time-efficient as it is of order big Oh (m) in which m is the length of the input as a pose to binary search which has big Oh (nlogn)
This file contains the *Class Hotel_Node* which forms a node with all the different features of the hotel (Hotel name, City, stars, price, country name and address).
The class *Hotel_HashMap* which sustains the Hashtables by maintaining a list of pointers to Hotel_Node and working through the following functionalities:

1. Constructor: initializes private attributes
2. HashCode: develops the hashCode which is the sum of all the ascii codes of the Key (hotel name, City name) multiplied by the position of the letter in the key to the power of 10
3. Insert: which takes all the features of a new user added hotel. It initializes the Key as the hotel name, city name. It will then add the hotel and all the features into the Hotel_Hash and increase the size of it
4. Search: Has an iterator that iterates through the Hotel_Hash and finds the Hotel that matches the key given. Additionally, there is a counter that is incremented that counts the number of comparisons made as well as the execution time by calculating the start and end time and subtracting them

5. Removes: takes the key as an argument and iterates to find the Hotel and features corresponding to that key and erases it from the two maintained data structures and decreases the size
6. get_size(): gives you the size of the HashMap
7. Dump: a vector "dump" is made that will store all the hotels and the features from the Hashtable into the vector. The vector is then sorted. The file the user inputs is opened for writing and the vector is read into the file line by line. The file is then closed
8. Destructor: deletes dynamically allocated memory

Main function:

Opens the file for reading and checks if the file opened successfully, otherwise throws an error message. A while loop is used to count the number of lines and we made a function that finds the next largest prime number smaller than a certain value and this function is used to find the max size.

myHotel_hashmap is the object of the Hotel_hashmap. A while loop is used to read the content of the file into the hashmap.

The user is then aked for input and based on input, the appropriate functions are called to be performed on the hashtable.

AllinCity function is the only function that used the Trie but every time a hotel is added/ deleted, it is also deleted from the Trie data structure.

Hotel_Functions.cpp

This file stores the Trie which needs 3 different classes with different functionalities to accommodate the function: Hotel, Node, Trie

Hotel class stores all the names of the hotels and features and has a function that allows you to display them

Node class has a hotel list and map of edges and will allow you to add hotels and display hotels

Both of these classes are used for the *Trie class* which will allow you to insert Hotels, find hotels in a certain city and would also allow you to remove a hotel in a certain city

This file further has two general functions. allinCity which takes the city and the Trie as arguments and calls the find function of the Trie which will then display all the hotels of a certain city. The second general function is deleteHotel which deletes a certain hotel and functionalities from the trie and maintains trie properties.