Lab5: Hash Table

Haoqi Wu

October 23, 2019

Description

This lab is designed to give you practice working with hash table. This is an individual assignment.

- **You may not share code with other students**
- You should implement your own hash table data structure and may not be confused with the `java.util.HashTable` class available in the Java standard library.
- This lab is designed based on Timothy Colburn's work!!

Specification

Overview

Your task is to complete the definition of a `HashTable` class using *separate chaining method* to handle collision. Specifically, you will:

- Implement `hash functions` for *integer* and *string* keys
- Implement a hash table `search` operation
- Implement a hash table `remove` operation

Preliminary

Download the zipped project archive — **Lab 5.zip**, and import it into your IDE as an existing project.

Open the `hashtabletest` package and run the `TestFrame` class. Clicking the insert button will insert new elements into the hash table. Note that the integer and string keys always hash to zero — you will fix this later.

Open the `hashtabletest` package and run the `HashTableTest` class. Note that the `testSearch` and `testRemove` tests should fail — you will fix this later, too.

Your Job

You need to complete the following methods in the `hashtable.HashTable` class:

- `integerHash` and `stringHash` methods: to exhibit good hashing performance for integer and string keys
- `search` method: to find and return the hash table element for a given key

- `remove` method: to remove a hash table element once it is found

Test

`integerHash` and `stringHash` , test them by running the `TestFrame` class.

- You should be able to insert random integers, random strings, and permuted strings with good hash function performance.
- Good hash function performance would generally (but perhaps not always) show **1.0 or less for a standard deviation** from the mean chain length after `CAPACITY` (in this case, 11) inserts.

`remove` and `search`, test them by running the `HashTableTest` class.

- When you have successfully implemented these operations this class will run without errors.

Submission

Deadline: In class / 23 Oct, 2019 18:00, any uploads after 25 Oct, 2019 18:00 wil get **ZERO** points.

Create a zip file named **YourStudentID.zip** that contains your code project and **upload your zip file to the FTP server**.