C S 272/463 Introduction to data structures
Fall 2019
Lab 14: Hash

1 Learning objectives

Objective 1 (hash), Objective 2 (recursive thinking), Objective 3 (searching), Objective 5, Objective 6, Objective 7 in course syllabus.

2 Requirements

2.1 Detailed instructions for program design and implementation

1. (60 points) Please design a hash table `EmployeeTable` to implement the open-address hashing data structure that we discussed in class. `EmployeeTable` is used to store information of employees. The employee structure is the data structure you implemented in lab 2. For this lab, each employee should also have an int variable `employee_no`, which is a unique identifying value of an employee. Implement the following methods for this class and put the code to **EmployeeTable.java**.

   (1) (5 points) Please include proper instance variables in `EmployeeTable`.
   (2) (5 points) Please include proper constructors.
   (3) (5 points) Please design the hash function to be

   ```
   the hash code of the employee no % size_of_array_for_keys)
   ```

   (4) (12 points) Add a new employee `e` into the hash table.

   ```
   public void put(Employee e)
   ```

   (5) (13 points) Remove a given employee with employee id `emp_no` from the hash table. Return false if an employee with `emp_no` does not exist in the hash table; Otherwise, remove it and return true.

   ```
   public boolean remove(int emp_no)
   ```

   (6) (10 points) Find the index of the given employee id `emp_no`. Return the index of the employee in the array if the employee with the given employee no exists in the hash table; otherwise, return -1.

   ```
   public int search(int emp_no)
   ```

   (7) (10 points) Put your test cases to the main method. You need to design test cases to test your program **thoroughly**. If your test cases cannot cover some important conditions, points may be deducted.

2. (40 points) The `HashMap` java class implements a hash table. Please design a main function to utilize the functions of this data structure. In particular,

   (1) (5 points) You need to use `Employee` data structure that you have implemented in lab 2. You can reuse your code in your lab 2. (Hint: you can set the key to be employee no and the value to be an instance of an employee)
   (2) (10 points) Add all the employees in the data file (`https://www.cs.nmsu.edu/~hcao/teaching/cs272/lab/lab1/core_dataset.csv`) to a `HashSet` structure. You can reuse your code in your lab 2.

(3) (8 points) Remove employees with nos `1112030816`, `1111030148`, and `602000312`.

(4) (8 points) Search employees with employee no `1501072093` and `1111030148`.

(5) (9 points) Show the size of the data structure that keeps the employees after the adding, removal, and search operation.

(6) * For your reference, an example to use `HashMap` can be found from `https://beginnersbook.com/2013/12/hashmap-in-java-with-example/`

3 Note

- **Specifications** for all your classes and methods:
  Please properly explain (1) the functionality of the methods, (2) the parameters, (3) the return values, (4) the pre-conditions if there is any;
  Please use inline comments, meaningful variable names, indentation, formatting, and whitespace throughout your program to improve its readability.
- You can (but are not required to) design and implement other facilitating methods (E.g., other get and set methods, toString method) to finish the implementation of the required methods.

4 Submission

Submit through canvas a zipped file containing your java file(s) (not `.class` files).

5 Grading criteria

(1) The score allocation is already put in the questions.

(2) Please make sure that you test your code **thoroughly** by considering all possible test cases.

(3) 5 points will be deducted if submitted files (including files types, file names, etc.) do not follow the instructions.

(4) At least 20 points will be deducted if your code cannot be run on CS servers.