

GTU Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 6
Due date: 23 May 2020 – 23:55

This assignment consists of 4 questions.

Q1:

- * A is an ordered integer array with 10 elements from small to large
- * B is an ordered integer array with 10 elements from large to small
- * C = {5, 2, 13, 9, 1, 7, 6, 8, 1, 15, 4, 11}
- * D = {'S', 'B', 'I', 'M', 'H', 'Q', 'C', 'L', 'R', 'E', 'P', 'K'}

Apply the following sorting algorithms for the given arrays.

- Shell Sort
- Merge Sort
- Heap Sort
- Quick Sort

Show your works step by step and make explanation. Express how many comparisons and how many displacements have been made for each. Write your solution by latex or word and upload it as a *StudentNumber.pdf* file.

Q2:

Implement Quick Sort and Merge Sort algorithms. Suppose the data is stored in a linked list. You should use LinkedList class in Java.

In the book, there are methods for 7 algorithms which are

- Selection Sort
- Bubble Sort
- Insertion Sort
- Shell Sort
- Merge Sort
- Heap Sort
- Quick Sort

Compare your implementation with the methods in the book. To do this, you must create 100 (20 of each) random and 5 (1 of each) sorted arrays/linked lists with a length of ten thousand (10000), forty thousand (40000), one hundred thousand (100000), one hundred and fifty thousand (150000) and one hundred and eighty thousand (180000). Run all 9 methods for each, save run times and expected run times in excell, and draw graphs for each method. Run times and expected run times should be drawn on the same graphic.

Q3:

Design a library automation system. The system has two types of users: administrators who can update information and users who browse books.

Book search is a public feature. To update the information, administrators have to enter the system with a password (a single general password is sufficient).

While designing this system, realize the following: In the system, for each book, the name of the author, the title of the book, the location of the book (for example, if a book on the 5th shelf in the 3rd corridor, such as "c3s5.1534" can be used) should be stored. Information about the books should be stored in nested map and set data structures. For the outermost map, the author name is used as a key, the value is another map whose keys are book names. In the inner map, the values are sets showing the location (or locations if there is more than one copy) of that book.

The system should include at least the following functionality:

1. When searching by author name, all books of the author in the library will be listed on the screen. Then, whichever book the user chooses, the location(s) of that book will be displayed.
2. When searching by book title, author name, location and status will be shown.
3. When the administrator enters the system with a password, he will be able to add books, delete books, update information.

Q4:

Implement KWHashMap interface in the book using the following hashing methods to organize hash tables:

- a) In chaining is a technique, each table element references a linked list that contains all the items that hash to the same table index. Use chaining with binary tree instead of linked list.
- b) In open addressing, if the table entry is full, subsequent probe locations are calculated using linear probing ($\text{hash}(x) + i$) or quadratic probing ($\text{hash}(x) + i^2$) methods. Use an alternative probing method called double hashing where a second hash function is used to calculate subsequent probe locations ($\text{hash}(x) + i * \text{second_hash}(x)$).

Compare the performance of your implementations with HashTableOpen and HashTableChain classes in the book. Note that you should generate large maps of several entries and perform several random operations to compare the performance of the methods. Report your experimental evaluation using tables and graphs.

RESTRICTIONS:

- Use only specified data types
- Can be only one main class in each question
- Don't use any other third part library

GENERAL RULES:

- For any question firstly use **course news forum** in Moodle, and then the contact TA.
- You can submit assignment one day late and will be evaluated over sixty percent (%40).

TECHNICAL RULES:

- Use given CSE222-VM to develop and test your Homeworks (**your code must be working on CSE222-VM**), CSE222-VM download link will be given on Moodle.
- Implement [clean code standards](#) in your code;
 - o Classes, methods and variables names must be meaningful and related with the functionality.
 - o Your functions and classes must be simple, general, reusable and focus on one topic.
 - o Use standard [java code name conventions](#).

REPORT RULES:

- Add all [javadoc](#) documentations for classes, methods, variables ...etc. All explanation must be meaningful and understandable.
- You should submit your homework code, Javadoc and report to Moodle in a "studentid_hw3.tar.gz" file.
- Use the given homework format including **selected parts from the table below**:

Detailed system requirements	
Use case diagrams (extra points)	
Class diagrams	X
Other diagrams	
Problem solutions approach	X
Test cases	X
Running command and results	X

GRADING :

- **No OOP design:** -100
- **No interface:** -95
- **No method overriding:** -95
- **No error handling:** -50
- **No inheritance:** -95
- **No polymorphism:** -95
- No javadoc documentation: -50
- No report: -90
- Disobey restrictions: -100
- **Cheating** : -200
- Your solution is evaluated over 100 as your performance.

CONTACT :

- Teaching Assistant : Başak Karakaş