CS401 SW Project for Spring 2024

1. Project Description

Create a list with a data file and/or user input.
Must program with a main application program (CS401prj.java) and user defined class (*.java file without main method) that provides sorting and searching operations.

Sorting

Your project compares complexities of ***sorting*** algorithms and ***searching*** algorithms.
Once a list is created, let user chooses two sorting algorithms between one simple sorting algorithm (selection sort, insertion sort, bubble sort) and one $O(N\log_2 N)$ sorts (Quick sort, Merge Sort, Heap sort). After the list is sorted, print count numbers of comparisons in the algorithm so we can conclude which algorithm performs better with the total counts. Discuss your program's result with theory of the algorithm you selected with the Big-O.

Choose only one algorithm from each sorting algorithm group (refer chapter 11)

You must implement the sorting algorithm and **do not use Java library for sorting class**.

Searching

Comparing search algorithms works the same way: do both linear search with the original list and BST with the sorted list (for binary searching, you may run sorting option first). Create a hash function list with the same data and check the complexities to compare with linear searching, binary searching, and hash function searching.

Other functions (Use another user defined class of your own)

Status of data quantity – Your program provides a way of how many data in the structure to the user. The user (TA) can understand the quantity of the data set immediately with this feature.

Adding data – Your program must accept user input (by keystroke) of data to add additional data. (if user enters invalid data type for your original data, provide a proper message for correction)

Delete data – Your program also deletes data what the user wants. The existing data set should be adjusted after deleting a data or multiple data.

Update data – Your program also updates data what the user wants. The existing data set should be adjusted after updating a data or multiple data.

Restore data – Your program also restores the data what the user deleted but wants to restore it/them. The existing data set should be adjusted after restoring a data or multiple data.

Analyze data – Your program provides analysis report of data and/or operations what the user have done. This feature will be dependent on your brilliant idea. Brainstorm how and what your project provides the analytic report and implement. Describe why your report is helpful for the user.

All above functions should be designed, implemented, and tested by your own idea.

You may use interface to define abstract methods of child classes and inheritance.

Depending on your SW plan and design, this SW project must use at least two classes (one for application and another for a user defined class OR multiple user defined class of yours). The more will be better but it does not hurt grading by your TA.

2. Project Requirements

- Your project has a menu to select each algorithm (searching, sorting, and other functions).
- Your project can accept any data types: integer number list, float numbers, or string.
- When a list is created, print out the list.
- Size of data: at least 100 (the more is better)
- User input any data and your project needs to print sorted list and total count of operation (or comparisons)
- Display data by each menu: well organized format (free design but do not list all values with a single column).
- Any implementation structure (array or linked list but **do not use ArrayList** – 30% deduction if you use the Java's ArrayList) is OK

- No standard interface is given.
- If your project does not include at least one user defined class for sorting and searching, and others, a 20% deduction applies (OOP requirement).
- Extra credit opportunities – add more brilliant functions/features. The maximum extra credit will be up to 10%.

Programming language: JAVA ONLY

What to submit?

Submit softcopy through the Course BB

1. Program files
    a. Source codes (with detail comments)
    b. Java Bytecodes (compiled *.class file to run directly)
2. Documents based on Software Development Life Cycle
    a. Problem specification – What problems are solving?
    b. Software specification - What functions are there?
    c. Design diagram document (including UML diagram and flow charts or pseudo code)
    d. Operational document (user's manual: how to run your program, what is expected result or screen shots)
    e. Testing document (your own created one) with input data file
    f. Debugging note (if used)
    g. Self-evaluation notes - list what parts of the project requirements are not implemented (if exist) – TA will not test that part(s) but apply a proper deduction. TA will deduct bigger point off for non-listed functions or operations.
    h. Project management/schedule – daily progress plan
    Hours per each task to be done.
3. Complexity analysis based on your results with the theory you learn.
4. Submit a single zip file containing all the documents and organize them into two folders.

- All documents except source codes formats: either PDF or MS Word

For the zip file:
Folder-1 should be named "Project Source Code" (should contain java files, class files, Testfile, and Operational Document).

Folder-2 should be named “SD Files” (should contain Problem Specification, Software Specification, Design Diagram Document, Testing Document, Debugging notes, Self-evaluation notes, Project Management/Schedule).

Final Project Due Date: ***Blackboard submission deadline***

- No late submission is allowed.
- No correction or replacing submitted file are allowed after your original submission.
- No email submission is accepted in any situation.

- If any questions, **must contact your TA first.**
- No demo or re-evaluation after final exam. The final exam means END of the semester. No more extra efforts can be allowed by the department.

- **Everyone must read the description and instruction carefully.**

  If you misunderstand or forget any, it’s your responsibility.