CS401 Project

1. Project Description

Create a system to compare sorting and searching algorithms. Your project must read a data file(s) to create an UL. Also let a user manually update (add, delete, adjust value) the data in the UL.

Sorting

Your project compares complexities of ***sorting*** algorithms and ***searching*** algorithms.
Once an unsorted list (UL) is created, use three sorting algorithms, bubble sort, merge sort, and heap sort, to sort the list. Use at least three different sizes of data:

1. Size N <= 50
2. Size N <=500
3. Size N > 1000
4. And more if you want to try

After the list is sorted, print count numbers (or timing duration) of comparisons in the algorithm so we can conclude which algorithm performs better with the result.

Searching

Comparing search algorithms works the same way: do both linear search with the original list (UL) and BST with the sorted list (for binary searching, you may run sorting option first).

Extra credit (5%): if you add hash table function for searching and analysis with above two searches.

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

GUI menu is not required but if your project runs as full GUI (mouse can choose project menu) frame and if your GUI is easy to operate, your project may be qualified to earn extra credit (up to 5%). Command window GUI is not considered as a full GUI. Since Java is a programming language, you may do either applet version or application version.

2. Project Requirements

- Your project has a menu to select each algorithm.
- Your project can accept any data types: integer number list, float numbers, or string.
- When a list is created, print out the list.
- User input any data and your project needs to print sorted list and total count of operation (or comparions)
- Any implementation structure (array, arraylist, or linked list) is OK

- No standard interface is given.
- If more creative functions are implemented well, extra credit points can be earned (up to 5%).

Programming language: JAVA ONLY

What to submit?

Submit softcopy through the Course BB

1. 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. Source codes (with detail comments)
e. Operational document (user's manual: how to run your program, what is expected result or screen shots)
f. Testing document (your own created one) with input data file
g. Debugging note (if used)
h. Future improvement document (if available)
i. Project management/schedule – daily progress plan
Hours per each task to be done

2. Complexity analysis based on your results with the theory you learn

- Compress your packages with a **ZIP** format file only
- File name should include your Last and First name:
  - e.g. **Course#_proj_seat#_Harry_Potter.zip**

- All documents except source codes formats: either PDF or MS Word (not *.docx)

Final Project Due Date: ***November 24th, 2019 Sunday*** 23:59 or earlier

- Late submission penalty: 10% in every 2 hours for 12 hours
  - e.g: submit on 11/25 3:59 AM – 20% penalty
  - e.g: submit on 11/25 12:01 PM or later – No accept

- TA will schedule your project demonstration during day time or during lab (11/22 Friday if your project is completed earlier) – for out of state people no demo is required but if the project does not run by your TA, you need to set a schedule with Skype demo

- 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.