

## CS401 Project

### 1. Project Description

Create a list by a data file and/or user input.

#### 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  $O(N\log_2 N)$  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 algorithms.

#### 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.

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

### 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.
- Size of data: at least 100
- 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) 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

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

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 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.