C S 272/463 Introduction to data structures
Fall 2019
Lab 13: Heap and Search

1 Learning objectives

Objective 1 (heap), 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) Implement the following methods for a min heap (i.e., the top element of the heap is the smallest element) by using the `ArrayList` java class to store the heap elements.
   Put all your methods to **MinHeap.java**.

   (1) (20 points) Add a new element `e` into the heap.

   ```
   public void add(int e)
   ```

   (2) (20 points) Get and remove the top element of the heap.

   ```
   public int remove()
   ```

   (3) (10 points) Get the top element of the heap.

   ```
   public int top()
   ```

   (4) (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) Please design the binary search function to search an element `e` in an array `A`. Assume that all the elements in array `A` are useful elements, and the values in `A` are ordered in ascending order.
   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.
   Put all your method and test code to **search.java**.

   ```
   public static int binarySearch (int[] A, int e)
   ```

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.