Tutorial/Lab 9 - S o r t i n g

Aim

This tutorial/lab aims to further the understanding of the topics covered in Week 9's lecture. Meanwhile, it aims to get students familiar with the question types that might be appeared in their future test.

Exercise 9.1 Bubble Sort

Suppose a list is {2, 9, 5, 4, 8, 1}. After the first pass of bubble sort, the list becomes _____

Exercise 9.2 Max Heap

21.Indicate the number of possible values that can be put in the '?' place to maintain the max heap_____

```
   30
  / \
 "?"  29
 / \
26  27
```

Exercise 9.3 Merge Sort

Write the following two generic methods using merge sort. The first method sorts the elements using the Comparable interface and the second uses the Comparator interface.

public static <E extends Comparable<E>>
 void mergeSort(E[] list)
public static <E> void mergeSort(E[] list,
 Comparator<? super E> comparator)

Hints:
● Given the list is {2, 3, 2, 5, 6, 1, -2, 3, 14, 12}
● Leverage Java's generics to full in the methods that can sort any type of objects using Comparable and Comparator.
● Implement the merge sort algorithm recursively by dividing the array into two halves, sorting each half, and then merging them back together.

Exercise 9.4 Insertion Sort

Write the following two generic methods using insertion sort. The first method sorts the elements using the Comparable interface, and the second uses the Comparator interface.

public static <E extends Comparable<E>>
 void insertionSort(E[] list)
public static <E> void insertionSort(E[] list,
 Comparator<? super E> comparator)

Hints:
- Given the list is {2, 3, 2, 5, 6, 1, -2, 3, 14, 12}
- Leverage Java's generics to full in the methods that can sort any type of objects using Comparable and Comparator
- Understand the insertion sort algorithm's mechanism, which involves building the sorted list one element at a time by comparing each new element with the ones already sorted and inserting it in the correct position.