COMP2011 Data Structures

Lab and Tutorial Seven (Total Marks: 100)

Deadline 11:59pm 6 Nov 2014

1. Download the sorted linked list Java programs from the Blackboard. Add the following functions in Class SortedListApp:

   (1) Create two sorted linked lists in main() and call mergeList() as described below.

   (2) Add the following method:

   public static SortedList mergeList(SortedList list1, SortedList list2)

   This method can merge two sorted lists into a new sorted one, and return a reference of the new list. For the practice purpose, you **cannot use insert()** that has been implemented in Class SortedList; instead, your program should **go through each link item** in both lists, and **directly link them** into the new list in order.

2. Download the "mergeSort.java" program from the Blackboard. Add a new method called *display-segment()* in Class DArray. The method *display-segment()* can display the **sorted** array segment and their corresponding "lower bound" and "upper bound" in each individual step (you can insert this into an existing method to make it work).

For example, given the following array:

64   21   33   70   12   85   44   3   97   24   51   40

after *display-segment()* is used somewhere in DArray, the output will be:

```
Step 1: 0-1    21:64:
Step 2: 0-2    21:33:64:
Step 3: 3-4    12:70:
Step 4: 3-5    12:70:85:
Step 5: 0-5    12:21:33:64:70:85:
Step 6: 6-7    3:44:
Step 7: 6-8    3:44:97:
Step 8: 9-10    24:51:
Step 9: 9-11    24:40:51:
Step 10: 6-11    3:24:40:44:51:97:
Step 11: 0-11    3:12:21:24:33:40:44:51:64:70:85:97:
```

Note: The sequence from the solution of the book (shown in the slide) is wrong. Think about why.

What to submit:

(1) The report that records the output of your program

(2) The java source code of your program