CSCI 241, Fall 2012 Assignment #4: Sorting (30pts).

Summary

- **Goal**: This assignment is worth 10% of the grade for the course. In this assignment you should become comfortable with sorting sequences, without having to write any actual code. Note that questions like these are fair game on the final exam, so you may want to play with more examples on your own.
- **Collaboration policy for this homework**: For this homework, you may not collaborate with another student, although you may work together to understand the course material in a general way.
- Due time: **9 AM in class, Wed, Nov 28.**

Tasks

1. (4 pts) Show what this array looks like after each major step of in-place **selection sort**:

17 3 4 9 18 26 8 6 15

(0.5 pt) Step 1:
(0.5 pt) Step 2:
(0.5 pt) Step 3:
(0.5 pt) Step 4:
(0.5 pt) Step 5:
(0.5 pt) Step 6:
(0.5 pt) Step 7:
(0.5 pt) Step 8:

Final array: 3 4 6 8 9 15 17 18 26

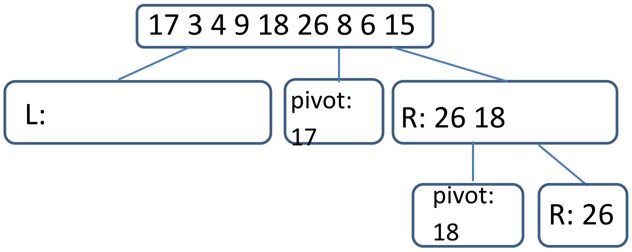
2. (4 pts) Show what the same array looks like after each major steps of in-place **insertion sort:**

17 3 4 9 18 26 8 6 15

(0.5 pt) Step 1:
(0.5 pt) Step 2:
(0.5 pt) Step 3:
(0.5 pt) Step 4:
(0.5 pt) Step 5:
(0.5 pt) Step 6:
(0.5 pt) Step 7:
(0.5 pt) Step 8:

Final array: 3 4 6 8 9 15 17 18 26

3. (11pts) In-place **quick-sort**. Alternately use the first element and the last element as pivot (that is, uses the first element as pivot in the first partition, use the last element as pivot in the second partition, so on and so forth.).
    a. (5pts) Follow the format to complete the tree.



    b. (2pt) Based on the quicksort algorithm given in class, what's the left subarray (the portion on the left of the pivot) after the 6th partition operation?

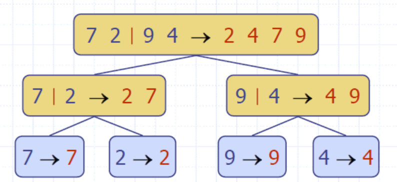
    c. (3 pts) Same as (a), but this time use 18 as your first pivot; in subsequent steps, pick a random pivot to partition. Draw a tree different from the answer you gave in (a).

    d. (2pts) Do the quicksort but this time be the adversary: Choose the worst possible sequence of pivots. Why you think it's the worst case?

4. (5 pts)
    a. (3pts) Execute **merge-sort** by drawing a merge-sort tree as we depicted in class on the same array (see the given example. Please remove this example from your submission.). When you are dealing with a sub-array of odd size, split it as

evenly as possible, with the left half of the array being on element longer than the right half.

17 3 4 9 18 26 8 6 15



b. (3pts) Circle the node representing the 10th recursive call on mergeSort in your tree. Which two sorted sublists are merged in this call?

5. (4pts)

a. (3pts) Show what this array looks like after each major step of in-place **bubble sort**:

17 3 4 9 18 26 8 6 15

	Pass1	Pass2	…
17			
3			
4			
9			
18			
26			
8			
6			
15			

b. (1pt) How many comparisons did it perform in pass2?