CSE 326 DATA STRUCTURES
HOMEWORK 7

Due: **Friday, March 5, 2010** at the beginning of class. Please put your quiz section in addition to your name at the top of your homework.

Problem 1. Selection Sort

For this problem please use pseudo code from slide 8. Assume that *Find index of smallest element* loops from left to right over the unsorted portion of the array.

(a) Sort the sequence 3, 1, 4, 1, 9, 5, 7, 2, 5 using selection sort. You should show the results of all steps that involve swapping elements, with enough text so that someone reading your solution can follow what you're doing.

(b) What is the running time of selection sort if all elements are equal?

Problem 2. Insertion Sort

For this problem please use the insertion sort code on p. 249 in the textbook.

(a) Sort the sequence 3, 1, 4, 1, 9, 5, 7, 2, 5 using insertion sort. You should show the results of all steps that involve swapping elements, with enough text so that someone reading your solution can follow what you're doing.

(b) What is the running time of insertion sort if all elements are equal?

Problem 3. Quicksort

Sort 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5 using the quicksort algorithm given by the book - using median of three partitioning and a cutoff of 3. You should show the results of all steps that involve swapping elements, with enough text so that someone reading your solution can follow what you're doing. The problem notes that there is a "cutoff" value - this refers to the idea of running insertion sort on subarrays that are as small as the cutoff value. You don't need to show any steps of insertion sort; simply assume it sorts the values.

Problem 4. Decision Trees (Parts a-e)

Each of the sorting algorithms we discussed in class, with the exception of bucket/radix sort, can be translated into a specific decision tree. In this question, you'll give the decision tree for three sorting algorithms for a very small sized input. Assume you have an array with exactly three keys: [a, b, c]. Assume there are no duplicate keys. For each of the following sorting algorithms, give the full decision tree. Draw your decision tree in the style of slide 43 from the sorting lecture (with possible orderings in each node, and each branch specifies a comparison). Be sure to draw an edge for every comparison that the algorithm does, regardless of whether there are multiple orderings possible at that point or not. Feel free to annotate your tree with other information such as the current state of the array at that point in the algorithm if you find it helpful to do so.

(a) selection sort (use the same code you used from Problem 1 above)

(b) insertion sort (use the same code you used from Problem 2 above)

(c) merge sort (use the code on p. 260 and p. 261 in the book)

(d) Considering only number of comparison operations, which of these three algorithms is the most efficient for a 3-element array? Explain why in terms of your decision trees.

(e) Considering only number of comparison operations, which of these three algorithms is the least efficient for a 3-element array? Explain why in terms of your decision trees.