CSE 373 -- Spring 1999 -- Assignment #7

(*due Friday, May 28*)

1. **Consider the following list: `29 55 43 12 31`**

   (a) List all of the inversions in this list.

   (b) Perform a series of adjacent swaps one at a time to get the list in sorted order. Show the resulting list after *each* adjacent swap.

2. **Given the same input array as above, simulate the *in-place* Heapsort step-by-step in array form (you may want to draw heap pictures to keep it all straight, but we're only grading the array's values):**

   (a) Show the array after `BuildHeap()` is applied to it (assume that there is no sentinel value used -- *i.e.*, 29 is the root node before `BuildHeap()` is run).

   (b) Show the array after each step of the sort takes place (*i.e.*, after each delete operation).

3. **Give a seven-value worst-case input for a 2-phase Shellsort with increment sequence: *`n-1, 1`*. Use the integers 1-7 without repeating any value. In one sentence explain why it's a worst case.**

4. **Give a seven-value worst-case input for an implementation of Quicksort that uses the middle element as its pivot. Use the integers 1-7 without repeating any value. Assume that the middle element for a list of even length is defined to be the lower of the two middle elements (*e.g.* the middle element of "`1 2 3 4`" would be "`2`"). In one sentence, explain why this input is a worst-case.**

5. **Early in the quarter, we informally (but convincingly) reasoned about the running time of various search algorithms. Now we'll do it a bit more formally.**

   (a) Write a recurrence relation expressing the running time of recursive binary search.

   (b) Solve the recurrence relation using repeated substition or telescoping to prove its O(log*n*) running time. Be sure to show all your work.

   (c) Write a recurrence relation expressing the running time of recursive linear search.

   (d) Solve the recurrence relation to prove its O(*n*) running time.

6. **The CEO of *Microlist* learns that Insertion Sort has a best case running time of O(*n*), while Heapsort, Mergesort, and Quicksort all have best case running times of O(*n*log*n*). As a result, the CEO declares that Insertion Sort will be the official sorting algorithm of the company, "since *Microlist* always likes to do best in the best possible situation."**

   (a) Explain to the CEO why this decision may be unwise, and suggest a better official sorting algorithm.

   (b) Annoyed by your failure to mindlessly accept corporate policy, the CEO says that s/he will accept your suggestion for the official sorting algorithm *if* you can make it run in O(*n*) time whenever it gets an input that would have been a best case for Insertion Sort. Your change must not asymptotically affect the running time for other inputs. Briefly describe a simple change you could make to your suggestion for part (a) that would meet the CEO's goal (*hint:* there is a simple solution that works regardless of your choice of algorithm in part (a)).

7. **I give my class a pop quiz of 20 true/false questions, each worth 5 points, each graded all-or-nothing. Briefly describe an efficient way for me to find the median grade in linear time (*hint*: we've learned *two* ways in the past week -- you only have to use one of them).**