CSci 231 Homework 3

Quicksort and Linear Time Sorting

CLRS Chapter 7 and 8

Write and justify your answers on this sheet in the space provided.[1]

1. (CLRS 7.2-2) What is the running time of Quicksort when all elements of arrary A have the same value?

2. (CLRS 7.2-3) Briefly sketch why the runnning time of Quicksort is $\Theta(n^2)$ when the arrary A contains distinct elements and is sorted in decreasing order.

[1] Collaboration is allowed, even encouraged, provided that the names of the collaborators are listed along with the solutions. Write up the solutions on your own.

3. (CLRS 7-3) Professors Dewey, Cheatham, and Howe have proposed the following "elegant" sorting algorithm:

```
STOOGE-SORT($A, i, j$)
if $A[i] > A[j]$
    then exchange $A[i] \leftrightarrow A[j]$
if $i + 1 \geq j$
    then return
$k \leftarrow \lfloor (j - i + 1)/3 \rfloor$
STOOGE-SORT($A, i, j - k$)
STOOGE-SORT($A, i + k, j$)
STOOGE-SORT($A, i, j - k$)
```

a. Argue that STOOGE-SORT($A, 1, length[A]$) correctly sorts the input array $A[1..n]$, where $n = length[A]$.

b. Give a recurrence for the worst-case running time of STOOGE-SORT and a tight asymptotic (Θ-notation) bound on the worst-case running time.

c. Compare the worst-case running time of STOOGE-SORT with that of insertion sort, merge sort, heapsort, sock sort, and quicksort. Do the professors deserve tenure?

4. (CLRS 8.3-2) Which of the following sorting algorithms are stable: insertion sort, merge sort, quicksort? Give a simple scheme that makes any sorting algorithm stable. How much additional time and space does your scheme entail?

5. (CLRS 8.3-4) Show how to sort n integers in the range 1 to n^2 in $O(n)$ time.

6. (CLRS 8.4-1) Illustrate the operation of BUCKET-SORT on the array

$$A = [.79, .13, .16, .64, .39, .20, .89, .53, .71, .42]$$

7. (CLRS 8-2 first part only) You are given an array of integers, where different integers may have different numbers of digits, but the total number of digits over *all* the integers in the array is n. (a) What is the worst-case running time of radix sort? Give an example of input that elicits this worst-case. (b) Show how to sort the array in $O(n)$ time.