CMPS 201
Spring 2010
Homework Assignment 6

1. (2 Points)
   Design an algorithm called Extrema(A, p, r) on the divide and conquer paradigm which finds and returns both the maximum and minimum values in $A[p \cdots r]$. Your algorithm should perform exactly $\lceil 3n/2 \rceil - 2$ comparisons, on an input array of length n.
   a. (1 Point) Prove the correctness of your algorithm.
   b. (1 Point) Write a recurrence for the number of comparisons performed on arrays of length n, and solve it exactly.

2. (1 Point) (This is problem 8.2-3 on page 170 of the 2nd edition.)
   Prove that Counting Sort is stable. Also show that if we reverse the order in which the final loop is executed, the resulting algorithm is correct, but not stable.

3. (1 Point) (This is a slight modification of problem 8.2-4, 2nd edition.)
   Describe an algorithm that, given n integers in the range 0 to k, preprocesses its input and then answers any query about how many of the n integers fall into the interval $(a,b]$, where a and b are integers satisfying $0 \le a \le b \le k$. Your algorithm should use $\Theta(n+k)$ preprocessing time, and queries should use $\Theta(1)$ time. (Hint: modify Counting Sort to do the preprocessing.)

4. (1 Point)
   Prove the correctness of RadixSort by (finite) induction on i, the column being sorted. Be sure to carefully state your induction hypothesis. It should be clear from your proof why the sort on line 2 must be stable, and why the algorithm must sort the digits from least to most significant. RadixSort() is provided here for reference.

   RadixSort(*A*, *d*) (Pre: *A*[1..*n*] consists of *d* digit numbers)
   1. for $i \leftarrow 1$ to d
   2. sort A on digit i using a stable sort