1. (25 points) Quick sort.
   
a. (5 points) Study the pseudocode of the Partition algorithm in lecture9.ppt. Using Slide #37 as a model, illustrate the operation of Partition on array \( A = [14 \ 16 \ 11 \ 12 \ 20 \ 4 \ 15 \ 3 \ 19] \). Importantly, indicate where is the pivot element when the algorithm terminates.

b. (5 points) Using Slide #38 as a model, illustrate the operation of Quick Sort on array \( A \) above.

c. (5 points) The Partition algorithm may have some problem when all elements are equal. Try to apply Partition on array \( D = [5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5 \ 5] \) and discuss the problem. What will be the running time of quick sort for this kind of array?

d. (5 points) In the pseudocode for Quicksort on Slide #31, Quicksort was recursively applied to two subarrays: one subarray from index \( p \) to \( q - 1 \), and the other from index \( q + 1 \) to \( r \). This is correct because element \( A[q] \) is already in the correct position. Now consider the following modified pseudocode for Quicksort:

   ```
   QUICKSORT(A, p, r)
   if (p < r)
       then q <- PARTITION(A, p, r)
       QUICKSORT(A, p, q-1)
       QUICKSORT(A, q, r)
   ```

   The only difference between the pseudocode above and the original one is the last line. In the above pseudocode, the second subarray starts from \( q \) instead of \( q + 1 \). Besides having to sort one extra element, this modified algorithm will sometimes fail to terminate. Give an example array for which the modified algorithm will fail to terminate.

e. (5 points) How many times was the procedure Partition called in the two examples above? (The answer can also help you solve the next problem.)

2. (20 points) Analysis of Randomized Quick Sort.
   During lecture, we analyzed Randomized Quick Sort and proved its expected running time using the substitution method. This homework problem concerns the number of times that the procedure Partition is called.

   a. (7 points) Using the analysis on Slide #57 as a template, write down a recurrence for the expected number of times that the procedure Partition is called in Randomized Quick Sort, for an input of size \( n \).

   b. (10 points) Using the substitution method to prove that the expected number of calls to Partition is \( O(n) \).

   c. (3 points) Provide an intuitive explanation why there is \( O(n) \) calls to Partition. (Hint: after each call to Partition, how many elements will be sorted into the right position?)
3. (10 points) Indicate whether the following statement is true or false. Briefly justify your answers.

   a. Quick sort runs in linear time for an already sorted array.

   b. Quick sort can run in $\Theta(n^2)$ time in the worst case.

   c. Randomized quick sort is guaranteed to run in $\Theta(n \log n)$ time.

   d. The average running time on all possible inputs is $\Theta(n \log n)$ for both quick sort and randomized quick sort.

4. (Extra credit: 15 points) Nuts and bolts. You are given a collection of $n$ bolts of different widths and $n$ corresponding nuts. You are allowed to try a nut and bolt together, from which you can determine whether the nut is larger than the bolt, smaller than the bolt, or matches the bolt exactly. However, there is no way to compare two nuts together or two bolts together. The problem is to match each bolt to its nut. Design an efficient algorithm for this problem with average-case efficiency in $\theta(n \log n)$. 