MA/CSSE 473 HW 7 Problem statements/hints

Problems 1 and 2 (copied here from the assignment document):

1. **(15)** (RSA decoding). If small primes are used, it is computationally easy to "crack" RSA codes.

   Suppose my public key is N=703, e= 53. You intercept an encrypted message intended for me, and the encrypted message is 361. What was the original message? How did you get the answer?

2. **(6)** (RSA attacks) Read about various ways of attacking the RSA cryptosystem.

   Write about two attacks that interest you. Explain how they work.

Problem 3: **(3)** 5.1.4 [4.1.4] (logarithm base in Master theorem)

4. We mentioned in Chapter 2, that logarithm bases are irrelevant in most contexts arising in the analysis of an algorithm's efficiency class. Is this true for both assertions of the Master Theorem that include logarithms?

Problem 4: **(6)** 5.1.5 [4.1.5] (simple application of Master theorem)

5. Find the order of growth for solutions of the following recurrences.

a. $T(n) = 4T(n/2) + n, \ T(1) = 1$

b. $T(n) = 4T(n/2) + n^2, \ T(1) = 1$

c. $T(n) = 4T(n/2) + n^3, \ T(1) = 1$

Author's hints for problems 3 and 4:

4. Look at the notations used in the theorem's statement.

5. Apply the Master Theorem.

3. **Problem 5:** **(6)** 5.2.2 [4.2.2] (Quicksort partition scan properties)

2. For the partitioning procedure outlined in Section 4.2:

a. Prove that if the scanning indices stop while pointing to the same element, i.e., $i = j$, the value they are pointing to must be equal to p.

b. Prove that when the scanning indices stop, j cannot point to an element more than one position to the left of the one pointed to by i.

c. Why is it worth stopping the scans after encountering an element equal to the pivot?

Author's hint for problem 5: Use the rules for stopping the scans.

Problem 6: **(10)**

Solve the average-case recurrence for quicksort. The recurrence is given on page 180 [133] of Levitin. Feel free to look up a solution, understand it, and write it in your own words (and symbols). The Weiss Data Structures book (Section 8.6.2) is one possible source. You should write a reasonable amount of detail.

Problem 7: **(6)**

Problem 7 5.2.8 [4.2.8] (Negatives before Positives)

8. Design an algorithm to rearrange elements of a given array of n real numbers so that all its negative elements precede all its positive elements. Your algorithm should be both time- and space-efficient.

Author's hint for problem 8: Use the partition idea.

Problem 8: **(8)** 5.2.9 [4.2.9] (Dutch National Flag)

9. ▶ The *Dutch national flag problem* is to rearrange an array of characters R, W, and B (red, white, and blue are the colors of the Dutch national flag) so that all the R's come first, the W's come next, and the B's come last. Design a linear in-place algorithm for this problem.

Author's Hints:

9. a. You may want to solve first the two-color flag problem, i.e., rearrange efficiently an array of R's and B's. (A similar problem is Problem 8 in this section's exercises.)

   b. Extend the definition of a partition.

Problem 9: **(8)** 5.2.11 [4.2.11] (nuts and Bolts)

11. ▶ *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 algorithm for this problem with average-case efficiency in $\Theta(n \log n)$. [Raw91]

Author's Hints:

Use the partition idea.