MA/CSSE 473 HW 7 Problem statements/hints

Problems 1 through 3 (copied here from the assignment document):

1. **(15)** (Miller-Rabin test) Let N = 1729.
   (a) How many values of **a** in the range 1..1728 pass the Fermat test [i.e. $a^{1728} \equiv 1 \pmod{1729}$]?
   (b) For how many of those values does the Miller-Rabin test provide a witness that N is composite?
   (c) For N=1729, if we pick **a** at random, what is the probability that running the test on **a** will show that N is composite? Note: Rabin showed that for any N, the probability is at least 75%, what is it for this case?

   [**Hint**: writing a few lines of code my help you here].

2. **(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?

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

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

Problem 4: **(3)** 5.1.4 [4.1.4]

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 5: **(6)** 5.1.5 [4.1.5]

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 4 and 5:

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

5. Apply the Master Theorem.

Problem 6: **(6)** 5.2.2 [4.2.2]

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 6: Use the rules for stopping the scans.

Problem 7: **(10)**

4. **7.** Solve the average-case recurrence for quicksort. 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 8: **(6)** 5.2.8 [4.2.8]

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.