MA/CSSE 473 – Design and Analysis of Algorithms

Homework 6A (54 points total) Updated for Winter, 2017

Problems for enlightenment/practice/review (not to turn in, but you should think about them):

How many of them you need to do serious work on depends on you and your background. I do not want to make everyone do one of them for the sake of the (possibly) few who need it. You can hopefully figure out which ones you need to do.

3.5.2 [5.2.2]	(adjacency matrix *vs* adjacency list for DFS)
3.5.7 [5.2.7]	(Use BFS/DFS to find a graph's connected components)
3.5.10 [5.2.10]	(DFS and mazes)
5.1.7 [4.1.7]	(Merge sort stability)
5.1.9 [4.1.9]	(O(n log n) algorithm to count inversions in an array)
5.2.1 [4.2.1]	(quicksort example)
5.2.4 [4.2.4]	(quicksort sentinel)
5.2.6 [4.2.6]	(increasing arrays in quicksort)

Problems to write up and turn in:

Problems 1-2 are related to Dasgupta pages 30-34 and Weiss section 7.4 (both on Moodle)

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 of RSA was the original message? How did you get your answer? [RSA details are found in Dasgupta, and in Weiss section 7.4.4]

2. (6) (RSA attacks) Find and read about various ways of attacking the RSA cryptosystem. Write about two attacks that interest you. Write a paragraph about each one to explain in your own words how it works.

Problems 3-7 relate to material that should be review from CSSE 230. In addition to the Levitin textbook, Weiss Chapter 8 should be good background for those.

3. (3) 5.1.4 [4.1.4] (logarithm base in the Master Theorem)

4. (6) 5.1.5 [4.1.5] (Simple application of the Master Theorem)

5. (6) 5.2.2 [4.2.2] (Quicksort partition scan properties) Note that the old (2nd) edition of the Levitin book has a part c, and I want you to do it, you can find it in the http://www.rose-hulman.edu/class/csse/csse473/201720/Homework/hw06A_levitin_probs.pdf document.

6. (10) Show how to 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 place that has a solution. You should write a reasonable amount of detail, enough to convince me that you understand it.

7. (8) 5.2.11 [4.2.11] (Nuts and bolts). In addition to writing the algorithm, write and solve a recurrence relation for average-case running time.