MA/CSSE 473 – Design and Analysis of Algorithms

Homework 11 (43 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.

7.2.2 [7.2.2] (Horspool for patterns in DNA)
7.2.5 [7.2.5] (is there a case where Horspool does more comparisons than brute force?)
7.2.9 [7.2.9] (left-to-right checking OK after a single character match in Horspool, Boyer-Moore?)
7.3.1 [7.3.1] (insert specific keys into hash table with specific hash function and separate chaining)
8.1.1 [8.1.1] (Compare and contrast dynamic programming with divide-and-conquer)
8.1.4 [8.1.9] (Space efficiency of dynamic programming for Binomial coefficients)

Problems to write up and turn in:

1. (6) 6.5.11 [6.5.10] (Factored form or not) 2 points for each part. Explain your answers.

2. (5) 6.6.4a [6.6.4a] (better than cubic check for length 3 cycle in graph)

3. (6) 7.2.3 [7.2.3] (Horspool for binary strings)

4. (9) 7.2.7 [7.2.7] (Boyer-Moore for binary strings)

5. (4) 7.2.8 [7.2.8] (does Boyer-Moore still work with just one table?)

6. (8) 7.2.11 [not in 2nd ed] (right cyclic shift) 3 points for part a, 5 for part b.

   You are given two strings S and T, each n characters long. You have to establish whether one of them is a right cyclic shift of the other. For example, PLEA is a right cyclic shift of LEAP, and vice versa. (Formally, T is a right cyclic shift of S if T can be obtained by concatenating the $(n - i)$-character suffix of S and the i-character prefix of s for some $1 \le i \le n$).
   a. Design a space-efficient algorithm for the task. Indicate the space and time efficiencies of your algorithm.
   b. Design a time-efficient algorithm for the task. Indicate the time and space efficiencies of your algorithm.

7. (5) 7.3.4 [7.3.4] (probability that n keys all hash to the same table location)