MA/CSSE 473 – Design and Analysis of Algorithms

Homework 11 (7 probs, 43 points total) Updated for Summer, 2017

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

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).
   Reminder: Our convention for this course is: If a problem does not explicitly say that a graph is directed, assume that it is undirected. That makes this problem much easier.

   Problem 2 previous questions and answers from Piazza:

   Q: How is the graph in this problem being represented? Should it matter whether we are using adjacency lists or an adjacency matrix in our algorithm, or should we state it more generally?
   A: Pick a graph representation that allows you to do this in less than cubic time. Indicate which representation you are using.
   Q: For this question, is cycle of length 3 the same as three nodes connect to each other?
   A: It is three nodes connected to each other in a cycle, a triangle (as opposed to connected as a star or straight line). Each of the three nodes is connected to the other two.

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.

   Problem 6 previous questions and answers from Piazza:

   Q: For this question, does right cyclic shift mean the case where the string rotate only once? For example, only PLEA is the right cycle shift for LEAP? Does APLE also count as a right cyclic shift for LEAP?
   A: APLE does count as a right cyclic shift. It could be shifted by 0, 1, 2, 3, ..., n-1 where n is the length of the text
   Q: When we are comparing strings with our algorithms, do we have to compare one character at a time? Or can we create strings and just compare one whole string to another?
   A: You have to compare a character at a time. No way to compare strings in constant timet
   Q: For part b, the hint says to enhance the input before the search. Does this mean we have to make something like a table or something similar to the algorithms like Horspool in the book? Or can we just precompute values and use those?

What is in the realm of "input enhancement"?
A: orting or any other massaging of the input could be an "enhancement". The idea is to do some work one time that makes each subsequent use of the data be faster.

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