CSI 5126 Algorithms in Bioinformatics — Fall 2011
Computer Science — University of Ottawa

Homework Assignment #2 (100 points, weight 15%)
Due: Friday October 21

Suffix trees and related data structures

1. (20 marks) Consider the text $T = acgcgcacgt\$$.

   (a) Show the suffix tree for T.

   (b) Show the suffix array for T.

   (c) Show $BW[1..11]$ and $C[a], C[c], C[g]$ and $C[t]$, as defined in the FM-index.

   (d) Demonstrate the steps for finding the pattern $P = cgc$ using backward search on the FM-index.

2. (20 marks) Given a set of k strings of total length n, give an $O(n)$-time algorithm which computes the longest common substring of all k sequences. Precisely detail what each step does. (You may use data structures studied in class as well as your knowledge of the running time needed to build them, without giving the explicit algorithm to build them.)

3. (20 marks) Consider the algorithm for finding all maximal palindromes of a string, given in the textbook section 3.3.5. In step 2, assume the tree was preprocessed to find $LCA(i, j)$ in constant time, and add an annotation to each internal node with its string depth. For string $S = acacabba\$$, show the annotated suffix tree generated after steps 1 and 2. Show all the calculations in steps 3 and 4 and the list of maximal palindromes as they are found in the order given by the algorithm (each palindrome is given by the indices of its beginning and end).

4. (20 marks) Show how to count the number of distinct substrings of a string $T[1..m]$ in $O(m)$-time. Show how to list one copy of each distinct string in time proportional to the length of all those strings. Hint: use suffix trees.

5. (20 marks) Study the main steps in Farach's algorithm for building a suffix tree in linear time (textbook pages 67-72). Given the string $bababbaa\$$, illustrate the main step of the algorithm, showing how the tree changes as given in Figure 3.8 (step 1), Figure 3.9 (step 2), Fig 3.10,3.12 (step 3).