CSI 5126 Algorithms in Bioinformatics   Fall 2010
Computer Science   University of Ottawa

Homework Assignment #2 (100 points, weight 15%)
Due: Wednesday October 27

Suffix trees and related data structures

1. (20 marks) Consider the text $T = acgtcga\$$.
   (a) Show the suffix tree for T.
   (b) Show the suffix array for T.
   (c) Show $BW[1..8]$ and $C[a], C[c], C[g]$ and $C[t]$, as defined in the FM-index.
   (d) Demonstrate the steps for finding the pattern $P = cg$ using backward search on the FM-index.

2. (20 marks) Given a string $S = S[1..n]$ and a number k, we want to find the smallest substring of S that occurs in S exactly k times, if it exists. Give an algorithm to solve this problem in $O(n)$ time. Describe all steps.
   (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 two DNA sequences S and T of total length n. Describe an $O(n)$ time algorithm which detects whether there exists a substring T' of T such that the score of the global alignment between S and T' is bigger than or equal to -1 (where the score for a match is 0 and the score for insert, delete or mismatch is -1).

   Hints: 1) Review the algorithm given in class for k-mismatch and note that one of the cases where the score can be -1 or 0 is if we can find a 1*-mismatch of S in T. 2) Think of the other possible cases for getting a score of -1, and try to use methods as efficient as the one for* 1*-mismatch to handle those cases.*

4. (20 marks) Given a set of k strings, we would like to compute the longest common substring of each of the $\binom{k}{2}$ pairs of strings.
   (a) Assume each string is of length n. Describe an algorithm to find all the longest common substrings in $O(k^2 n)$ time. Briefly justify the running time.
   (b) Assume the string lengths are different but sum to m. Describe an algorithm to find all the longest common substrings in $O(km)$ time. Briefly justify the running time.

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 $abaaabba\$$, illustrate the main steps 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).