CAP5510 Introduction to Bioinformatics

Fall 2009

Homework #2

(Assigned Sept.14. Due: Sept.21, 2009)

1. Write in high-level pseudo code algorithms for (a) and (b):
   (a) Write a non-recursive algorithm to obtain all the Fibonacci numbers of maximum size *n*. (5 points)
   (b) Given a pattern *P* of length *m* and a text *T* of length *n* over the alphabet Σ= (*A, C, G, T*) for both pattern and text, compute the beginning locations of the occurrences of the pattern *P* in *T*. (*P* may or may not occur in *T* or it might occur multiple times possibly in overlapping fashion.) (12 points)
   (c) Sketch an algorithm ("sketch" means you explain the basic idea of the algorithm and describe the steps in precise English language) to find the beginning locations of the occurrences of the pattern *P* in *T* over the alphabet Σ= (*A, C, G, T*) such that the Hamming distance between *P* and |*P*| consecutive characters in *T* is less than equal to 1. For example, if *P=AGA* and *T=ATAGAGCGATA*, locations in *T* satisfying the condition are 1, 3, 5, 7, 9. Analyze the computational complexity of your algorithm. (8 points)

   For both (a) and (b), If you prefer writing actual programs using a language of your choice, it will be fine.

2. Write a high level program to compute the minimum edit distance of two sequences over some finite alphabet. Assume a cost model for insert, delete, substitution and match as appropriate. What changes you need to make if the program is to compute the longest common subsequence of the two sequences. (20 points)

3. Given two DNA sequences S_1=A G T T C A G and S_2=G A G T A G T find all optimal global alignments with minimum edit distance by computing the dynamic programming table, row by row and then trace back the alignments to exhibit them in pairs of rows. The cost model is : insertion and deletion have cost of 1 unit, substitution has a cost of 2 units and match has cost 0. (25 points)

4. Given two amino acid sequences **v =** P A W H E A E and **w =** H E A G A W G H E E. Using dynamic programming find an optimal alignment. To score the alignment score, use BLOSUM50 score matrix and a gap cost per unaligned residue of -8. (30 points)