Algorithms and Applications (15-499)
Assignment #3 Due Date: 20 Mar 03

Problem 1: 10 Given two strings S_1 and S_2 and a text T, you want to find whether there is an occurrence of S_1 and S_2 interwoven in T, possibly with spaces. For example, the strings `abac` and `bbc` occur interwoven in `cabcbabcca`. Give an efficient algorithm for this problem (i.e. one that is polynomial in the size of the inputs).

Problem 2: 10 Consider the following gap model – each insertion or deletion costs a unit. However, if there are more than k consecutive insertions, or k consecutive deletions, they cost only k units. Give an algorithm that finds the minimum edit distance under this cost model in time $O(nm)$. Note that the time should not depend on k. (Do not worry about space efficiency).

Problem 3: 10 The *shortest superstring problem* is the problem of finding the shortest string that contains all given strings as its substrings. Formally, given a set of m strings $S_1, \cdots, S_m$, find the shortest string T such that each S_i is a substring of T. The motivation for this problem is given in the Chapter on Computational Gene Hunting I handed out (Section 1.4).
Reduce this problem to a Traveling Salesman Problem. The reduction needs to take polynomial time (in $n = \sum_{i=1}^{m} |S_i|$). For extra credit prove that the shortest superstring problem is NP-hard.

Problem 4: 10 The main heuristic behind FAST and BLAST is that of finding small exact matches and using them to compose larger approximate matches. This heuristic is partially justified by the following lemma.

Lemma 1 *Suppose that $a_1, \ldots, a_t$ and $b_1, \ldots, b_t$ can be aligned with at most l mismatches (i.e., $a'_i \neq b'_i$), then for $k \leq \lfloor \frac{t}{l+1} \rfloor$, $a_1, \ldots, a_t$ and $b_1, \ldots, b_t$ share at least $t - (l+1)k + 1$ k-tuples (possibly overlapping).*

Prove the lemma, and show how it can be used to rapidly search all sequences in a database which can be aligned with a given query.

Problem 5: 10

A. Give the best parsimony score for the following five DNA sequences: ACA, GGC, ACG, GGA, and GCA.

B. Give a small example of a distance matrix such that the optimal tree by the least square measure is different from the tree found by the clustering method (Unweighted Pair Group Method with Arithmetic mean). Show the trees generated by both methods. The least square measure you should use is $\sum_i \sum_{j \neq i} (D_{ij} - d_{ij})^2$.

Problem 6: 10 Solve either one of the following two questions. Solving both is extra credit.

A. Describe how Hirshberg's linear space algorithm can be used for local alignment if all you care about is the one alignment with maximum score.

B. Given sequences S_1 and S_2 of length n, let the alignment cost between these sequences be given by $D(S_1, S_2) = \sum_{i \leq n} \delta(S_1[i], S_2[i])$, where δ satisfies the triangle inequality. Prove that D also satisfies the triangle inequality.

Given a set of m strings of length n, $S_1, \cdots, S_m$, the *consensus* sequence is a sequence T that represents the common features of all the given strings. In other words, it is the sequence minimizing the cost $E(T) = \sum_{i \leq m} D(T, S_i)$. Using the fact that D is a metric, give a simple 2-approximation to the consensus sequence. In other words, give an algorithm that produces a sequence C, such that $E(C) \leq 2E(T)$.