Algorithms in the Real World (15-853), Fall 05
Assignment #4

Due December 5. Do all 5 problems.

Problem 1: 10pt
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: 10pt
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: 10pt

(A) Find the best multiple alignment of the following three sequences, under the all-pairs min-edit-distance measure. That is, the cost of an alignment is given by the sum of the edit distance over all pairs of sequences. Assume that the edit distance between two gaps is 0. Also report the alignment cost.

$$\begin{array}{c} C\,A\,G\,A\,G \\ A\,T\,A\,G\,A \\ C\,T\,G\,A\,G \end{array}$$

(B) Along the lines of the Myers & Ukkonen algorithm for min-edit-distance, the all-pairs min-edit-distance problem for 3 sequences of length n each can be solved by constructing a three-dimensional graph, where each dimension corresponds to one of the sequences. As mentioned in class if the all-pairs min-edit-distance d is small and we run Dijkstra's shortest path algorithm on the graph, we will only explore a small part of the graph. Bound the number of nodes visited (in order notation) in terms of n and d.

Problem 4: 10pt
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.
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 5: 10pt

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.