CS 341, Fall 1999 due Friday Nov. 5

A. Lubiw, D. Schuurmans

ASSIGNMENT 3

Do not copy from others, and acknowledge your sources.

1. [10 points] Given 3 strings, X, Y, Z, we say that Z is a *shuffle* of X and Y if Z can be formed by interspersing the characters from X and Y, maintaining the left-to-right ordering of the characters from each string. For example 'cchocohilaptes' is a shuffle of 'chocolate' and 'chips', but 'chocochilatspe' is not. Devise a dynamic programming algorithm to decide whether Z is a shuffle of X and Y. Analyze your algorithm. Hint: The values in the table you construct should be Boolean.

2. [20 points] Suppose you need to store n books on shelves in a library. The order of the books is fixed by the cataloging system. Let book b_i, $i = 1, \ldots, n$ have thickness t_i and height h_i. The length of each bookshelf is L. In each of the following two cases give an efficient algorithm to place the books on the shelves. Justify correctness, and do a worst case analysis.
   (A) Each shelf has a fixed height, tall enough to fit any book. In this case you should place the books to minimize the number of shelves needed.
   (B) The heights of the shelves can be adjusted. In this case you should minimize the sum of the heights of the shelves used.

3. [10 points] Give a linear time algorithm to find a deletion order for the vertices of an undirected graph that leaves the graph connected at each step. Hint: use an algorithm we've covered.

4. PROGRAMMING. [30 points] (Problem suggested by Jeff Shallit.) This question is about designing a coin system with four coins. Suppose you want to be able to make change for any amount between $1¢$ and $99¢$. Your coin system must have the property that the greedy method of making change always gives the minimum number of coins. Furthermore, of all such coin systems, you want one that minimizes the average number of coins needed to make change for any amount from $1¢$ to $99¢$. Write a program to do this. You are required to implement the dynamic programming algorithm for making change. Write your algorithm so that it handles any number of coins k and any amount of change n. You need only run it for $k = 4$ and $n = 99$. Analyze the worst case running time in terms of k and n.

5. BONUS QUESTION. [Only for grad. students and others with too much time on their hands.] A set of distinct points $p_0, \ldots, p_n$ in the plane defines a *polygon* if no two of the line segments (p_i, p_{i+1}) and (p_j, p_{j+1}) (addition modulo $n+1$) intersect except at a common endpoint. A polygon has a natural inside and outside. A *chord* of a polygon is a line segment (p_i, p_j) that stays inside the polygon. A *triangulation* of a polygon is a set of chords, no two intersecting except at end-points, that divide the polygon into triangles. Give an efficient algorithm to find a *minimum ink* triangulation of a polygon—i.e. a triangulation that minimizes the sum of the Euclidean lengths of the chords used. Justify.