CSE 441T/541T: Advanced Algorithms	Fall Semester, 2004

Practice Problems on Dynamic Programming

September 21, 2004

Below are four practice problems on designing and proving the correctness of dynamic programming algorithms. For those of you who feel like you need us to guide you through some additional problems (that you first try to solve on your own), these problems will serve that purpose.

The solutions are available on the course web page (under homeworks). You can use these solutions as a guide as to how you should write-up your solutions. These problems will be MUCH more valuable to you if you first solve them and then check the solutions. Hints to get you going are also available on the web page if you want a hint without seeing the whole solution. If you need further guidance, let us know.

Practice Problems

1. Suppose we want to make change for n cents, using the least number of coins of denominations $1, 10$, and 25 cents.

   Describe an $O(n)$ dynamic programming algorithm to find an optimal solution. (There is also an easy $O(1)$ algorithm but the idea here is to illustrate dynamic programming.)

2. Here we look at a problem from computational biology. You can think of a DNA sequence as sequence of the characters "a","c","g","t". Suppose you are given DNA sequences D_1 of n_1 characters and DNA sequence D_2 of n_2 characters. You might want to know if these sequences appear to be from the same object. However, in obtaining the sequences, laboratory errors could cause reversed, repeated or missing characters. This leads to the following sequence alignment problem.

   An alignment is defined by inserting any number of spaces in D_1 and D_2 so that the resulting strings D_1' and D_2' both have the same length (with the spaces included as part of the sequence). Each character of D_1' (including each space as a character) has a corresponding character (matching or non-matching) in the same position in D_2'. For a particular alignment A we say $cost(A)$ is the number of mismatches (where you can think of a space as just another character and hence a space matches a space but does not match one of the other 4 characters).

   To be sure this problem is clear suppose that D_1 is `ctatg` and D_2 is `ttaagc`. One possible alignment is given by:

   ```
   ct at g
    tta agc
   ```

   In the above both D_1' and D_2' have length 8. The cost is 5. (There are mismatches in position 1, 3, 5, 6 and 8).

   Give the most efficient algorithm you can (analyzed as a function of n_1 and n_2) to compute the alignment of minimum cost.

3. You are traveling by a canoe down a river and there are n trading posts along the way. Before starting your journey, you are given for each $1 \leq i < j \leq n$, the fee $f_{i,j}$ for renting a canoe from post i to post j. These fees are arbitrary. For example it is possible that $f_{1,3} = 10$ and $f_{1,4} = 5$. You begin at trading post 1 and must end at trading post n (using rented canoes). Your goal is to minimize the rental cost. Give the most efficient algorithm you can for this problem. Be sure to prove that your algorithm yields an optimal solution and analyze the time complexity.

4. For bit strings $X = x_1 \ldots x_m$, $Y = y_1 \ldots y_n$ and $Z = z_1 \ldots z_{m+n}$, we say that Z is an *interleaving* of X and Y if it can be obtained by interleaving the bits in X and Y in a way that maintains the left-to-right order of the bits in X and Y. For example if $X = 101$ and $Y = 01$ then $x_1 x_2 y_1 x_3 y_2 = 10011$ is an interleaving of X and Y, whereas 11010 is not. Give the most efficient algorithm you can to determine if Z is an interleaving of X and Y. Prove your algorithm is correct and analyze its time complexity as a function $m = |X|$ and $n = |Y|$.