CS 3343 Analysis of Algorithms – Fall 2007

10/23/07

7. Homework

Due **10/30/07** before class

1. **LCS traceback**
   Give pseudocode that performs the traceback to construct an LCS from a filled dynamic programming table *without* using the "arrows", in $O(n+m)$ time.

2. **Knapsack traceback**
   Give pseudocode that performs the traceback to construct an optimal solution for the Knapsack problem. (Similar to the LCS problem, you may either use "arrows" or recompute the local solutions while you trace back. If you use arrows you need to tell us how and when you compute the arrows.)

3. **Saving space**
   (a) The bottom-up dynamic programming algorithm computing the n-th Fibonacci number $F(n)$ takes $O(n)$ time and uses $O(n)$ space. Show how to modify the algorithm to use only constant space.
   (b) Suppose we only want to compute the *length* of an LCS of two strings of length m and n. Show how to alter the dynamic programming algorithm such that it only needs $O(\min(m,n))$ space.

4. **Binomial coefficient**
   In class we discussed a bottom-up dynamic programming algorithm which computes the binomial coefficient $\binom{n}{k}$, for given $n \geq k \geq 0$, in $O(nm)$ time using the recurrence below:

$$\begin{aligned}\binom{n}{k} &= \binom{n-1}{k-1} + \binom{n-1}{k}, \text{ for } n > k > 0\\ \binom{n}{0} &= \binom{n}{n} = 1, \text{ for } n \geq 0\end{aligned}$$

   Now assume you use memoization to compute $\binom{4}{3}$ using the above recurrence. In which order do you fill the entries in the DP-table? Give the DP-table for this case and annotate each cell with a "time stamp" when it was filled.