CS 3343 Analysis of Algorithms – Spring 09

3/24/09

7. Homework

Due: Tuesday 3/31/09 before class

Justify all your answers.

1. **LCS traceback (3 points)**
   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. **Saving space (5 points)**
   (a) (2 points) 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) (3 points) 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.

3. **Binomial coefficient (4 points)**
   Given n and k with $n \geq k \geq 0$, we want to compute the binomial coefficient $\binom{n}{k}$. However, we are only allowed to use additions, and no multiplications.

   a) (3 points) Give a bottom-up dynamic programming algorithm to compute $\binom{n}{k}$ using the recurrence

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

   b) (1 point) What are the runtime and the space complexity of your algorithm, expressed in n *and* k?

4. **Knapsack DP (6 points + 3 extra credit points)**
   Design a dynamic programming algorithm for the 0-1-Knapsack problem.

   - **(3 points)** Let $D[i,w]$ be the total value of a solution considering items $1..i$ only and with maximum weight w. Come up with a recurrence relation for $D[i,w]$. *(Hint: Distinguish two cases depending on whether $w_i \leq w$ (i.e., item i fits into the knapsack) or $w_i > w$ (i.e., item i does not fit into the knapsack))*
   - **(3 points)** Use dynamic programming to compute $D[n,W]$. Give pseudo code for your algorithm. What is the runtime of your algorithm in terms of n and W?
   - **(3 extra credit points)** Extract the optimum set of items from the dynamic programming table. What is the runtime in terms of n and W?