CS 3343 Analysis of Algorithms – Fall 10

10/26/10

8. Homework

Due: Tuesday 11/2/10 before class

Justify all your answers.

1. **LCS traceback (4 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. **LCS in less space (4 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 (7 points)**
   Given n and k with $n \geq k \geq 0$, we want to compute the binomial coefficient $\binom{n}{k}$.

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

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

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

   (c) (3 points) 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" (i.e., with a number $1, 2, 3, ...$) when it was filled.

4. **Subsets of integers (8 points)**
   Consider the following problem:

   > Given a positive integer S and an array $A[1..n]$ of n positive integers.
   > Is there a subset of integers in A that sum up to exactly S?

   (a) (2 points) Give a brute-force algorithm for this problem that runs in exponential time in n.

   (b) (3 points) Let $T[i, s]$ be true if there is a non-empty subset of integers in $A[1..i]$ which sum to s, and false otherwise. Develop a recurrence relation for $T[i, s]$. You do not have to prove the correctness, but please justify your answer shortly.

   (c) (3 points) Use dynamic programming to solve the above problem using the recurrence that you developed. What is the runtime of your algorithm in terms of n and S?