CMPS 2200 Introduction to Algorithms – Fall 14

9/30/14

5. Homework

Due **10/14/14** at the beginning of class

Remember, you are allowed to turn in homeworks in groups of two. One writeup, with two names.

1. **LCS pseudo-code (3 points)**
   Give pseudo-code for computing the length of an LCS of two strings of length m and n by filling a dynamic programming programming table.

2. **DP in less space (5 points)**
   (a) (1 point) 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. Give pseudo-code for your solution.
   (b) (4 points) Suppose we want to compute only the *length* of an LCS of two strings of length m and n. Describe how to alter the dynamic programming algorithm such that it only needs $O(\min(m, n))$ space. Give pseudo-code for your solution.
   (Hint: Try to first develop an algorithm that runs in either $O(m)$ or $O(n)$ space, and then figure out how to cut the space down to $O(\min(m, n))$.)

3. **LCS traceback (8 points)**
   (a) (4 points) Give pseudocode that performs the traceback to construct an LCS from a filled dynamic programming table *with* using the "arrows", in $O(n + m)$ time.
   (Hint: For an elegant solution you could use recursion to use the recursion stack to reverse the output sequence on the fly.)
   (b) (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. Justify shortly why your algorithm is correct.
   (Hint: You need to essentially "recompute" the information.)

FLIP OVER TO BACK PAGE ⟹

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 have developed. What is the runtime of your algorithm in terms of n and S?