Australian National University
Department of Computer Science

COMP3600/COMP6466 in 2007 – Tutorial Two

Question 1.
Give an $O(n^2)$-time dynamic programming algorithm to find a longest decreasing subsequence of a sequence of n numbers.

Question 2.
Show how to reconstruct an LCS from the completed c table and the original sequence $X = \langle x_1, x_2, \ldots, x_m \rangle$ and $Y = \langle y_1, y_2, \ldots, y_n \rangle$ in $O(m + n)$ time, without using the b table.

Question 3.
Consider the problem of making change for n cents using the least number of coins. Does the obvious greedy algorithm work for coin denominations 100, 50, 20, 10, 5? What if the coin denominations were 17, 10, 4, 2, 1? Give an algorithm that always works.

Question 4.
Suppose you are given two sets A and B, each containing n positive integers. You can choose to reorder each set however you like. After reordering, let a_i be the ith element of set A, and let b_i be the ith element of set B. You then receive a payoff of $\prod_{i=1}^{n} a_i^{b_i}$. Given an algorithm that will *maximize* your payoff. Prove that your algorithm maximizes the payoff and state its running time.

Question 5.
Explain how to implement a CHANGE-KEY procedure in a max-heap. That is, given an element in the heap and a new key value, change the key of the element to the new value. The new key value might be either larger or smaller than the old key value.