CS 332 Homework Assignment 5

Fall, 2002

This assignment is not a programming assignment. You can either write up your solutions to these problems using pen and paper or you can produce an electronic document using programs like MS Word, LaTeX, or Maple (for example, this document was created using Maple). Whatever you turn in should be neat, readable, and well organized. This assignment is due on Friday, November 8.

Problem 1. Suppose that you are given a sorted array of *distinct* integers $\{ a_1, a_2, ..., a_n \}$. Give an $O(\ln n)$ algorithm to determine whether there exists an index i such that $a_i = i$. For example, in $\{-10, -3, 3, 5, 7\}$, $a_3 = 3$. In $\{2, 3, 4, 5, 6, 7\}$ there is no such index i. (Note: This is Exercise 3-12 (page 80) from the course textbook.)

Problem 2. Suppose that S and T are sorted arrays, each containing n distinct elements. Give either an $O((\ln n)^2)$ or an $O(\ln n)$ algorithm to find the nth smallest of the $2n$ elements.

Problem 3. In the "art gallery guarding" problem we are given a line segment L that represents a hallway in an art gallery. We are also given a set $X = \{ x_1, x_2, ..., x_n \}$ of real numbers that specify the positions of paintings in this hallway. Suppose that a single guard can protect all the paintings within distance at most 1 of his or her position (on both sides). Design an algorithm for finding a placement of guards that uses the minimum number of guards to guard all of the paintings with positions in X.

Problem 4. A "maximum independent subset" of an array is a subset of the values whose sum is as large as possible but for which no two elements of the subset are contiguous in the array.

- (a) Give a divide and conquer algorithm to find the sum of a maximum independent subset of an array of integers.
- (b) Give an $O(n)$ dynamic programming algorithm to find the sum of a maximum independent subset of an array of integers. Be sure to give both the recurrence that defines your dynamic programming algorithm (with the proper basis cases) and a pseudocode implementation using "bottom-up" dynamic programming.
- (c) Give a reasonable "greedy" heuristic for this problem. Show that the heuristic need not produce an optimal solution.

Problem 5. Given an array of n integers, consider the problem of finding the maximum sum in any contiguous subvector of the array. For example, in the array $\{31, -41, 59, 26, -53, 58, 97, -93, -23, 84\}$ the maximum is achieved by summing the third through seventh elements where $59 + 26 - 53 + 58 + 97 = 187$. When all numbers are positive, the entire array is the answer, while when all numbers are negative, the empty array maximizes the total at 0.

- (a) Give a simple $O(n^2)$ algorithm to find the sum of a maximum contiguous subvector.
- (b) Give an $O(n \ln n)$ divide and conquer algorithm to find the sum of a maximum contiguous subvector.
- (c) Give an $O(n)$ dynamic programming algorithm to find the sum of a maximum contiguous subvector.