Theoretical Computer Science Qualifying Exam

Spring 1998

Please do 4 problems from the 6 given below.

1. For this problem you are to consider a two-player game played on an array A[1..n]. There are two players, Hide and Seek, who beforehand agree on an integer x in {1,..,n}. Hide moves first, and puts 1,2,...,n into the array (in any order, but he must use each number once). Seek then looks for x in the array, by examining its elements in any order she chooses, and must pay Hide 1 dollar for each examination of an element. We assume that Hide knows the search method used by Seek.
   a) Show that if Seek's strategy is deterministic, Hide can force her to pay n dollars. [Hint: use an adversary argument.]
   b) Give a randomized strategy that Seek can use that has average cost $n/2 + O(1)$, regardless of what Hide does. (Hint: one coin flip will suffice.)

2. a) What is the smallest integer n for which a finite-state device, equipped with n **stacks,** can recognize any recursively enumerable set?
   b) What is the smallest integer n for which a finite-state device, equipped with n **queues,** can recognize any recursively enumerable set?

3. The graph N-coloring problem, a variation of the classical graph coloring problem, is defined as follows. The input is a graph G and an integer $N \geq 2$; the problem is to color the nodes of G with N colors, so that the number of conflicting edges are minimized. A conflicting edge is one with both endpoints colored by the same color.
   a) A language formulation of the problem is as follows: given a graph G, and integers N and k, can the nodes of G be colored with N colors so that the number of conflicting edges is at most k? Prove that this problem is NP-complete.
   b) Consider the following "local search" algorithm for the problem.
   Initially, color the nodes arbitrarily.
   Repeat the following until no longer possible: [Find a node v and a color c such that by changing the color of v to c, the number of conflicting edges can be reduced. Color v with color c.]
   Output the resulting coloring.

   Show that this algorithm can be implemented in polynomial time, and that on all inputs, the output will have at most $|E|/N$ conflicting edges.

4. A roman numeral is a string over the alphabet $\{I, V, X, L, C, D, M\}$. Given a number n the Roman numeral for n can be constructed from left to right as follows:
   Repeatedly
   choose the symbol or pair of symbols from

   $$\{I, V, X, L, C, D, M, IV, IX, XL, XC, CD, CM\}$$

   which has the largest value $\leq$ n; add that to the string (i.e., the Roman numeral) and subtract that value from n.
   until n = 0.
   The values are defined as follows:

   $$I = 1, V = 5, X = 10, L = 50, C = 100, D = 500, M = 1000,$$

   $$IV = 4, IX = 9, XL = 40, XC = 90, CD = 400, CM = 900.$$

   For example the number 4999 is written MMMMCMXCIX while the number 1678 is written MDCLXXVIII.

   Is the set of Roman numerals regular?

5. Let x_1, x_2, ..., x_n be a set of positive integers, and let $S = \sum_{i=1}^{n} x_i$. Design an algorithm to partition the set into disjoint subsets of equal sum, or to determine that this is not possible. The algorithm should run in time $O(nS)$.

6. Let A[1..n] be an array of n distinct numbers. If i < j and A[i] > A[j], then the pair (i,j) is called an inversion.
   a) What array with elements from the set {1, 2, ..., n} has the most inversions?
   b) What is the relationship between the running time of insertion sort and the number of inversions in the input array?
   c) Prove the following: all sorting algorithms that do a single interchange of adjacent elements for each comparison make $\Omega(n^2)$ comparisons in the worst case.