CS 600 Final Exam

Spring 2002

1. Let L be an array of size n, let $L[i]$ denote the i-th key of L and let x be the key being searched for in L. Suppose that the probability for $x = L[i]$ is ci, where c is a constant.

(a) Formulate the equation for computing the average time of the Binary Search with the probability distribution given above in terms of c and n. Do not compute the average yet. [5]
(b) Compute the average time of (a). You do not have to compute the value of c yet. [15]
(c) Determine the value of constant c in terms of n and express the average time in terms of n only. What is the asymptotic average time ? [5]

You may use the following formulas if you need them:

$\Sigma_{i=1}^{n} i2^i = (n-1)2^{n+1} + 2,$ $\quad\quad$ $\Sigma_{i=1}^{n} i^2 = n(n+1)(2n+1)/6,$
$\Sigma_{i=1}^{n} i^3 = (n(n+1)/2)^2.$

2. Consider solving the following problem by Branch-and-Bound:

Maximize $\quad 20x_1 + 22x_2 + 10x_3 + 16x_4$
Subject to:

$$19x_1 + 20x_2 + 10x_3 + 15x_4 \leq 55$$

All x_i's are nonnegative integers.

(a) Give your bound and branching strategy. If your answer is different from widely accepted ones, justify your selection. [7]
(b) Solve the problem using the bound and strategy of (a).[18]

3. Answer whether or not the following statements are true. You DO NOT need to give your reasons. [25]

(a) If all the problems in NP are reduced to a problem (not necessarily in NP), then that problem is NP-complete.
(b) You can solve any NP-complete problem in $O(a^{p(n)})$ time in the worst case for some constant a and some polynomial $p(n)$, where n is the size of the problem.
(c) $2^n = O(3^n)$
(d) A sequence of vertices is a certificate for the graph color problem.
(e) Separable convex problems can be solved as linear programming problems because the rate of growth of their objective function is non-increasing (in addition to the linearity of objective function and constraints).
(f) If the auxiliary problem for finding an initial basic feasible solution has a nonnegative solution, then the linear programming problem has a feasible solution.
(g) The number of paths from a source to a destination that are disjoint from each other, that is, do not share edges between them, can be found by a maximum flow algorithm in polynomial time (even in the worst case).
(h) If a problem is in NP, then it can be solved by an algorithm with back trackings of depth polynomial in the size of the problem.
(i) A problem is NP-complete if a certificate of polynomial length exists which can be verified by a polynomial time algorithm.
(j) The multicommodity max flow problem can be solved by a modified Ford-Fulkerson algorithm in polynomial time.

4. People from districts X, Y and Z need to be assigned to committees A, B and C. Committee A consists of from 3 to 5 members, B from 4 to 7 and C from 5 to 7. Each district must send at least one person to each committee. But X can send total of at most 4 people, Y at most 7 and Z at most 5 to the committees.
Find an assignment with the smallest total number of people.

(a) Explain how you would solve this problem for the general case. You may refer to known algorithms. **DO NOT** give descriptions of those known algorithms. But if you devise your own algorithm, **DO EXPLAIN** it in detail. [5]
(b) Find an assignment with the smallest total number of people using the method of (a). [20]