COMP 271 Design and Analysis of Algorithms
2003 Spring Semester
Written Assignment # 4
Distributed: April 22, 2003
Due: May 6, 2003 at 5:00PM

Note: Please follow the guidelines on doing your own work and avoiding plagiarism given on the class home page.

2nd Note: Problem 1 is taken from the book *Problems on Algorithms,* by Ian Parberry, Prentice-Hall, 1995.

Problem 1. (25 pts)

(a) Recall that in each step of Huffman's algorithm, we merge two trees with the lowest frequencies. Show that the frequencies of the two trees merged in the ith step are at least as large as the frequencies of the trees merged in any previous step.

(b) Suppose that you are given the n input characters, already sorted according to their frequencies. Show how you can now construct the Huffman code in $O(n)$ time. (*Hint:* You need to make clever use of the property given in part (a). Instead of using a priority queue, you will find it advantageous to use a simpler data structure.)

Problem 2. (25 pts) (CLRS 16-2 a) Suppose you are given a set $S = \{a_1. \ldots, a_n\}$ of tasks where task a_i requires p_i units of processing time to complete, once it has started. You have one computer on which to run the tasks, and the computer can only run one task at a time. Let c_i be the *completion time* of task a_i, that is, the time at which task a_i completes processing. Your goal is to minimize the average completion time, that is, to minimize $(1/n)\sum_{i=1}^{n} c_1$. For example, suppose that there are two tasks, a_1 and a_2, with $p_1 = 3$ and $p_2 = 5$, and consider the schedule in which a_2 runs first, followed by a_1. Then $c_2 = 5$, $c_1 = 8$ and the average completion time is $(5 + 8)/2 = 6.5$.

Give an algorithm that schedules the tasks so as to minimize the average task completion time. You may assume that each task must run nonpreemptively, that is, once task a_i starts, it must run continuously for p_i units of time. Prove that your algorithm minimizes the average completion time and state the running time of your algorithm.

Problem 3. (15 pts) (CLRS 34.2-1) Consider the language

$$\text{GRAPH ISOMORPHISM} = \{< G_1, G_2 >\} : G_1 \text{ and } G_2 \text{ are isomorphic graphs}\}$$

(you can review the definition of isomorphic graphs on page 1082 of CLRS). Prove that GRAPH ISOMORPHISM $\in \mathcal{NP}$ by describing a certificate for the problem and a polynomial-time algorithm to verify the certificate.

Problem 4. (20 pts) (CLRS 34.2-7)) Show that the Hamiltonian-Path problem can be solved in polynomial-time on Directed-Acyclic-Graphs. Give an efficient algorithm for the problem; that is, your algorithm either gives a Hamiltonian path or it reports that there is no Hamiltonian path. Prove the correctness of your algorithm and analyze its running time.

Problem 5. (15 pts) (CLRS 34.2-8)) Let ϕ be a boolean formula constructed from the boolean input variables $x_1, x_2, \ldots, x_k$, negations ($\neg$), ORss ($\vee$), ANDs ($\wedge$), and parentheses. The formula ϕ is a *tautology* if it evaluates to 1 for *every* assignment of 0 and 1 to the input variables. Define TAUTOLOGY as the language of Boolean formulas that are tautologies. Show that TAUTOLOGY $\in$ Co-$\mathcal{NP}$. (Note. It is not necessarily true that TAUTOLOGY $\in \mathcal{NP}$).

Extra Credit Prove that $2-SAT$ is in $\mathcal{P}$. Give as efficient an algorithm as possible for either finding a satisfying assignment for a $2-SAT$ formula or showing that it is not satisfiable. Prove the correctness of your algorithm and analyze its running time.