PhD Qualifier Exam for Spring 2009— Theory Area

CS Department, IIT

Your number: ______________________________

There are 4 questions in this 2.5-hour exam. For every question, please write your answer in a clean and concise way.

If you are asked to write an algorithm for a question, you have to write the **pseudo-code** of your algorithm and also put explanations about your pseudo-code. Also show correctness and estimate the running time. Every statement must be proven (or be obvious to the grader).

Use separate pages for each problem or subproblem, writing on only one side of the paper.

1. Prove the following: if L_1 and L_2 are context-free languages, then $L_1 \circ L_2$ (the concatenation of the languages) is a context-free language.

2. Given an undirected graph $G = (V, E)$ with an real-valued function $f : V \to \Re$, a *local optimum* of f on G is a vertex $v \in V$ such that $f(v) \le f(w)$ for all edges $(v, w) \in E$. f is very expensive to evaluate at a vertex, so such evaluations dominate the cost of algorithms on such graphs.

   (a) Suppose $G = (V, E)$ is an undirected simple path; that is, $V = \{v_1, \ldots, v_n\}$ and $E = \{(v_i, v_{i+1}) | 1 \le i < n\}$. Give a divide-and-conquer algorithm to find a local optimum in worst-case $O(\log n)$ evaluations of f. Use recurrence relations to analyze the worst-case number of evaluations used by your algorithm.

   (b) Suppose $G = (V, E)$ is an undirected binary tree T with root $r \in V$ and, for all $v \in V$, edges $(v, \text{LEFT}(v))$, $(v, \text{RIGHT}(v))$ ($\text{LEFT}(v)$ and/or $\text{RIGHT}(v)$ can be NIL, in which case there is no such edge. Show how to find a local optimum in worst-case $O(h(T))$ evaluations of f, where $h(T)$ is the height of T.

3. Let $G = (V, E)$ be a simple undirected graph. The *inductivity* of an ordering $\langle v_1, v_2, \cdots, v_n \rangle$ of V, which is defined by

   $$\max_{2 \le j \le n} |\{1 \le i < j : v_i v_j \in E\}| .$$

   Give a polynomial algorithm (pseudocode) to produce a least-inductivity vertex ordering of G, together with the proof of correctness and an upper bound on the running time.

   Hint Use a greedy strategy paying attention to nodes with smallest degree in G.

4. Assume that you *only* know that the following problems are NP-complete: SAT, 3SAT, VERTEX-COVER, CLIQUE, HAMPATH, SUBSET-SUM (for the definition of the problems, look at the included notes from Sipser)

   Consider the following problem, called SHORTEST SIMPLE $s - t$ PATH: Given a directed graph $G = (V, E, w)$, where $w(e)$ is defined as a (possibly negative) integer for each edge $e \in E$, vertices $s, t \in V$, and a *positive* integer K, answer YES if there is a simple $s - t$ path of total weight at most K. An $s - t$ path starts at s and ends at t, a path is *simple* if it does not repeat any vertices, and the total weight of a path is the sum of the weights of its edges.

   Prove that the SHORTEST SIMPLE $s - t$ PATH problem is NP-complete.