Examination "Breadth" – Theory

February 11, 2005

SOLVE EACH PROBLEM ON A SEPARATE PAGE. WRITE ONLY ON ONE SIDE OF A PAGE. WRITE YOUR CODE AT THE TOP OF EACH PAGE.

IF YOU HAVE PROBLEMS WITH THE MEANING OF THE QUESTIONS, ASK THE PROCTOR.

1. What are the deterministic time and space complexities of regular languages? In other words, if you are given a regular language description and asked to build a Turing machine to accept that regular language, what can you say about the time the Turing machine would need to accept a string of length n? How much additional memory (besides the read-only input tape) would the Turing machine need?

   Sketch proofs of your answers.

2. Let $k \geq 2$ be an integer and let L be a language that is accepted by a k-tape Turing machine M. Prove that there is a one-tape Turing machine that accepts L.

3. (a) Give the definition of a Binary Search Tree for a set of n numbers.

   (b) Suppose that the search for key k in a binary search tree ends up in a leaf. Consider three sets: A, the keys to the left of the search path; B, keys on the search path; and C, keys to the right of the search path. Prove, or disprove by showing an example, the following statement: For any three keys $a \in A$, $b \in B$, and $c \in C$, $a \leq b \leq c$.

4. Let $S = a_1, ..., a_n$ be a sequence of n numbers. Provide a recursive formula for $l(i)$, the length r of the longest monotonically increasing sequence $a_{k_1} < \ldots < a_{k_r}$, $k_1 < \ldots < k_r$, which is a subsequence of $a_1, ..., a_i$ and $k_r = i$.

   Based on this formula give an $O(n^2)$ dynamic programming algorithm to find the length of the longest monotonically increasing subsequence of S. Justify that the running time is indeed $O(n^2)$.

   Example:

   For $S = 2, 5, 4, 8, 6, 7, 10, 5$, $l(3) = 2$ and the desired monotonic subsequence is $2, 4$. Likewise $l(4) = 3$ and, in fact there are two different subsequences witnessing this equality. Note that $l(8) = 3$ while the length of the longest monotonic subsequence of our sequence S is 5.

5. Let X be a set consisting of n elements. Prove that there are exactly 3^n pairs $\langle A, B \rangle$, where A, B are subsets of X and $A \subseteq B$.

SELECT ONE OF PROBLEMS BELOW AND SOLVE IT. WRITE ON PAGE ONE YOUR SELECTION. **DO NOT INCLUDE SOLUTIONS OF MORE THAN ONE PROBLEM, ONLY THE PROBLEM LISTED ON PAGE ONE WILL COUNT**.

(The two requested topics are: *Graph Algorithms* and *Computational Complexity.*)

6. (*Graph Algorithms*) A tournament is a directed graph such that for all nodes $u \neq v$, exactly one of the edges $\langle u, v\rangle$ and $\langle v, u\rangle$ is present.

   Given a directed graph $G = \langle V, E\rangle$, a set D is a *dominating set* in G if for every node $v \notin D$, there is a node $d \in D$ such that $\langle d, v\rangle \in E$ (in such case, we say that d *dominates* v).

   Show that every tournament with n nodes has a dominating set of size $\Theta(\log n)$. (Show that in any tournament there is a player who beats at least half of remaining players; add this player to the dominating set. What is left to dominate?) Design an efficient algorithm finding such a dominating set.

7. (*Computational Complexity*) Let UP be the subset of NP consisting of those languages accepted by nondeterministic polynomial time Turing machines that accept on at most one computation per string. (The "U" stands for "unique".)

   Let f be a one-to-one, polynomial-time computable function such that there is a polynomial p such that $|x| \leq p(|f(x)|)$. (Note that f computable in polynomial time only implies the existence of a polynomial q such that $|f(x)| \leq q(|x|)$.)

   Show that if f has no polynomial-time computable inverse, then P $\neq$ UP. In other words, if there is no polynomial-time computable g such that for all x, $g(f(x)) = x$, then there is a language L in UP $\setminus$ P. Give a description of L in terms of f, and show that it is in UP, then show that L in P would imply that f is polynomial-time invertible.

   For partial credit, show that if f has no polynomial-time computable inverse, then P $\neq$ NP.