Written Theory Qualifier Exam

Your number: _____

Time limit: 2.5 hours. Use only the notes supplied by the Department

Fall 2012, CS Department, IIT

For every question, please write your answer in a clean and concise way. Use additional pages, start a new page with each problem and write only on one side of the paper.

Use procedures if you want – – marking clearly what the parameters are and what they do, and with what running time in terms of its parameters. Unless the procedures are from the textbooks, write pseudocode for the procedures.

Problem 1.

We want to search for z in an ordered array $x_1 < x_2 < \cdots < x_n$. For simplicity, we consider only the worst-case in unsuccessful searches.

1. One way to prove that binary search is optimal is to model it with a binary tree. Let T_h be largest number of external nodes in a binary tree that has height at most h. Prove that $T_h = 2T_{h-1}$ and that $T_0 = 1$. Use this to prove that (unsuccessful) binary search, which uses $\lceil \log_2 n \rceil$ comparisons is optimal.

2. The analysis in the previous part assumes that a "greater than" result costs the same as a "less than" result when making a comparison $z : x_i$. Suppose instead that the search is lopsided so that a "greater than" result costs twice as much as a "less than" result. Now, let T_h be largest number of external nodes in a binary tree that has *search cost* at most h. Prove that $T_h = T_{h-1} + T_{h-2}$ and that $T_0 = T_1 = 1$.

3. Use the result in the previous part to prove that $\lceil \log_\phi n \rceil$ comparisons is optimal, $\phi = (1 + \sqrt{5})/2$, for such a lopsided search.

Problem 2.

You are given a tree represented by the following data structure. Nodes have three fields: element, parent, and auxiliary. The auxiliary field is 0 and can be changed during traversals. However, your algorithm should finish with the tree as in the input.

You are given the location (or pointer to) two nodes x and y. The goal is to print the elements on the path from x to y in tree, in time proportional to the length of the path. That is, if the tree has n nodes and this path has length k, the running time of the algorithm should be $O(k)$ ($O(n)$ is not good enough, and the tree does not have to be binary and/or balanced).

Argue that your algorithm algorithm indeed has running time $O(k)$. Don't forget to bring back the tree to its original state. Present full pseudocode.

Problem 3.

An independent set of an undirected graph $G=(V,E)$ is a subset I of V, such that no two vertices in I are adjacent. That is, if $u,v \in I$, then $(u,v) \notin E$. A ***maximal independent set*** M is an independent set such that, if we were to add any additional vertex to M, then it would not be independent any longer. Every graph has a maximal independent set. (Can you see this? This question is not part of the exercise, but it is worth thinking about.) Give an efficient algorithm that computes a maximal independent set for a graph G.

Present full pseudocode and analyze the running time. $O(|V| + |E|)$ is achievable and needed for full marks.