Analysis of Algorithms

Comprehensive Exam

Fall 2009

ID:

You are allowed to use white paper and pens.

1. For two positive integers *a*, and *n*, naively computing a^n takes multiplying *a* with itself *n*-times, i.e., Θ(n) time. Write a *divide-and-conquer* algorithm for the same purpose. Show your algorithm's asymptotic time-complexity by setting up the corresponding recurrence equation and solving it.

2a. Explain in a line or two how can one detect in O(*n*) time whether an input directed graph is a tree or not, where *n* is the number of vertices of the graph.

2**.** The following is a recurrence formula. Write a Dynamic Programming algorithm for computing all a[i,j]'s, where i and j are integers between 0 and a constant N>0.

$a[i, 0]= -i$, $a[0, j]= -j$,

$a[i, j] = \max\{$

$a[i,k]-2,\ \ 0\leq k< j;$

$a[p, j]-2,\ \ 0\leq p< i;$

$a[p-1, k-1] -1\},\ 0\leq p< i,\ 0\leq k< j\}$, for both i and j >0.

Analyze the time-complexity of the algorithm.

3. Answer *true/false* for the following sentences. Explain your answer *if* you cannot answer as *true/false*.

a. Sets of NP-class problems and NP-complete problems have null intersection.

b. The set of NP-complete problems is a superset of the P-class problems.

c. The set of NP-hard problems is a superset of the NP-complete problems.

d. In order to prove a problem *X* to be NP-hard one needs to develop a polynomial transformation from a known NP-hard problem *Y* to *X*.

e. 2-SAT (where each clause in a Boolean Satisfiability problem (SAT) has two literals) is a P-class problem. So, SAT is in $\text{P} \cap \text{NP-complete}$.

f. There may be an exponential-time algorithm for finding the shortest paths between all pairs of nodes in a directed and weighted graph.

g. Naive *DFS* algorithm takes $O(n^2)$ time for a graph with n nodes.

h. This sentence (in question 5h) is false.

i. Suppose that G is a connected and undirected graph. If the edge e is crucial in keeping the graph G connected, then e is a tree edge in the depth-first search spanning-tree of G.

j. Suppose that e is a minimum weight edge of a weighted undirected graph G, and all the edge weights are distinct. Then e is always contained in the minimum spanning tree of G.

k. *Minimum Spanning Tree* generation for a weighted undirected graph is NP-complete problem.

4.a) What is the value of the variable *count* in terms of *n* after the following algorithm-fragment is executed?

(1) count = 0;
(2) For *i* = 1 through *n* do
(3)   For *p* = 1 through 3 do
(5)   For *k* = 0 through *i/2* do
(4)   count = count +1;
        end for loops;

b) What is the asymptotic time complexity of this algorithm?

5. Write a recursive divide-and-conquer algorithm for computing the value of a polynomial of degree n-1 for a given value of its variable. Analyze its time-complexity.