Indian Institute of Technology, Kharagpur

Date..... FN/AN Time: 2 Hrs Full Marks: 30 No. of Students: 89
Mid (Autumn) Sem 2012-13 Deptt: MA/EC/CS/IM/HS/BT/EX/CH
Sub. No. MA 21007 Subject Name: Design and Analysis of Algorithms

Instruction: Answer **all** questions.

Question 1 [1 + 1 + 1 = 3 marks]

a) Prove or disprove: $O(f(n) + g(n)) = f(n) + O(g(n))$, if $f(n)$ and $g(n)$ are positive for all n.

b) Formulate and prove by induction a rule for the sums $1^2, 2^2 - 1^2, 3^2 - 2^2 + 1^2, 4^2 - 3^2 + 2^2 - 1^2, 5^2 - 4^2 + 3^2 - 2^2 + 1^2$, etc.

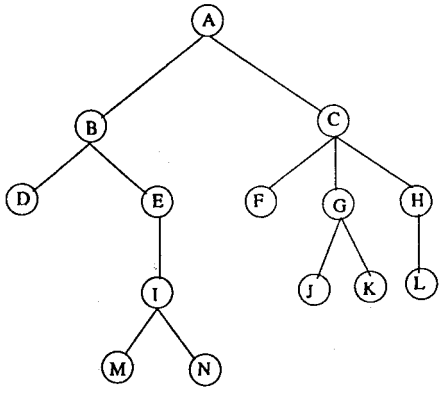
c) Is the operation deletion "commutative" in the sense that deleting x and then y from the binary search tree leaves the same tree as deleting y and then x? Argue why it is or give a counterexample.

Question 2 [3 + 2 + 3 = 8 marks]

a) Find the number of distinct binary trees with n nodes.

b) Prove that the recurrence $T(n) = mT(n/2) + an^2$ has the solution $T(n) = O(n^{\log_2 m})$

c) Consider the tree:



Starting with the root, and with the convention that children are visited in *reverse* alphabetical order (i.e. from right to left in the figure), list the order that the vertices are visited in (*i*) pre-order, (*ii*) post-order, and (*iii*) level-order.

——P.T.O.——

Question 3 [2 + 4 = 6 marks]

a) Find the average time to build a binary search tree. You should express your answer using O-notation

b) Starting with an empty binary search tree, insert the following items (in this order):

$$1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10,\ 11,\ 12,\ 13,\ 14,\ 15,\ 16.$$

Let T be the binary search tree obtained above.

(i) Draw the binary search tree T_1 after performing DELETE(16), SPLAY(15) on T.

(ii) Draw the binary search tree T_2 after performing DELETE(15), DELETE(14), SPLAY(13) on T_1.

(Here DELETE(x) deletes node with key x and SPLAY(x) performs a sequence of rotations to make the node with key x become the root of the binary search tree.)

Question 4 [6 + 2 = 8 marks]

a) Consider the following sorting methods: Insertion Sort, Selection Sort, Merge Sort, and Quick Sort. What is the running time using O-notation for each method

(i) When all the the array values are equal?

(ii) When the values are in order?

(iii) When the values are in reverse order?

Explain your answers.

b) Suppose you are given a sequence S of n elements each of which is an integer in the range $[0, n^2 - 1]$. Design an algorithm for sorting S in $O(n)$ time.

Question 5 [3 + 2 = 5 marks]

a) Write a pseudo-code for finding the k-th largest element in an array of n elements in linear time. Illustrate your algorithm on the following sequence by finding the 3-rd largest element: $3.72, 4.11, 5.34, 6.25, 7.76, 2.66, 1.83, 0.41, 9.10$

b) Explain why the average computing time for your algorithm is linear?

——The End——