PES University

Department of Computer Science & Engineering

Data Structures and its Applications (UE19CS202)

UNIT 3

Trees and Heaps

Question Bank

1) Prove that the root of a binary tree is an ancestor of every node in the tree except itself.
2) Prove that a node of a binary tree has at most one parent.
3) How many ancestors does a node at level n in a binary tree have? Prove your answer.
4) Write recursive and non recursive algorithms to determine:
   a) The number of nodes in a binary tree
   b) The sum of the contents of all the nodes in a binary tree
   c) The depth of a binary tree
5) Write an algorithm to determine if a binary tree is:
   a) Strictly binary
   b) Complete
   c) Almost complete
6) Prove that a strictly binary tree with n leaves contains 2n-1 nodes.
7) Given a strictly binary tree with n leaves, let level(i) for i between 1 and n equal the level of the ith leaf. Prove that

$$\sum_{i=1}^{n} \frac{1}{2^{level(i)}} = 1$$

8) Prove that the nodes of an almost complete strictly binary tree with n leaves can be numbered from 1 to 2n – 1 in such a way that the number assigned to the left child of the node numbered i is 2i and the number assigned to the right child of the node numbered i is 2i + 1.
9) Two binary trees are similar if they are both empty or if they are both nonempty, their left subtrees are similar, and their right subtrees are similar. Write an algorithm to determine if two binary trees are similar.
10) Two binary trees are mirror similar if they are both empty or if they are both nonempty and the left subtree of each is mirror similar to the right subtree of the other. Write an algorithm to determine if two binary trees are mirror similar.
11) Write algorithms to determine whether or not one binary tree is similar and mirror similar (see the previous exercises) to some subtree of another.
12) Develop an algorithm to find duplicates in a list of numbers without using a binary tree. If there are n distinct numbers in the list, how many times must two numbers be compared for equality in your algorithm? What if all n numbers are equal?

13) a) Write an algorithm that accepts a pointer to a binary search tree and deletes the smallest element from the tree.

 b) Show how to implement an ascending priority queue as a binary search tree. Present algorithms for the operations pqinsert and pqmindelete on a binary search tree.

14) Write an algorithm that accepts a binary tree representing an expression and returns the infix version of the expression that contains only those parentheses that are necessary.

15) Write a C function that accepts a pointer to a node and returns TRUE if that node is the root of a valid binary tree and FALSE otherwise.

16) Write a C function that accepts a pointer to a binary tree and a pointer to a node of the tree and returns the level of the node in the tree.

17) Write a C function that accepts a pointer to a binary tree and returns a pointer to a new binary tree that is the mirror image of the first (that is, all left subtrees are now right subtrees and vice versa).

18) Write C functions that convert a binary tree implemented using the linked array representation with only a parent field (in which the left child's parent field contains the negative of the pointer to its parent and right child's parent contains a pointer to its parent) to its representation using left and right fields, and vice versa.

19) Write C routines to traverse a binary tree in preorder and postorder .

20) Implement inorder traversal, maketree, setleft, and setright for right in-threaded binary trees under the sequential representation.

21) Write C functions to create a binary tree given the preorder and inorder traversals of that tree. Function should accept two character strings as parameters. The tree created should contain a single character in each node.

22) The solution to the Towers of Hanoi problem for n disks can be represented by a complete binary tree of level n-1 as follows.

 a) Let the root of the tree represent a move of the top disk on peg frompeg to peg topeg. (We ignore the identification of the disks being moved, as there is only a single disk [the top one] that can be moved from any peg to any other peg.) If nd is a leaf node (at level less than n-1) representing the movement of the top disk from peg x to peg y, let z be the third peg that is neither the source nor the target of node nd. Then left(nd) represents a move of the top disk from pegx to pegy and right(nd) represents a move of the top disk from peg z to peg y. Draw sample solution trees as described previously for n=1, 2, 3, and 4, and show that an inorder traversal of such a tree produces the solution to the Towers of Hanoi problem.

 b) Write a recursive C procedure that accepts a value for n and generates and traverses the tree as discussed previously.

 c) Because the tree is complete, it can be stored in an array of size 2^n-1. Show that the nodes of the tree can be stored in the array so that a sequential traversal of the array produces the inorder traversal of the tree, as follows: The root of the tree is in position

$2^{n-1} – 1$; for any level j, the first node at that level is in position 2^{n-1-j} – 1 and each successive node at level j is 2^n-j elements beyond the previous element at that level.

d) Write a nonrecursive C program to create the array as described in part c and show that a sequential pass through the array does indeed produce the desired solution.

e) How could the preceding programs be extended to include within each node the number of the disk being moved?

23) Prove that the leftmost element node at level n in an almost complete strictly binary tree is assigned the number 2^n.

24) Show how to represent a linked list as an almost complete binary tree in which each list element is represented by one tree node. Write a C function to return a pointer to the kth element of such a list.

25) Write a C function compute that accepts a pointer to a tree representing an expression with constant operands and returns the result of evaluating the expression without destroying the tree.