EE4013
C and Data Structures

EE18BTECH11050

Krati Arela

October 11, 2021

Assignment 1

GATE 2018 Paper-CS section Q.20

The postorder traversal of a binary tree is 8,9,6,7,4,5,2,3,1. The inorder traversal of the same tree is 8,6,9,4,7,2,5,1,3. The height of a tree is the length of the longest path from the root to any leaf. The height of the binary tree above is _____?

Solution

The height of the above binary tree is **4**.

Postorder traversal

Postorder traversal of a binary tree is a depth-first search in which we first traverse the left subtree, then traverse the right subtree, and finally visit the root node.

Inorder traversal

In **inorder traversal** we first traverse the left subtree, visit the root and then traverse the right subtree.

Height of a binary tree

Given n nodes in a binary tree, maximum and minimum height of the binary tree are:

Maximum height

$$h_{max} = n - 1$$

Minimum height

$$h_{min} = floor(log_2 n) = ceil(log_2(n+1) - 1)$$

Minimum height of a binary tree

A complete binary tree will have minimum height. Given n,

$$2^{h_{min}} \leq n \leq 2^{h_{min}+1} - 1$$

$$\implies h_{min} \leq log_2 n, h_{min} \geq log_2(n+1) - 1$$

$$\implies h_{min} = floor(log_2 n) = ceil(log_2(n+1) - 1)$$

Here $n = 9$, so $h_{max} = 8$ and $h_{min} = 3$.

Algorithm

- The last node in postorder[] is the root, which is "1" in this case
- Search “1” in inorder[] to find left and right subtrees of root.
- Recur the above process for following two:
    - Recur for $inorder = [8, 6, 9, 4, 7, 2, 5]$ and $postorder = [8, 9, 6, 7, 4, 5, 2]$. The created tree will be left child of root.
    - Recur for $inorder = [3]$ and $postorder = [3]$. The created tree will be right child of root.

Construct Binary Tree

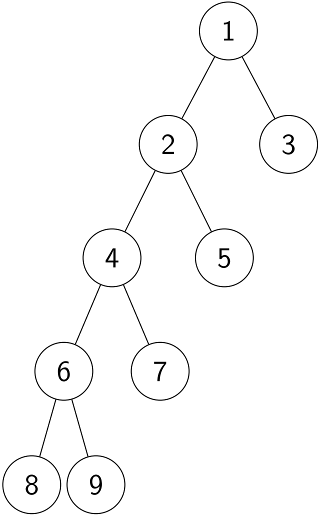

Figure 1: Constructed Binary Tree

Binary Tree

The constructed tree from given algorithm shows that height of tree is 4.

Generating Binary Tree

We can generate random inorder and postorder traversal arrays for any given height h.

- Initialize array of size $2 * h + 1$ for inorder and postorder arrays.
- Initialize root of binary tree as 1 and recursively create child nodes of root with values as $2 * (i + 1)$ and $2 * (i + 1) + 1$, with $i = 0, 1, .., h - 1$.
- Insert the values of the nodes of the tree in inorder or postorder traversal as required in array and return the array.

By generating the inorder and postorder array randomly, the algorithm to find height of a binary tree can be verified.