Tree Exercise 2

Cite sources; write solutions/explanations.

ITEM 1

1 What is the difference between a perfectly balanced binary tree and a complete binary tree?

2 What is the difference between a complete binary tree and a full binary tree?

3 A full binary tree has a height of 5. How many nodes does it contain?

4 A complete binary tree contains 125 nodes. What is its height?

5 How many nodes are on a given level L in a full binary tree? Express your answer in terms of L.

1. In a balanced binary tree, the left and right subtrees (or depth) of every node and the two subtrees never differ more than one.

For it to be considered a complete binary tree, the last level of a binary tree must be entirely filled, and all of its leaves must be on the left side of the tree. All levels except for the last two must have two children. All nodes in the last level must also be as far left as possible.

Source: https://stackoverflow.com/questions/14756648/difference-between-complete-binary-tree-and-balanced-binary-tree

2. The existence of gaps and the number of child nodes for each node are the key differences between a complete binary tree and a full binary tree. A complete binary tree might include gaps in the last level, but a complete binary tree has every node with 0 or 2 child nodes. The node in a complete binary tree should be filled from left to right. A full binary tree has no order of filling nodes in the last level.

In a practical aspect, complete binary trees are used in heap or heavy-based data structures. A full binary tree can have applications such as Huffman coding.

Sources:

https://www.geeksforgeeks.org/difference-between-full-and-complete-binary-tree/

Full Binary Tree vs Complete Binary Tree | What's the difference? - javatpoint

Full v.s. Complete Binary Trees (pdx.edu)

3. Formula for finding the number of nodes using height is 2^(h+1)-1. Therefore, it will be:

= (2^(5+1)) – 1

= (2^6)-1

= 64 – 1

= 63

The total number of nodes is 63. If a binary search tree has height, then the minimum number of nodes is h+1.

Source: https://www.geeksforgeeks.org/relationship-number-nodes-height-binary-tree/

4. Formula for finding the number of finding the height using the number of nodes from a binary tree is log2(n+1)−1.

= log2(125 + 1) - 1

= log2(126) - 1

≈ 6 - 1

= **5**

The number of nodes in a full binary tree is linked to its height by the formula n = 2(h+1) - 1, where n is the number of nodes and h is the tree's height.

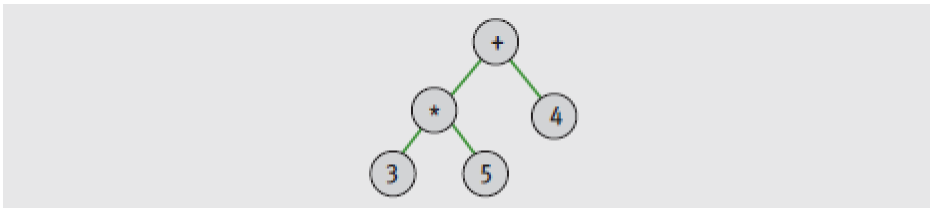
Source: https://cs.stackexchange.com/questions/6277/why-is-the-minimum-height-of-a-binary-tree-log-2n1-1

5. Each level of a full binary tree has twice as many nodes as the level before it. There is one node at the root level (L = 1). The next level (L = 2) has two nodes, then four at level three, eight at level four, and so on. Therefore, the answer is **2^l-1.** The number of nodes in a binary tree follows a patter, and the root level has one node. Each next subquest level has twice the number of nodes.

Source: https://eng.libretexts.org/Courses/Delta_College/C_-_Data_Structures/14%3A_Binary_Trees/14.03%3A_Binary_Tree_Properties

ITEM 2

1 Write the expression represented by the following expression tree in infix, prefix, and postfix notations. (*Hint*: Use the inorder, preoder, and postorder traversals described in this section to obtain your answers.)



Insert screenshots of code and output (create nodes and tree, plus call the traversal methods).

Infix Notation:

3 * 5 + 4

Prefix Notation:

Expression	Stack	Prefix(reversed)
	(	
4	(	4
	(	4
+	(+	4
	(+	4
5	(+	4 5
	(+	4 5
*	(+ *	4 5
	(+ *	4 5
3	(+ *	4 5 3
)		4 5 3 * +

+ * 4 5 3

Postfix notation:

Expression	Stack	Prefix(reversed)
	(	
3	(	3
	(	3

*	(*	3
	(*	3
5	(*	3 5
	(*	3 5
+	(+	3 5 *
	(+	3 5 *
4	(+	3 5 * 4
)		3 5 * 4 +

3 5 * 4 +