CS & DA

GATE

Data Structure through Python

DPP: 2

Tree

Q1 Suppose a binary search tree with 1000 distinct elements is also a complete binary tree. The tree is stored using the array representation of binary heap trees. Assuming that the array indices start with 0, the 3rd largest element of the tree is stored at index _____ .

Q2 What will be post order traversal of a binary Tree T, if preorder and inorder traversals of T are given by ABCDEF and BADCFE respectively ?

(A) BEFDCA (B) BFDECA

(C) BCFDEA (D) BDFECA

Q3 We create a binary search tree B_1 by inserting the numbers 1, 2, 3, 4, 5 into an empty binary search tree. We create another binary search tree B_2 by inserting the numbers into an empty binary search tree in the reverse order. What is the difference between the right-most element of B_1 and the left-most element of B_2?

(A) 1 (B) 3

(C) 4 (D) 2

Q4 Consider that N distinct elements (N > 3) are inserted into an initially empty binary search tree (BST). Which of the following statements are true ?

(A) The worst case height of the resulting BST is $\log_2 N$

(B) None of the above

(C) Swapping the order of insertion of any two elements can always half the height of the resulting BST.

(D) Consider that a given order of insertion result in a BST of height N. One can always find two elements in the original where swapping the order of insertion of the two elements can half the height of the resulting BST.

Q5 In searching an element in a binary tree, number of comparisons is _____:

(A) O(log n – 1)

(B) O(log n – 2)

(C) O(log n)

(D) O(n log n)

GATE

Answer Key

Q1 509

Q2 (D)

Q3 (C)

Q4 (D)

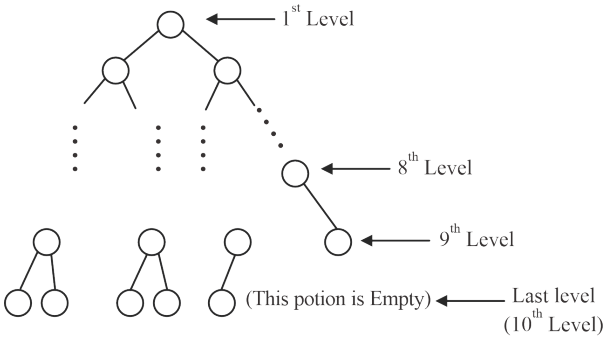
Q5 (C)

GATE

Hints & Solutions

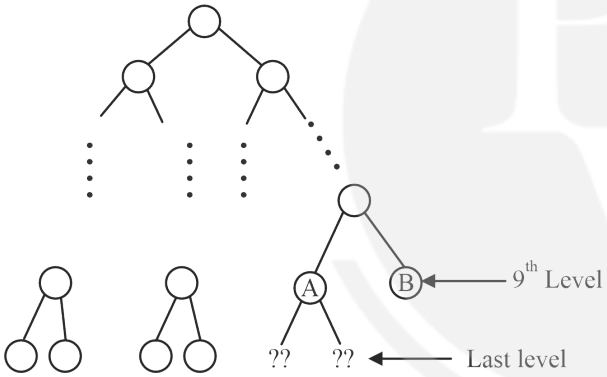
Q1 Text Solution:

For 1000 elements, there will be 10 levels (root is at level 1). But last level will not be completely filled since there are only 1000 elements.



As you can see, some portion is empty. Since largest element can not have children as there are not enough nodes. - we need 1023 nodes to fill all levels.

Now lets see if A in below diagram has children ?



Again, we do not have enough nodes that's why A will not be having subtree. (This is key point in this question, let's have a quick explanation of why A doesn't have any children.)

Total nodes till 2nd last level = 29-1 = 511

If we want to fill the last level completely, we need = 210 11023 nodes.

A and B can have just 2 children each, so for 1023-4 = 1019 nodes, A and B does not have children.

for 1020 nodes, A has a left child

for 1021 nodes, A has both children, for 1022 nodes, B has a left child, and for 1023 nodes, B has both children.

Now,

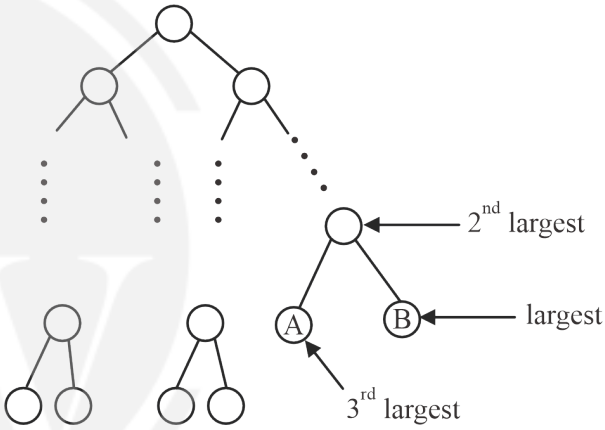
largest will be at rightmost of 9th level, (Easy to see.....just keep going right) 2nd largest will at rightmost of 8th level, (because if you want to find larger than this then you need to go on right side and there is just one element on right side)

Where will be 3rd largest?

Since we know that inorder of BST is sorted hence 3rd largest will be a element just before 2nd largest in inorder. What we call such element? - inorder predecessor.

Hence 3rd largest will be inorder predecessor of 2nd largest. - which is A in above diagram.

Lets find index of 2nd largest then we can find index of 3rd largest.



index of 2nd largest. (Assuming index starts from 1)

$1 \rightarrow 3 \rightarrow 7 \rightarrow 15 \rightarrow 31 \rightarrow 63 \rightarrow 127 \rightarrow 255$

Index of 3rd largest = 510

Since indices are starting from 0 hence answer is 509.

Q2 Text Solution:

The given data,

preorder = ABCDEF

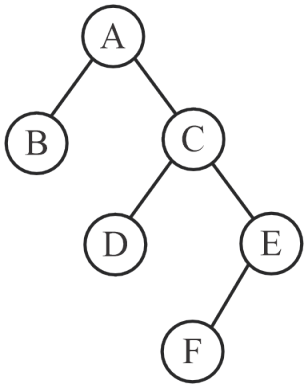
In order BADCFE

Tree traversal			
Method	Inorder	Preorder	Postorder
Sequence			

GATE

	Left Sub-tree	Root	Left Sub-tree
	Root	Left Sub-tree	Right Sub-tree
	Right Sub-tree	Right Sub-tree	Root

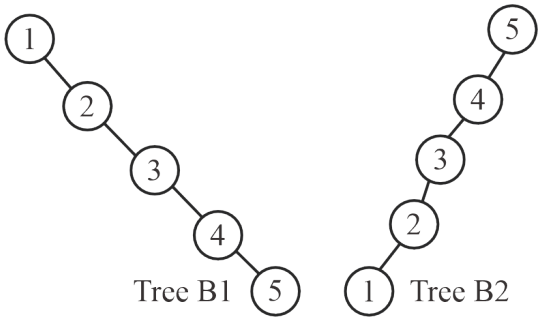
The binary tree for the traversal is,



Post order for the above tree is,
BDFECA
Hence the correct answer is BDFECA.

Q3 Text Solution:

A binary search tree, also known as an ordered or sorted binary tree is a rooted binary tree data structure in which each internal node stores a key that is higher than all keys in the node's left subtree but less than those in the node's right subtree.



right-most element of $B_1 = 5$
left-most element of $B_2 = 1$
Difference between both $B_1 - B_2 = 5 - 1 = 4$
Hence the correct answer is 4.

Q4 Text Solution:

Consider that a given order of insertion result in a BST of height N. One can always find two elements in the original where swapping the order of insertion of the two elements can half the height of the resulting BST.

Q5 Text Solution:

Note:
The answer is according to the official key and the previous year's paper it was given as a binary tree.
Binary tree:
A binary tree is a tree data structure in which each node has a maximum of two children, known as the left and right children.
Binary search tree:
A rooted binary tree data structure whose internal nodes each store a key greater than all the keys in the node's left subtree and less than those in the node's right subtree is known as a binary search tree, also known as an ordered or sorted binary tree.
Search Operation of binary search tree operation:
In a binary search tree if the root node has null then it returns null and unsuccessful search.
Otherwise, compare searching node with root data if it is true then searching element is found and successful search.

- If the searching node is not found at the root node compare with either the left subtree or right subtree.
- A searching node is less than root node data compare with left subtree root data and follow the previous steps.
- A searching node is greater than root node data compare with right subtree root data and follow the previous steps.