HackerRank
For Work

Info6205_Summer18_Quiz8

35 minutes

Question - 1
BST

SCORE: **5 points**

What is the worst case time complexity guarantee for search, insert and delete operations in a Binary Search Tree?

○ O(log n) for all

● O(n) for all

○ O(log n) for search and insert, O(n) for delete

○ O(log n) for search, O(n) for insert and delete

Question - 2
BST

SCORE: **5 points**

The following numbers are inserted into an empty binary search tree in the given order: 10, 1, 3, 5, 15, 12, 16. What is the depth of the binary search tree?

○ 2

● 3

○ 4

○ 6

Question - 3
BST

SCORE: **5 points**

What’s the depth of a complete tree with N nodes? (The depth of the root node is zero)

○ logN+1

○ N

● logN

○ N/2

Question - 4
BST

SCORE: **5 points**

Suppose the numbers 7, 5, 1, 8, 3, 6, 0, 9, 4, 2 are inserted in that order into an initially empty binary search tree. The binary search tree uses the usual ordering on natural numbers. What is the in-order traversal sequence of the resultant tree?

- ○ 7 5 1 0 3 2 4 6 8 9
- ○ 0 2 4 3 1 6 5 9 8 7
- ● 0 1 2 3 4 5 6 7 8 9
- ○ 9 8 6 4 2 3 0 1 5 7

Question - 5

BST

SCORE: **30 points**

Implement binary search tree Insert and Search operations.

```
The first line of the input is number of
operations that will happen.
The subsequent lines represent the operation and
key, value pair.

Sample Input1:
4 // number of operations
1,5 //(operation,nodeValue)
1,1
1,3
2,3

Sample Output1:
true

Sample Input2:
4 // number of operations
1,5 //(operation,nodeValue)
1,1
1,3
4

Sample Output2:
3

Types of Operation:
1 -> insert (inserts an element into the tree)
2 -> search (returns true/false depending on
whether or not the value is found)
3 -> isEmpty (returns true for an empty tree else
false)
4 -> countNodes (returns number of nodes in the
tree)
5 -> preorder traversal
6 -> inorder traversal
7 -> postorder traversal
```