

Quiz

It's quiz time! Test yourself by solving these questions about binary search trees.

1

The worst-case time complexity of insert, search and delete operations in a binary search tree is as follows:

- ☐ A) $O(1)$
- ☐ B) $O(\log n)$
- ☐ C) $O(n)$
- ☐ D) $O(n \log n)$

2

BST property:

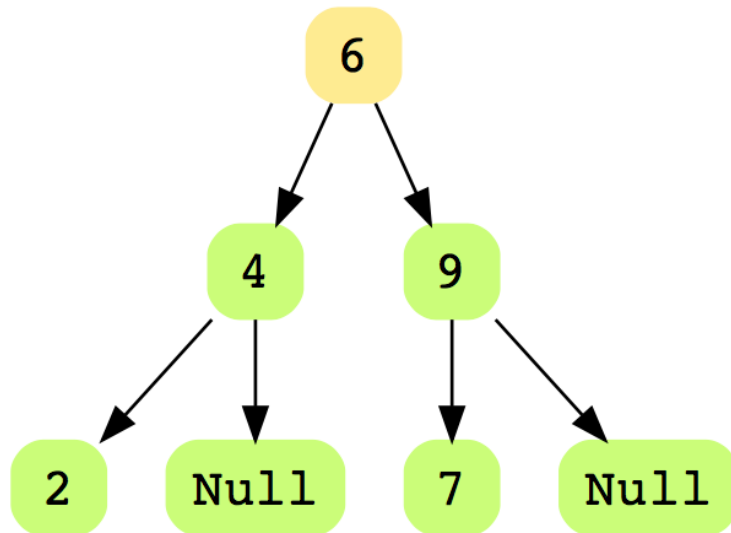
The value of the left child of any node in a binary search tree will be less than whatever value we have in that node, and the value of the right child of a node will be equal to the value in that node.

- ☐ A) True

☐ B) False

3

Is the following tree a valid binary search tree?



☐ A) True

☐ B) False

4

What is the average time complexity of the following code?

```
def search(self, find_val):
    return self.search_helper(self.root, find_val)

def search_helper(self, current, find_val):
    if current:
        if current.data == find_val:
            return True
        elif current.data < find_val:
            return self.search_helper(current.right, find_val)
        else:
```

```
else:  
    return self.search_helper(current.left, find_val)
```

- ☐ A) $O(n/2)$
- ☐ B) $O(\log n)$
- ☐ C) $O(n)$
- ☐ D) $O(n \log n)$

5

Which of the following is a valid implementation of the `Node` class for a binary search tree?

☐ A)

```
class Node:  
    def __init__(self, val):  
        self.val = val  
        self.Child = None  
        self.parent = None
```

☐ B)

```
class Node:  
    def __init__(self, val):  
        self.val = val  
        self.parent = None
```

☐ C)

```
class Node:  
    def __init__(self, val):  
        self.val = val  
        self.leftChild = None
```

```
self.rightChild = None
```

☐ D)

```
class Node:  
    def __init__(self, val):  
        self.val = val  
        self.Child = None
```

CHECK ANSWERS