

Question - 1
Question 1

SCORE: 5 points

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

- ☒ $O(n)$ for all
- ☐ $O(\log n)$ 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
Question 2

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 height of the binary search tree?

- ☐ 2
- ☒ 3
- ☐ 4
- ☐ 6

Question - 3
Question 3

SCORE: 5 points

Which of the following is true about Red Black Trees?

- ☐ At least one child of every black node is red
- ☐ The root may be red
- ☐ A leaf node may be red
- ☒ None of the above

Question - 4
Question 4

SCORE: 5 points

Is the following statement true? A Red-Black Tree which is also a perfect Binary Tree can have all black nodes.

☒ True

☐ False

Question - 5
Left Leaning Red Black Tree Implementation

SCORE: 30 points

You are required to implement following methods of RedBlackBST class:

1. get method for standard BST search

```
public Value get(Key key) {}
```

2. put method for RedBlackBST insertion
// hint: need to keep the Left Leaning RedBlackBST structure after each insertion

```
public void put(Key key, Value val) {}
```