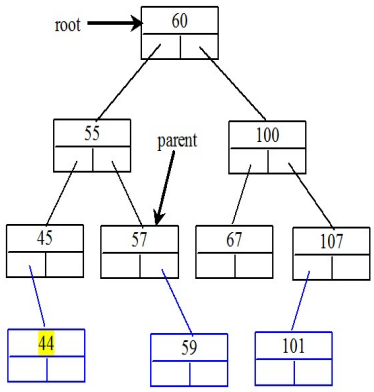
Due to the print book page limit, we cannot inlcude all good CheckPoint questions in the physical book. The CheckPoint on this Website may contain extra questions not printed in the book. The questions in some sections may have been reordered as a result. Nevertheless, it is easy to find the CheckPoint questions in the book on this Website. Please send suggestions and errata to Dr. Liang at y.daniel.liang@gmail.com. Indicate the book, edition, and question number in your email. Thanks!

Chapter 25 Check Point Questions

Section 25.2

▼25.2.1

Show the result of inserting 44 into Figure 25.4b.



Hide Answer

▼25.2.2

Show the inorder, preorder, and postorder of traversing the elements in the binary tree shown in Figure 25.1b.

Inorder: A F G M R T
Preorder: G F A R M T
Postorder: A F M T R G

Hide Answer

▼25.2.3

If a set of elements is inserted into a BST in two different orders, will the two corresponding BSTs look the same? Will the inorder traversal be the same? Will the postorder traversal be the same? Will the preorder traversal be the same?

If a set of the same elements is inserted into a binary tree in two different orders, will the two corresponding binary trees look the same? No. Will the inorder traversal be the same? Yes. Will the postorder traversal be the same? No. Will the preorder traversal be the same? No.

Hide Answer

▼25.2.4

What is the time complexity of inserting an element into a BST?

The time complexity of inserting an element to a binary tree is O(n).

Hide Answer

▼25.2.5

Implement the search(element) method using recursion.

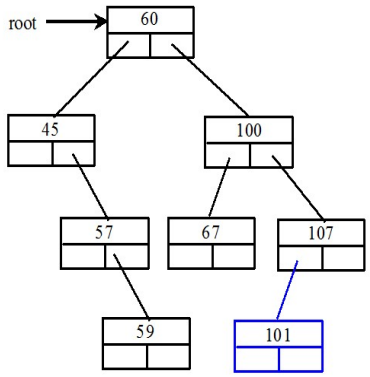
```
@Override
/** Returns true if the element is in the tree */
public boolean search(E e) {
  return search(root, e);
}

public boolean search(TreeNode<E> root, E e) {
  if (root == null)
    return false;
  else if (e.compareTo(root.element) < 0)
    return search(root.left, e);
  else if (e.compareTo(root.element) > 0)
    return search(root.right, e);
  else
    return true;
}
```

Hide Answer

Section 25.3

▼25.3.1

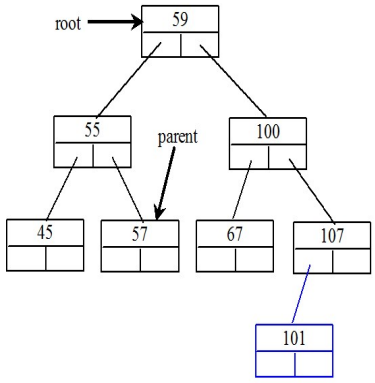
Show the result of deleting 55 from the tree in Figure 25.4b.



Hide Answer

▼25.3.2

Show the result of deleting 60 from the tree in Figure 25.4b.



Hide Answer

▼25.3.3

What is the time complexity of deleting an element from a BST?

The time complexity of deleting an element from a binary tree is O(n).

Hide Answer

▼25.3.4

Is the algorithm correct if lines 203-207 in Listing 25.4 in Case 2 of the delete() method are replaced by the following code?

```
parentOfRightMost.right = rightMost.left;
```

No. Consider the case when current is parentOfRightMost and current.left is rightMost. You have to assign rightMost.left to parentOfRightMost.left.

Hide Answer

Section 25.4

▼25.4.1

How many times will the displayTree method be invoked if the tree is empty? How many times will the displayTree method be invoked if the tree has 100 nodes?

The displayTree method will be invoked 0 time if the tree is empty. The displayTree method will be invoked 100 times if the tree has 100 nodes.

Hide Answer

▼25.4.2

In what order are the nodes in the tree visited by the displayTree method: inorder, preorder, or postorder?

The nodes in the tree are visited in preorder.

Hide Answer

▼25.4.3

What would happen if the code in lines 47-52 in BTView.java is moved to line 33?

You will see the line that connecting the nodes to be displayed on top of nodes starting from the center of the node, not from the edge of the node, because the line would be displayed after the node.

Hide Answer

▼25.4.4

What is MVC? What are the benefits of the MVC?

See the text.

Hide Answer

Section 25.5

▼25.5.1

What is an iterator?

An iterator is an object that provides a uniform way for traversing the elements in a container such as a set, list, binary tree, etc.

Hide Answer

▼25.5.2

What method is defined in the java.lang.Iterable<E> interface?

The iterator() method is defined in the java.lang.Iterable interface.

Hide Answer

▼25.5.3

Suppose you delete implements Collection<E> from line 3 in Listing 25.3, Tree.java. Will Listing 25.10 still compile?

No. Since Tree is not iterable now, the foreach loop cannot be used.

Hide Answer

▼25.5.4

What is the benefit of being a subtype of Iterable<E>?

Being a subtype of Iterable, the elements of the container can be traversed using a for-each loop.

Hide Answer

▼25.5.5

Write one statement that displays the maximum and minimum element in a BST object named tree.

```
System.out.println("Max elememt in the tree is " +
  java.util.Collections.max(tree) +
  " and the min elememt in the tree is " +
  java.util.Collections.min(tree));
```

Hide Answer

Section 25.6

▼25.6.1

Every internal node in a Huffman tree has two children. Is it true?

Yes.

Hide Answer

▼25.6.2

What is a greedy algorithm? Give an example.

A greedy algorithm is often used in solving optimization problems. The algorithm makes the choice that is optimal locally in the hope that this choice will lead to a globally optimal solution.

Hide Answer

▼25.6.3

If the Heap class in line 50 in Listing 25.9 is replaced by java.util.PriorityQueue, will the program still work?

Yes, except that you also have to change heap.getSize() to heap.size().

Hide Answer

▼25.6.4

How do you replace lines 94-99 in Listing 25.11 using one line?

```
return root.weight < t.weight ? 1 : root.weight == t.root.weight ? 0 : -1;
```

Hide Answer