Worksheet CS165: Binary Search Trees

1. Stepwise build a BST from 80, 20, 100, 10, 40, 90, 30, 50, 35

2. Give the pre-order, in, and post order traversal of the tree

3. Implement the search algorithm for a binary search tree, given the following node.

```java
class TreeNode<E> {
    E element;
    TreeNode<E> left;
    TreeNode<E> right;

    public TreeNode(E o) {
        element = o;
    }
}
```

```java
class TreeNode<E> {
    E element;
    TreeNode<E> left;
    TreeNode<E> right;

    public search(E element) {
        TreeNode<E> current = root;
        // Implementation of search algorithm
    }
}
```
3. Show how the breadth first traversal of the above tree is implemented using a queue:

<table>
<thead>
<tr>
<th>Queue</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init</td>
<td>[80]</td>
</tr>
</tbody>
</table>

4. The above tree is a binary search tree (BST). Redraw the above BST after 95 is added and 20 is deleted. Show both possible results.