CSC 212 Tutorial #9 Solution
Binary Search Trees

Problem 1

```
public int getRange() {
    BSTNode<T> p = root;
    int min, max;
    while (p.left != null)
        p = p.left;
    min = p.key;
    p = root;
    while (p.right != null)
        p = p.right;
    max = p.key;
    return max - min;
}
```

Problem 2

```
public boolean isInRange(int k) {
    if (root == null)
        return false;
    BSTNode<T> p = root;
    Boolean inRange = false;
    while (p != null) {
        if (p.key <= k) {
            inRange = true;
            break;
        }
        p = p.left;
    }
    if (!inRange)
        return false;
    p = root;
    inRange = false;
    while (p != null) {
        if (k <= p.key) {
            inRange = true;
            break;
        }
        p = p.right;
    }
```

```
    return inRange;
}
```

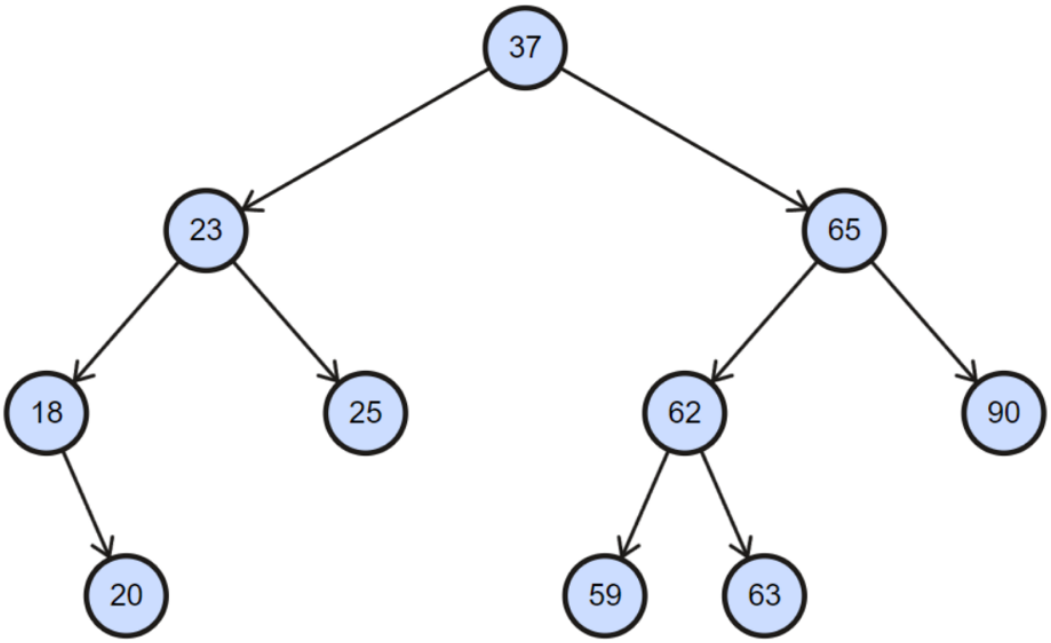
Problem 3

```
public int countNodesIn(int k) {
    BSTNode<T> p = root;
    while (p != null) {
        if (p.key == k)
            break;
        else if (k < p.key)
            p = p.left;
        else
            p = p.right;
    }
    return countNodesInRec(p);
}

private int countNodesInRec(BSTNode<T> p) {
    if (p == null)
        return 0;
    return 1 + countNodesInRec(p.left) + countNodesInRec(p.right);
}
```

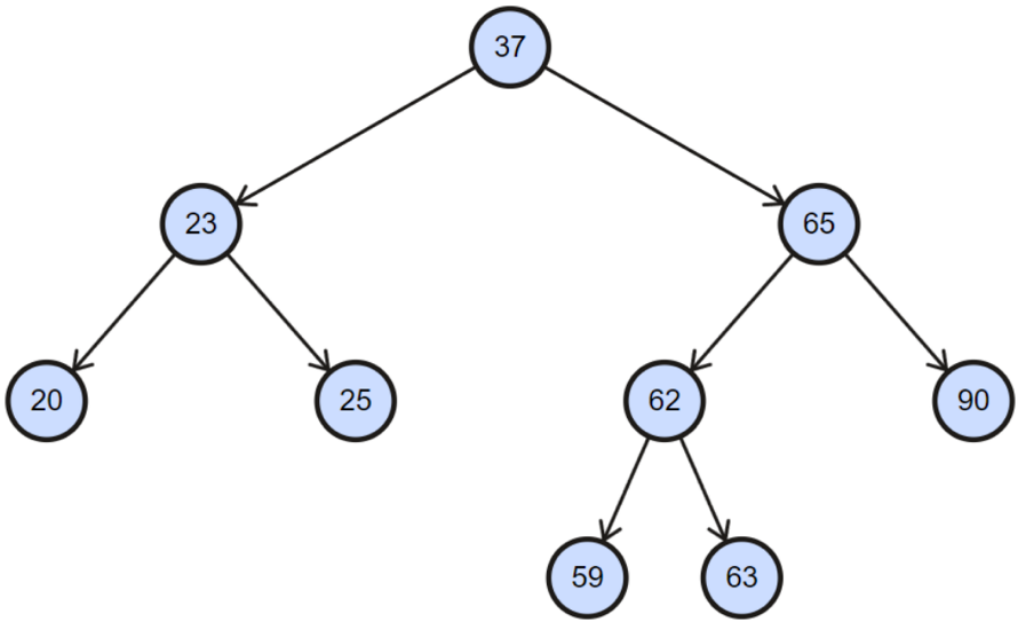
Problem 4

1. Inserting

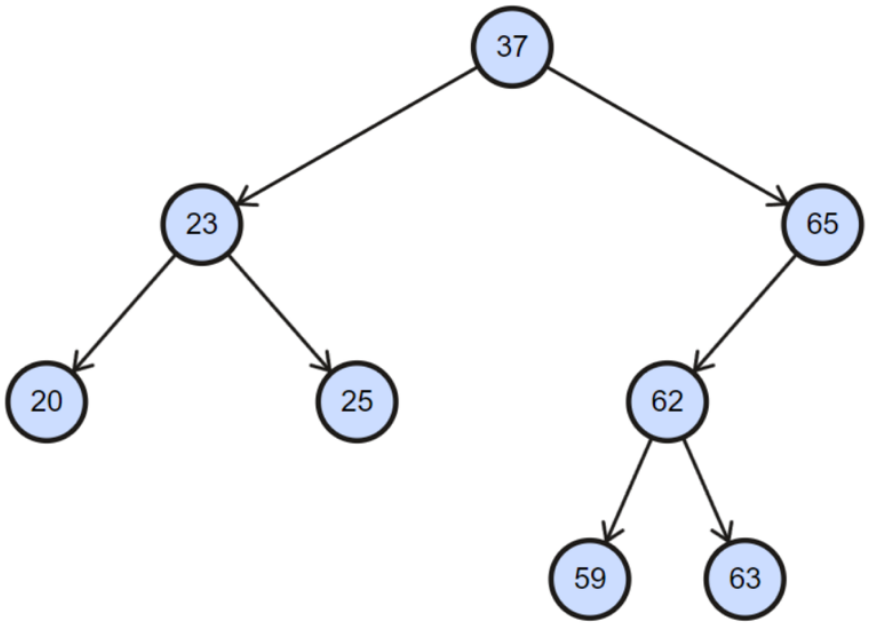


After inserting all keys

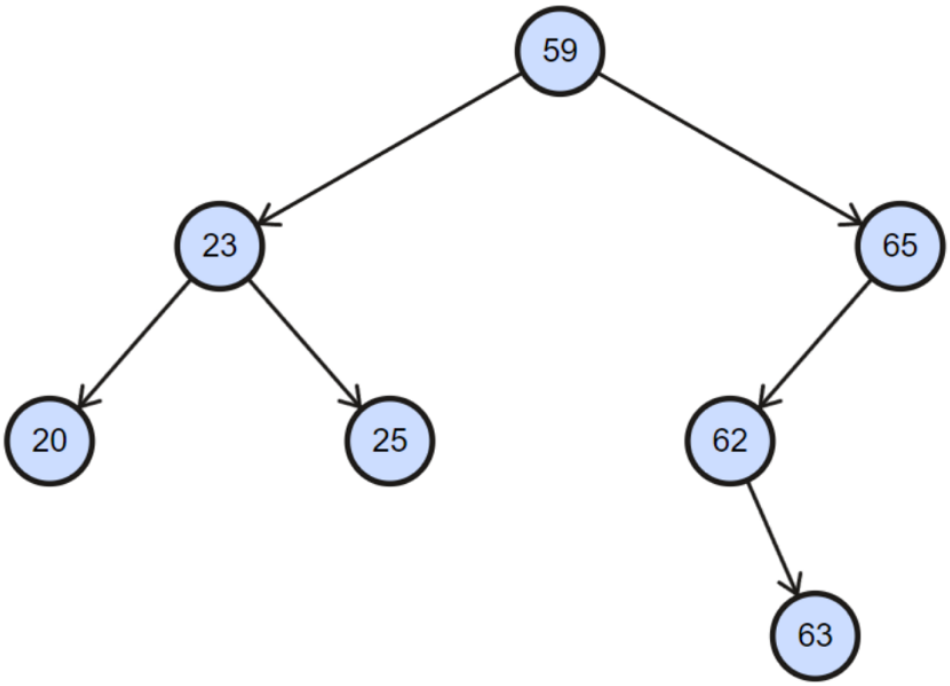
2. Removing



Removing 18 (case 2: having one child)



Removing 90 (case 1: having no children)



Removing 37 (case 3: having two children; left-most node in the right sub-tree)

3. In-order (left-root-right)