1 Written problems

1. (15 points) Draw the trees that result from inserting 10 and removing 1 and 7 from the original tree below.

<table>
<thead>
<tr>
<th>Original</th>
<th>Insert 10</th>
<th>Delete 1</th>
<th>Delete 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 7 7 8</td>
<td>7 / \ 6 9 6 9 6 9</td>
<td>7 / \ 1 8 11 4 8 11 4 11</td>
<td>8 / \ 4 10 2 5 10 2 5 10</td>
</tr>
<tr>
<td>/ \ 6 9 / \ 6 9</td>
<td>/ \ 4 / \ 4 / \</td>
<td>/ \ 4 / \ 4 / \</td>
<td></td>
</tr>
<tr>
<td>/ \ 1 8 11 / \ 1 8 11 / \ 1 8 11 / \ 1 8 11</td>
<td>/ \ 1 / \ 1 / \</td>
<td>/ \ 1 / \ 1 / \</td>
<td></td>
</tr>
<tr>
<td>1 8 11 4 8 11 4 11 2 5 10 2 5 10</td>
<td>2 5 3 3</td>
<td>2 5 3 3</td>
<td></td>
</tr>
</tbody>
</table>

Please double-check my grading for this problem. I may have mismarked your result for the deletion of the 1. If so, please let me know and I will change your score.

2. (20 points) Draw the tree that results from inserting 3 into the original 2-3 tree below. Then show the results of deleting 7 from the original tree.

<table>
<thead>
<tr>
<th>Original</th>
<th>Insert 3</th>
<th>Delete 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>--- ---</td>
<td>-</td>
<td>--- ---</td>
</tr>
<tr>
<td></td>
<td>4 6</td>
<td>4 4</td>
</tr>
<tr>
<td>--- ---</td>
<td>-</td>
<td>--- ---</td>
</tr>
<tr>
<td>/ \</td>
<td>/ \ / \</td>
<td>/ \ / \</td>
</tr>
<tr>
<td>--- - -</td>
<td>- -</td>
<td>--- ---</td>
</tr>
<tr>
<td>1 2 5</td>
<td>2 5 3 3</td>
<td>2 5 3 3</td>
</tr>
<tr>
<td></td>
<td>5 7</td>
<td>5 6 1 2 5 6</td>
</tr>
<tr>
<td>--- - -</td>
<td>- -</td>
<td>--- ---</td>
</tr>
<tr>
<td>/ \ / \</td>
<td>/ \ / \</td>
<td>/ \ / \</td>
</tr>
<tr>
<td>- - - -</td>
<td>- - - -</td>
<td>- - - -</td>
</tr>
<tr>
<td>1 3 5 7</td>
<td>1 3 5 7</td>
<td>1 3 5 7</td>
</tr>
</tbody>
</table>
3. (5 points) Is the tree below a valid red-black tree?

```
8
/ \
4R 10
/ \ / \
3 6 9 11
/ \ / \
2R 5 7
```

No – the tree does not have perfect “black balance”.

4. (10 points) Draw the tree that results from inserting 2 into the red-black tree below.

```
original insert 2

5
/ \ / \
3R 6 1 5
/ \ / \ / \ 
1 4 0 2 4 6
```

5. (10 points) Draw a diagram which depicts the state of a hash table using separate chaining, after insertion of 3, 1, 8, 5, 2, 11, 6 into a table of size 5, using the hash function h(x) = x % 5.

```
index list

0 5
1 1 11 6
2 2
3 3 8
4
```
2 Coding problems

(a) (20 points) Complete `indexOf` method in the BST class that you have downloaded. The method should be written to return the number of items of data in the tree which occur “before” the parameter val, as determined by the `compareTo` method. For example, in a set containing "a", through "k", `set.indexOf("g")` should return 6.

```java
private int indexOf(Node x, T val) {
    if (x == null) return 0;
    int cmp = val.compareTo(x.val);
    // in this case, all the elements that are less than val are in the
    // left subtree, although it may be that not all of them are
    if (cmp <= 0)
        return indexOf(x.left,val);
    // in this case, all the elements in the left subtree are less than val,
    // and some that are in the right subtree may be less than val also.
    else return indexOf(x.left,val) + 1 + indexOf(x.right,val);
}
```

(b) (20 points) Complete the `Point` class that you have downloaded so that the program’s output is as follows:

i. The first line of output should be `true`.
ii. The next 2 lines should be `(1,2)` and `(3,4), or (3,4) and (1,2). The ordering of these 2 lines is not important.

```java
public int hashCode() {
    return ((Integer) (x+y)).hashCode();
}
```

// since equals is inherited from the Object class, it
// must be overridden so that it takes an Object as a parameter
public boolean equals(Object o) {
    Point p = (Point) o;
    return x == p.x && y == p.y;
}

Explanation: Objects that are stored in a HashSet or HashMap must have `hashCode` and `equals` methods. The `hashCode` method determines which “bucket” of the table an item belongs in. The `equals` method is used to compare data items. In a HashSet, the `contains` and `add` methods use the data’s `equals` method.