Lab Tasks 7

No submission! Use this lab to check your understanding of binary trees.

For this lab task, the class java.util.Random is useful. You can generate random numbers as follows:

```
Random r = new java.util.Random();
int someRandomNumber = r.nextInt();
```

The object r creates a sequence of random numbers. Make sure you only use a single instance of the class Random in your program.

Consider the implementation BinarySearchTree, given in the module homepage.

1. One way to measure the imbalance in a binary tree is to subtract the number of right edges from the number of left edges. If the result is positive, the tree is imbalanced to the left, if it is negative, the tree is imbalanced to the right. Add a function

   ```
   public int imbalance()
   ```

   which returns the imbalance of the tree.

2. Verify experimentally that repeated random insertions and removals from a binary search tree leads to an imbalanced tree (Figure 4.27, p 122).

3. One source of the imbalance is the asymmetry of the remove(...) function. Remove the asymmetry by alternating between the smallest child in the right sub-tree and the largest child in the left sub-tree. Repeat your experiment. Has the imbalance decreased?

4. Another strategy to avoid imbalance is to randomly choose between the smallest child in the right sub-tree and the largest child in the left sub-tree. Implement this behavior and repeat your experiment.