

Lab 12

Goal: To give you experience with implementing splay tree operations.

All the code is in the dict package. You can compile it from your lab12 directory with "javac -g dict/*.java". Test code is provided and can be run with "java dict.SplayTree".

The fields and methods of the SplayTree class are the same as the BinaryTree class you modified in Lab 11; it even uses the same BinaryTreeNodes and implements the same Dictionary interface.

We have provided implementations of the find() and insert() methods. (We have omitted the remove() method.) These implementations are similar to those in an ordinary binary search tree, but they always end by splaying a node to the root. However, the splay operation splayNode() does not work correctly, because it does not use the zig-zig operation. Instead, it splays a node to the root by doing a sequence of zig operations. Unfortunately, this improper splaying operation does not rebalance an extremely unbalanced splay tree.

Your job is to write a method zigZig() that implements the zig-zig step, then fix splayNode() so that it uses the zig, zig-zag, and zig-zig steps at the correct times. After you finish each part, use the test code to check your progress.

Part I: Implement zigZig() (2 points)

We have provided implementations of tree rotations in the methods rotateLeft() and rotateRight(). We have also provided implementations of the zig and zig-zag steps in the methods zig() and zigZag(); we suggest you examine those methods. Then fill in the body of a method zigZig() that implements the zig-zig step. Refer to the Lecture 36 notes if you need help remembering the difference. Make sure that the test for Part I runs without printing any error messages; it should say "Successfully completed Part I".

We suggest that you don't change splayNode() until you have Part I working, because changes to splayNode() might break our tests that tell you whether you have implemented Part I successfully.

Part II: Fix splayNode() (2 points) -----
The method splayNode() currently uses only zig(). Fix the implementation so that it chooses appropriately between zig(), zigZag(), and zigZig() at each iteration of the loop that splays the node to the root of the tree. Again, refer to the Lecture 36 notes for details.

The tests for Part II include a test that shows how some trees cannot be balanced without zig-zig steps.

Check-off

Show the code you have written, and run the test program. You'll receive 2 points for each part that runs without printing any error messages.