My program consists of two tree class that use different types of nodes. The AVL tree nodes contain information about their height, and a reference to its left and right children. The Splay tree nodes only contain their key, but also have a reference to their parents.

The AVL tree has a balance checking method that makes sure that the height of the tree does not exceed O(log n). The balance is checked with every insertion and deletion of a node. The Splay tree on the other hand is efficient through rotating the tree around whenever an element is accessed. The operations on the Splay tree should take at most O(n log n) time.

I expected the Splay Tree to be faster on average which was confirmed by my test outputs below, but my way of counting "comparisons" is probably wrong, as AVL trees should be faster for smaller elements.

I was able to implement the bracket notation for tree output for the Splay tree but not the AVL tree, as I could not figure out a way to do this without somehow referencing a node's parent, which I did incorporate into my design of AVL tree.

The program is written in Java.

Compile:

javac AVLTree.java

javac SplayTree.java

javac Main.java

Execute:

java Main "filename.txt"

Example outputs:

p2_test0

```
Splay tree comparisons: 526
AVL tree comparisons: 631
```

p2_test1

```
Splay tree comparisons: 5306
AVL tree comparisons: 6966
```

p2_test2

```
Splay tree comparisons: 18839
AVL tree comparisons: 21531
```

p2_test3

```
Splay tree comparisons: 20181
AVL tree comparisons: 21560
```

p2_test4

```
Splay tree comparisons: 19022
AVL tree comparisons: 21511
```