Chapter 5, Binary search trees:
  ▶ Binary search trees; the balanced BST problem (Today)
  ▶ AVL trees (Monday after break)
  ▶ Traditional red-black trees (Wednesday after break)
  ▶ Left-leaning red-black trees (Friday after break)
  ▶ “Wrap-up” BST (the following Monday)

Today:
  ▶ The quest for a better map, motivation for BST
  ▶ BST definition and iterative implementation
  ▶ BST performance and the balanced BST problem
  ▶ Introduction to the code base
public interface Map<K, V> extends Iterable<K> {
    void put(K key, V val);
    V get(K key);
    boolean containsKey(K key);
    void remove(K key);
}

List $\Theta(n)$
BST $\Theta(\lg n)$
Hashtable $\Theta(1)$
<table>
<thead>
<tr>
<th></th>
<th>Unsorted</th>
<th>Sorted</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find</strong></td>
<td>$\Theta(n)$</td>
<td>$\Theta(lg n)$</td>
</tr>
<tr>
<td><strong>Array</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert</td>
<td>$\Theta(1)$ expected, $\Theta(n)$ worst</td>
<td>$\Theta(n)$</td>
</tr>
<tr>
<td>Delete</td>
<td>$\Theta(n)$</td>
<td>$\Theta(n)$</td>
</tr>
<tr>
<td><strong>Linked structure</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
</tr>
<tr>
<td>Delete</td>
<td>$\Theta(1)$</td>
<td>$\Theta(1)$</td>
</tr>
</tbody>
</table>
6, 0, 5, 1, 4, 2, 3

0, 3, 5, 2, 6, 1, 4

4, 2, 5, 3, 0, 1, 6

1, 6, 5, 2, 4, 3, 0

1, 2, 5, 4, 3, 0, 6

3, 1, 5, 0, 2, 4, 6

6, 0, 5, 1, 4, 2, 3

0, 3, 5, 2, 6, 1, 4

4, 2, 5, 3, 0, 1, 6

1, 6, 5, 2, 4, 3, 0

1, 2, 5, 4, 3, 0, 6

3, 1, 5, 0, 2, 4, 6

height 7
total depth 21
ANI 4

height 4
total depth 14
ANI 3

height 4
total depth 11
ANI 2.57

height 4
total depth 11
ANI 2.57

height 5
total depth 14
ANI 3

height 3
total depth 10
ANI 2.43

height 6
total depth 16
ANI 3.29

height 5
total depth 14
ANI 3

height 3
total depth 10
ANI 2.43

height 7
total depth 21
ANI 4

height 4
total depth 14
ANI 3

height 4
total depth 11
ANI 2.57

height 6
total depth 16
ANI 3.29

height 5
total depth 14
ANI 3

height 3
total depth 10
ANI 2.43

height 7
total depth 21
ANI 4

height 4
total depth 14
ANI 3

height 4
total depth 11
ANI 2.57

height 6
total depth 16
ANI 3.29

height 5
total depth 14
ANI 3

height 3
total depth 10
ANI 2.43

height 7
total depth 21
ANI 4

height 4
total depth 14
ANI 3

height 4
total depth 11
ANI 2.57

height 6
total depth 16
ANI 3.29

height 5
total depth 14
ANI 3

height 3
total depth 10
ANI 2.43
Map

BSTMap<K,V,I>

Node

<<interface>>
implements
RealNode
key: K
value: V
left, right: Node
info: I

recompute()
putFixup(Node): Node
removeFixup(Node): Node
rootFixup(Node)
newInfo(Node): I

Balancer<K,V,I>

<<interface>>

putFixup(Node): Node
removeFixup(Node): Node
rootFixup(Node)
newInfo(Node): I

RealNode

key: K
value: V
left, right: Node
info: I

NodeInfo

<<interface>>
recompute()

Balancer

AVLInfo

AVLBalancer

AVLBalancer

 RBInfo

TradRBBalancer

LLRBBalancer
Coming up:

Catch up on older projects?

*Due Mon, Mar 14 (class time)*
*Read Section 5.(1 & 2)*
*Do Exercises 5.(2 & 6)*
*Take quiz*

*Do BST rotations project (suggested by Tuesday, Mar 15)*

*Due sometime in the week after break*
*Read Section 5.3*
*Do Exercises 5.(7 & 8)*
*Take quiz*