Chapter 5, Binary search trees:

- Binary search trees intro; the balanced BST problem (**Today** and next week Friday)
- AVL trees (next week Friday and Monday, Oct 24)
- Traditional red-black trees (Wednesday, Oct 26)
- Left-leaning red-black trees (Friday, Oct 28)
- “Wrap-up” BST (Monday, Oct 31)

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
Coming up:

* Catch up on older projects?

* **Due Fri, Oct 21 (class time)**
  * Read Section 5.(1 & 2)
  * Do Exercises 5.(2 & 6)
  * Take quiz

* Do **BST rotations project (suggested by Mon, Oct 24)**

* **Due Tues, Oct 25 (end of day)**
  * Read Section 5.3
  * Do Exercises 5.(7 & 8)
  * Take quiz
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>Find</strong></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>
Coming up:

Catch up on older projects?

*Due Fri, Oct 21 (class time)*
Read Section 5.(1 & 2)
Do Exercises 5.(2 & 6)
Take quiz

Do BST rotations project (suggested by Mon, Oct 24)

*Due Tues, Oct 25 (end of day)*
Read Section 5.3
Do Exercises 5.(7 & 8)
Take quiz