

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

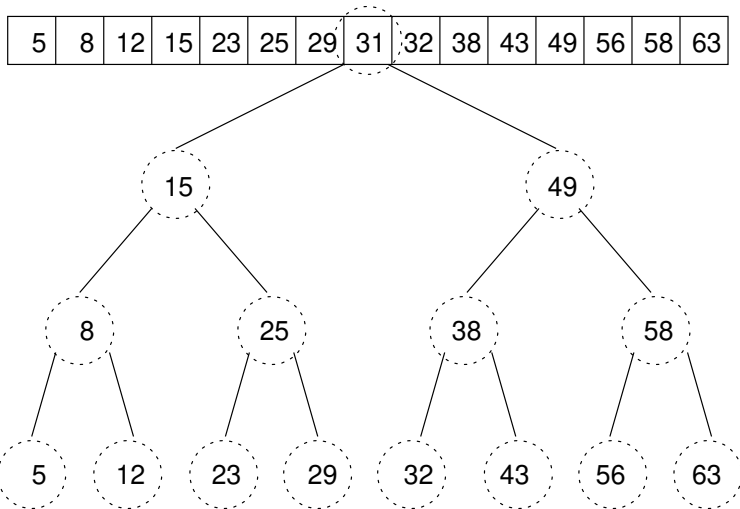
Do Exercises 5.(7 & 8)

Take quiz

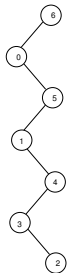
```
public interface Map<K, V> extends Iterable<K> {  
    void put(K key, V val);  
    V get(K key);  
    boolean containsKey(K key);  
    void remove(K key);  
}
```

List	$\Theta(n)$
BST	$\Theta(\lg n)$
Hashtable	$\Theta(1)$

		Unsorted	Sorted
Array	Find	$\Theta(n)$	$\Theta(\lg n)$
	Insert	$\Theta(1)$ expected, $\Theta(n)$ worst	$\Theta(n)$
	Delete	$\Theta(n)$	$\Theta(n)$
Linked structure	Find	$\Theta(n)$	$\Theta(n)$
	Insert	$\Theta(1)$	$\Theta(1)$
	Delete	$\Theta(1)$	$\Theta(1)$

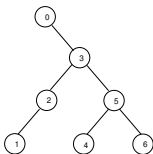


6, 0, 5, 1, 4, 2, 3



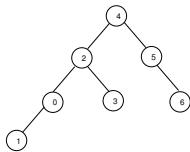
height 7
total depth 21
ANI 4

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



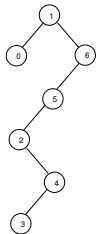
height 4
total depth 14
ANI 3

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



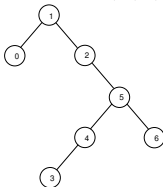
height 4
total depth 11
ANI 2.57

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



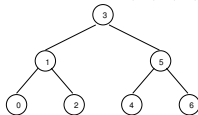
height 6
total depth 16
ANI 3.29

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

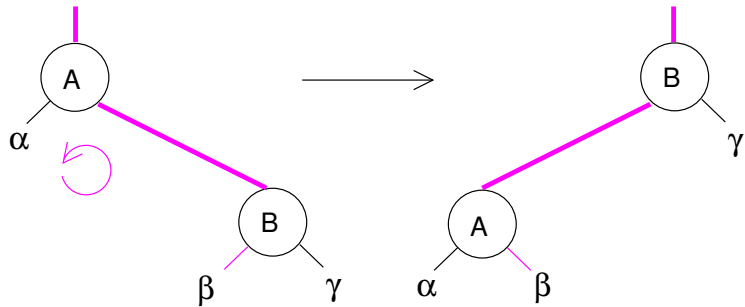


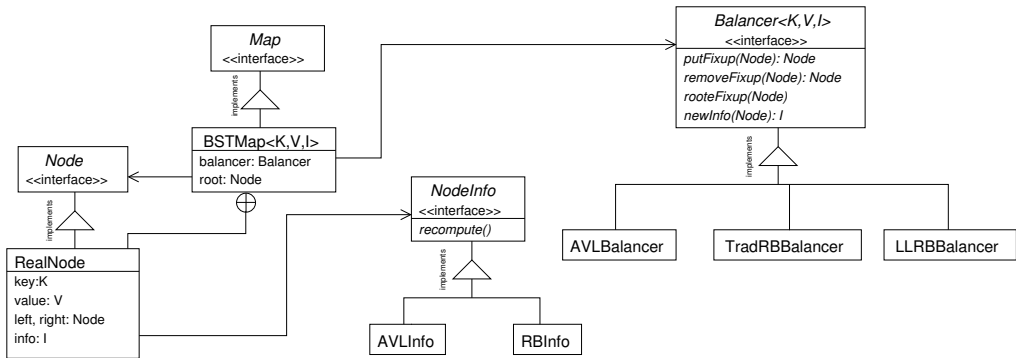
height 5
total depth 14
ANI 3

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



height 3
total depth 10
ANI 2.43





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