

PA1 due Tuesday
PA3 Late/Resubmit due Tuesday

Binary Search Tree (BST)

implements Map<K,V>

```

class BST<K, V> {
    Node<K, V> root;
    BST() {this.root = null;}
    BST(Node<K, V> root) { this.root = root; }

    private V get(Node<K, V> node, K key) {
        if (node == null) { //throw error } } not found
        if (node.key.equals(key)) {
            return node.value;
        }
        if (node.key < key) {
            return get(node.left, key);
        }
        else {
            return get(node.right, key);
        }
    }

    public V get(Key key) {
        return this.get(root, key);
    }
}

```

base case
recursive case

What error should we throw in get() if the key isn't found?

Where is the get() method broken?

> does not work

How can we fix the get() method to work with Objects?

Interface

↳ Comparator / also on Object
↳ pass to the constructor
save as a field

Comparable

↳ compareTo()
 < 0 less than
 0 equal
 > 0 greater than

↳ public String implements Comparable

What error should we throw in get() if the key isn't found?

NoSuchElementException / ElementNotFoundException

What would the code that uses get() look like to prevent the program crashing if the key is missing?

BST tree = ...
try {

tree.get(?);

{
 catch (NoSuchElementException e) {
 // what do we do?
 // print error?
 }
 catch (Exception e) {
 //
 }
}

boolean find(E toFind; E value)

Comparable comp = (Comparable) toFind;

while loop

if (comp.compareTo(value) == 0) {
 return true
}

return false;

① < E extends Comparable?

class Test < E extends Comparable? {

boolean find(E toFind)

if (toFind.compareTo(...)) {
 return true
}

return false;

Assume the key and value are identical for this example:

Trace the path for get(4)

How many nodes does it touch?

4 nodes

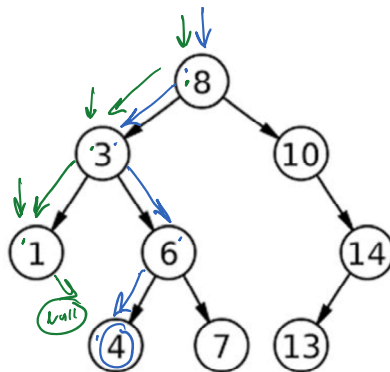
Trace the path for get(2)

How many nodes does it touch?

3 nodes

What happens when the node isn't found?

throws exception



Value smaller node
 → go left
 Value greater node
 → go right
 Value equals node
 → found it

3

Assume the key and value are identical for this example:

Trace the path for get(4)
How many nodes does it touch?

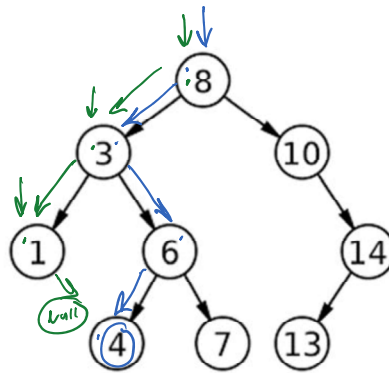
4 nodes

Trace the path for get(2)
How many nodes does it touch?

3 nodes

What happens when the node isn't found?

throws exception



Value smaller node
→ go left
Value greater node
→ go right
Value equals node
→ find it

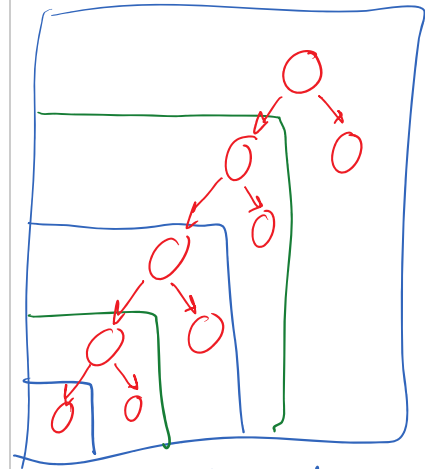
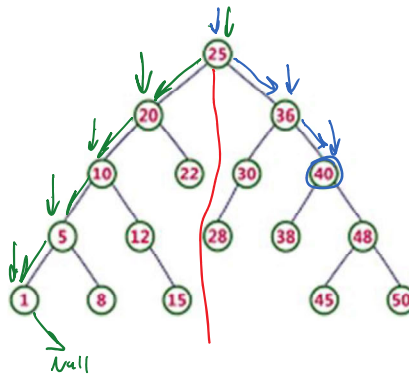
Assume the key and value are identical for this example:

Trace the path for get(40)
How many nodes does it touch?

3 nodes

Trace the path for get(4)
How many nodes does it touch?

5 nodes



recursive data structure

