

CSE12 - Lecture 18

Wednesday, May 8, 2024 10:00 AM

PA6 due Wednesday @ 8am

PA4 Late/Resubmit due Wednesday @ 8am

Exam 2 - May 15th

- <https://ucsd-cse12-sp24.github.io/lectures/exam2.html>

Lecture 18

Binary Search Tree (BST)

```

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 }
        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);
    }
}

```

Where is the get() method broken?

→ does not work .equals()? → more appropriate

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

Interface

→ Comparator / also an Object

→ passes to the constructor same as a field

→ Comparable

↳ compareTo()

↳ less than 0

↳ equal 0

↳ greater than 0

```

public String implements Comparable<String> {
    ...
}

```

boolean find(E toFind, E key)

→ Comparable <E> comp = (Comparable <E>) toFind;

comp.compareTo(key)

↳ E extends Comparable?

class Test <E extends Comparable> {

boolean find(E toFind) {

if (toFind.compareTo(...) < 0)


}

↳ runtime error bad!

```

class Node<K, V> {
    K key;
    V value;
    Node<K, V> left;
    Node<K, V> right;
    public Node(K key, V value,
                Node<K, V> left,
                Node<K, V> right) {
        this.key = key;
        this.value = value;
        this.left = left;
        this.right = right;
    }
}

```



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

No Such Element Exception() / Element Not Found Exception()

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

```

Bst< > true = ...
try {
    true.get(x);
}
catch (Element Not Found Exception e) {
    // what do we do?
    // print error message
}
catch (Exception e) {
    // other errors
}

```

↳ boolean true.contains()

Name: _____ PID: _____ Code: 9971

Assume the key and value are identical for this example:

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

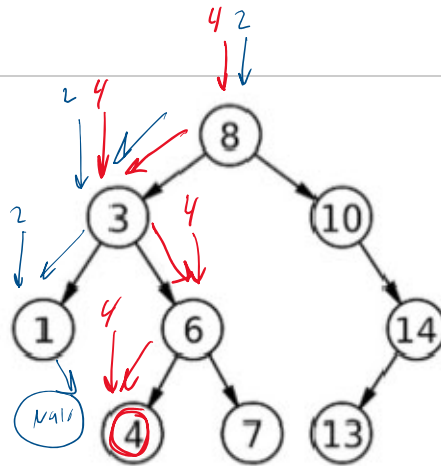
4 nodes

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

3 nodes, 4 comparisons

What happens when the node isn't found?

throws exception



Key smaller node.key?
→ go left

Key greater node.key?
→ go right

Key equal node.key?
→ found it
→ return value

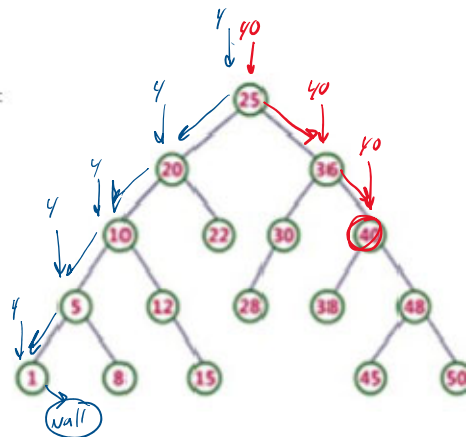
Assume the key and value are identical for this example:

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

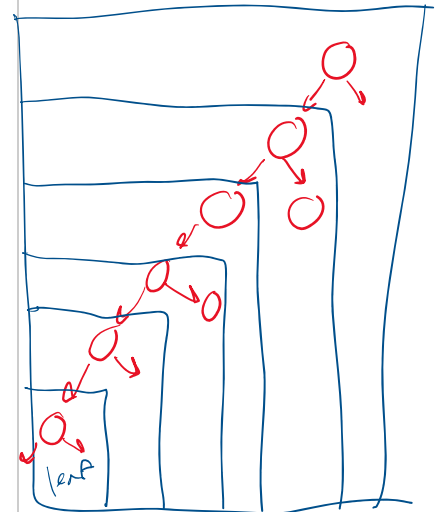
3 nodes

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

5 nodes, 6 comparisons



BST → recursive data structure



binary search (sorted)

