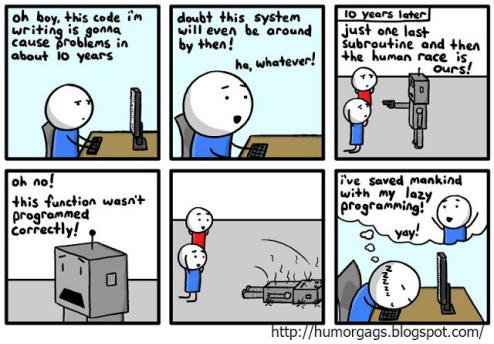
CSE 143

Computer Programming II

More BSTs



http://humorgags.blogspot.com/

Outline

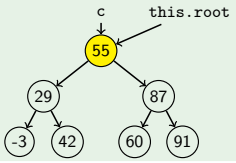
Adding to a BST 1

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) { // 55 > 49
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) {
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (tree.add(49))

value = 49



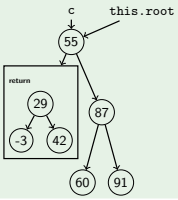
Adding to a BST 2

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) {
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (tree.add(49))

value = 49



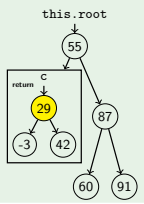
Adding to a BST 3

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) { // 29 < 49
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (tree.add(49))

value = 49



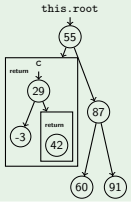
Adding to a BST 4

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) { // 29 < 49
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (tree.add(49))

value = 49



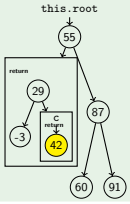
Adding to a BST 5

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) { // 42 < 49
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (tree.add(49))

value = 49



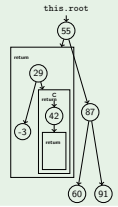
Adding to a BST 6

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) { // 42 < 49
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (tree.add(49))

value = 49



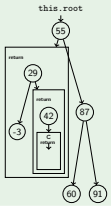
Adding to a BST 7

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) {
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (tree.add(49))

value = 49



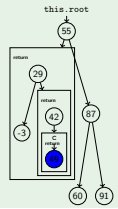
Adding to a BST 8

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) {
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (tree.add(49))

value = 49



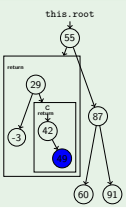
Adding to a BST 9

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) {
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (tree.add(49))

value = 49



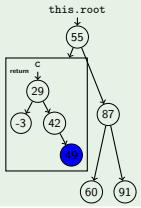
Adding to a BST

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) {
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (`tree.add(49)`)

value = 49



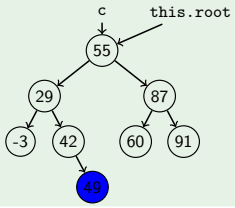
Adding to a BST

Code

```
1  private IntTreeNode add(IntTreeNode c, int value) {
2     if (c == null) {
3        c = new IntTreeNode(value);
4     }
5     else if (c.data > value) {
6        c.left = add(c.left, value);
7     }
8     else if (c.data < value) {
9        c.right = add(c.right, value);
10    }
11    return c;
12 }
```

Example (`tree.add(49)`)

value = 49



first

first

Write a function `first` in the BST class with the following signature:

```
public int first();
```

that returns the smallest value in the tree. If the tree is empty, `first` should throw a `NoSuchElementException`.

```
1  public int first() {
2     return first(this.root);
3  }
4
5  private int first(IntTreeNode current) {
6     if (current == null) {
7        throw new NoSuchElementException();
8     }
9     /* Keep on going left as far as we can */
10    else if (current.left != null) {
11       return first(current.left);
12    }
13    else {
14       return current.data;
15    }
16 }
```

increment

increment

Write a function `increment` in the BST class with the following signature:

```
public void increment(int value);
```

that increments every value visited on a standard "search" for `value`.

replaceWithSum

replaceWithSum

Write a function `replaceWithSum` in the BST class with the following signature:

```
public void replaceWithSum(int value);
```

that replaces all subtrees "rooted" at `value` with their sum.

copy

copy

Write a function `copy` in the BST class with the following signature:

```
public IntTree copy();
```

that returns a new `IntTree` with the same values as `this` one.