Trees 4
Equality

AVL trees

insert	1000
insert	500
insert	750
insert	250
insert	375
insert	625
insert	562
insert	590
insert	615
insert	608
insert	1000
insert	850

Tree Equality

- When should trees (BST) be considered equal?
  - Same Contents
  - Same Structure
  - Same Structure with the same contents
    - can trees have same contents & not same structure?
  - Mirror images?

Structurally Identical

```
public <G extends Comparable<G>> boolean isStructurallyIdentical(LinkedBinaryTree<G> otherTree) {
        return isSIUtil(root, (Node<G>)otherTree.root);
    }
```

- "<G extends Comparable<G>>" ???!!
- idea, traverse both trees at the same time
- base case(s)?

Same Structure and Content

```
protected class Node<F extends Comparable<F>> implements Comparable<Node<F>>{
   F payload;
   Node<F> right;
   Node<F> left;
   public Node(F e) {
        payload = e;
        right = null;
        left = null;
    }


    @Override
    public int compareTo(LinkedBinaryTree<E>.Node<F> o) {
        return this.payload.compareTo(o.payload);
    }
}
```

Same Structure and Content

- What needs to be added to sameStructure?

Mirror Structure

- Changes to same structure???

Same Contents

- Problem: trees with the same contents can have different structures!
- Naive algorithm
  - 1. Ensure trees have same number of nodes
  - 2. Go through tree1.
    At each node in tree1
    - Ask does tree2 contain the same data item
  - If you ever get a NO stop and return false

- Time complexity??

Same Contents -- improved

- Check that the trees have the same number of items
- a1 = ArrayList from tree1 that contains the items in tree 1 in sorted order
- a2 = ArrayList from tree1 that contains the items in tree 1 in sorted order
- Compare items in a1 and a2

- Does this work?
- Time Complexity?