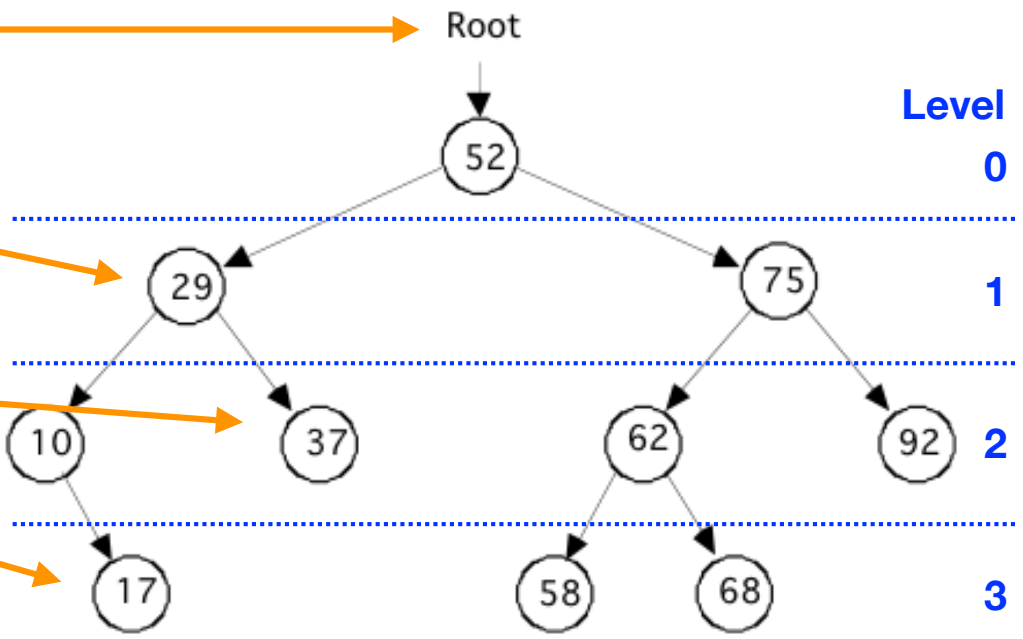
Binary Trees

David Greenstein
Monta Vista High School

Binary Trees

VOCABULARY

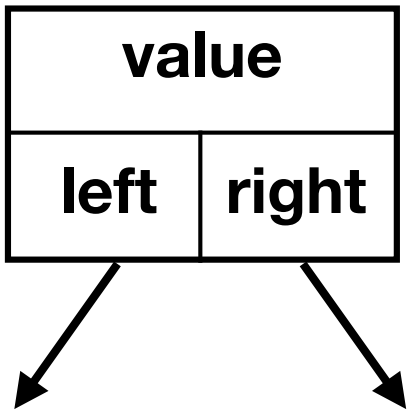
- **Root** - the top of the tree
- **Parent Node** - a node that points to one or more nodes below
- **Child Node** - a node pointed to by a parent node
- **Leaf** - a node with no children
- **Level** - the distance from the root node (level 0)
- **Edge** - a connector between a parent and child node (arrow connector)



TreeNode

```
public class TreeNode<E extends Comparable<E>>
{
  private E value;
  private TreeNode<E> left;
  private TreeNode<E> right;

  // constructors
  public TreeNode(E myValue, TreeNode<E> myLeft, TreeNode<E> myRight)
  {
    value = myValue; left = myLeft; right = myRight;
  }
  public TreeNode(E myValue) {
    this(myValue, null, null);
  }
  // accessors and modifiers
  . . .
}
```



Building Binary Tree

Draw a tree

- Values: 42 13 35 97 64 43 81 74 26 38

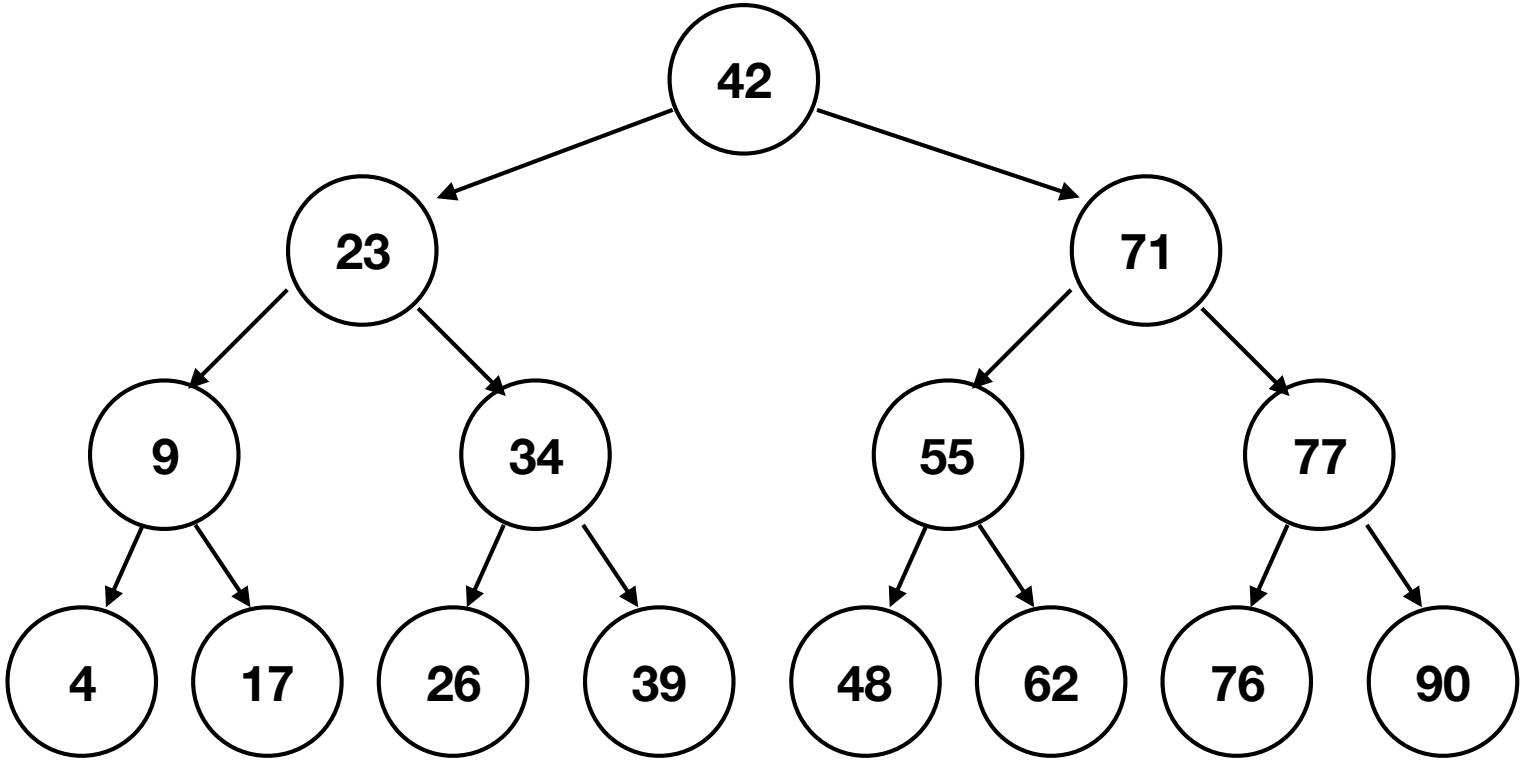
Building Binary Tree

The performance of the binary tree depends on the initial order of the data set.

- Try: 1 2 3 4 5 6 7

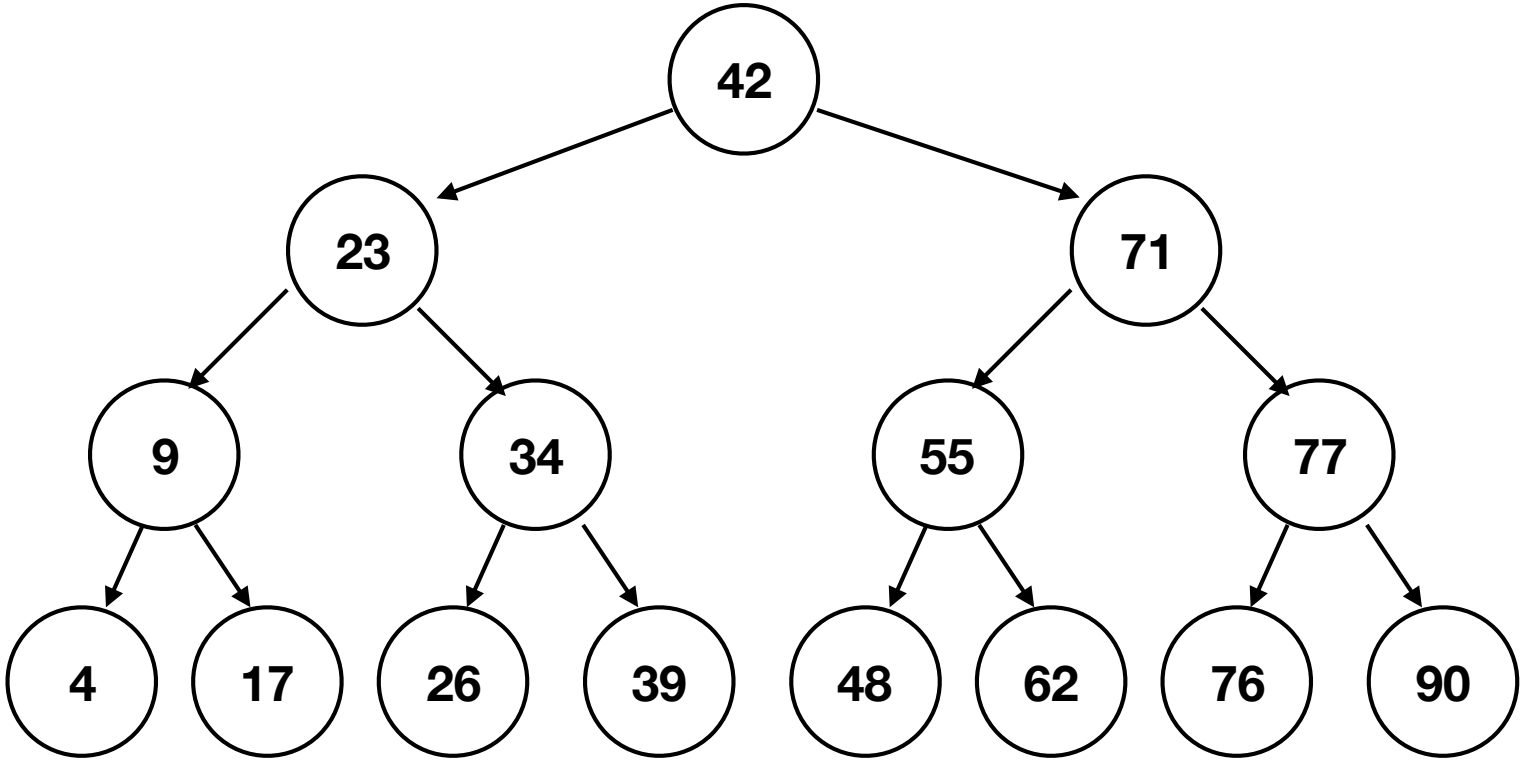
Balanced Binary Tree

- **Balanced Tree** - one in which the left and right sides of all subtrees have the same number of nodes.
- **Random order data** sets are best for balanced tree.



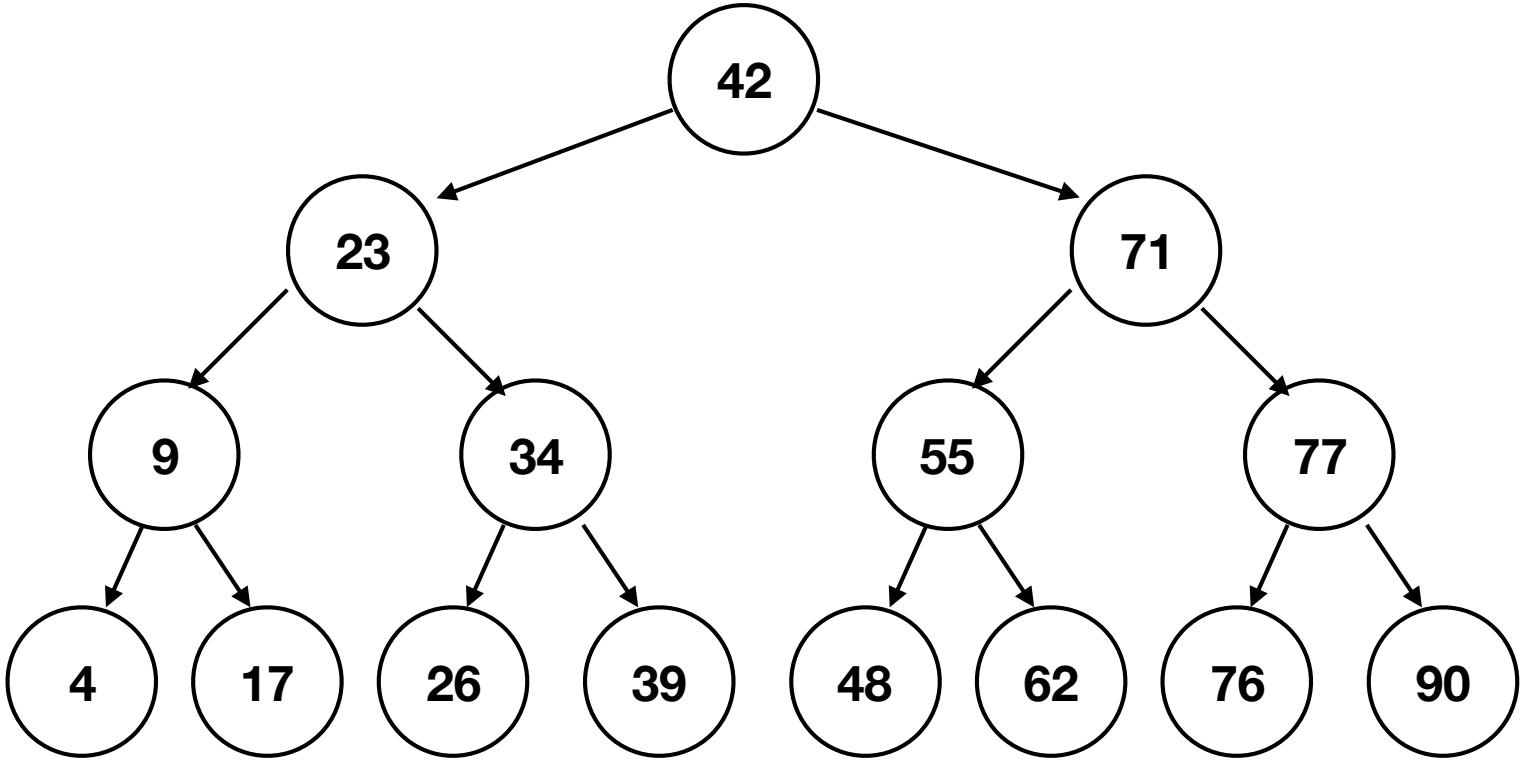
Tree Traversal: Inorder

1. Visit **left subtree**
2. Visit **node**
3. Visit **right subtree**



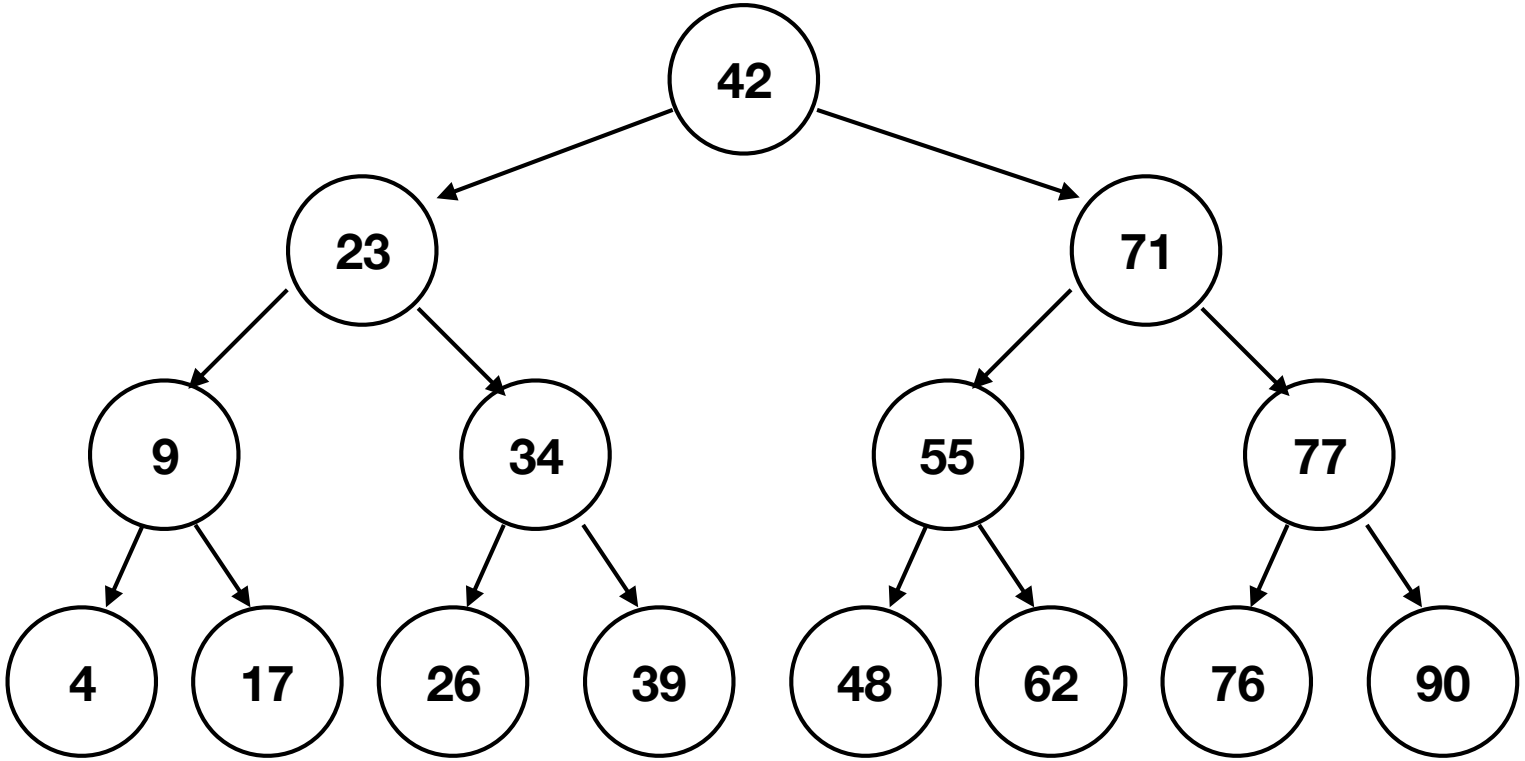
Tree Traversal: Preorder

1. Visit **node**
2. Visit **left subtree**
3. Visit **right subtree**



Tree Traversal: Postorder

1. Visit **left subtree**
2. Visit **right subtree**
3. Visit **node**



Building Binary Tree

Draw a tree

- Values:
  20 34 40 28 6 18 39 3 15 12 48 24 19 27 22

Building Binary Tree

Draw a tree

- Values:
  21 13 33 8 22 20 6 17 43 37 18 19 12 23 1

Building a Balanced Tree

How do you make a balanced tree?

Using which order of a unbalanced tree?

1 6 8 12 13 17 18 19 20 21 22 23 33 37 43