Binary Trees

A binary tree has the following fundamental qualities:

- Each node has at most 2 children
- Every left node should be less than its parent and every right node should be greater than its parent.

Traversal of Trees

Breadth First

This is printing every node of certain level before moving onto the next level.

```
public void breadthFirst()
{
    BSTNode<T> p = root;
    Queue<BSDNode<T>> queue = new Queue<BSTNode<T>>();
    if (q != null)
    {
        queue.enqueue(p);
        while (!queue.isEmpty())
        {
            p = queue.dequeue();
            visit(p);     //any processing we wish to do
            if (p.left != null)
                queue.enqueue(p.left);
            if (p.right != null)
                queue.enqueue(p.right);
        }
    }
}
```

Depth First

Traversal goes as far as possible one way, until an end is found, and then backtacks to go down another path.

Pre-Order

```
protected void preorder(BSTNode<T> p)
{
    if (p != null)
    {
        visit(p);
        preorder(p.left);
        preorder(p.right);
    }
}
```

In-order

```
protected void inorder(BSTNode<T> p)
{
    if (p != null)
    {
        inorder(p.left);
        visit(p);
        inoder(p.right);
    }
}
```

Post-order

```
protected void postoder(BSTNode<T> p)
{
    if (p != null)
    {
        postorder(p.left);
        postorder(p.right);
        visit(p);
    }
}
```

Stack-Less Depth-First

Threaded Trees.

These are trees where the null right children point to the immediate node above them to the right.