

CS 6301.002. Implementation of advanced data structures and algorithms
Short Project 4 (Trees)

Group 10 : Gaurav Ketkar, Vijay Mungara, Malav Shah, Madhuri Abnave

a. Level order

Write the following method for the BST class:

```
// Return an array with the elements using a level order traversal of the tree
public Comparable[] levelOrderTraversal(){
    Comparable[] result = new Comparable[size];
    Queue<Entry> nodeQ = new LinkedList<Entry>();
    nodeQ.add(root);
    int i = 0;
    Entry<T> removedNode;
    while(!nodeQ.isEmpty()){
        removedNode = nodeQ.remove();
        if(removedNode.left != null) nodeQ.add(removedNode.left);
        if(removedNode.right != null) nodeQ.add(removedNode.right);
        result[i++] = (Comparable) removedNode.element;
    }
    return result;
}
```

c. Modifying remove in BST class

Rewriting remove() so that it alternates between two possibilities:

```
static int rightOrLeft = 0;
// remove node that has two children,
// from either right or left depending on int rightOrLeft
void removeTwo(Entry<T> node) {
    if (0 == rightOrLeft){
        remove the minimum from the right side
        and then,
        rightOrLeft = 1;
    }
    else if(1 == rightOrLeft)
    {
        remove the maximum from the left side
        and then,
        rightOrLeft = 0;
    }
}
```