Lab 3

Name: Srivenkatesh Nair
Student ID: 378000329
Username: sn6711
ISTE.612

Task 1: Binary Tree Index Construction and Searching

1)

```java
//This method inserts a new node into the existing binary tree
public void add(TreeNode node, TreeNode iNode) {
    if(node == null) {
        node = iNode;
        return;
    }
    if(iNode.data.getTerm().compareTo(node.data.getTerm()) < 0) {
        if(node.left == null) {
            node.left = iNode;
        }
        else{
            add(node.left, iNode);
        }
    }
    else if(iNode.data.getTerm().compareTo(node.data.getTerm()) > 0) {
        if(node.right == null) {
            node.right = iNode;
        }
        else{
            add(node.right, iNode);
        }
    }
}
```

```java
//This method searches a node in the existing binary tree with the term similar as key
public TreeNode search(TreeNode n, String key) {
    if(n == null || n.data.getTerm().equals(key)) {
        return n;
    }

    if(key.compareTo(n.data.getTerm()) < 0) {
        return search(n.left, key);
    }
    else{
        return search(n.right, key);
    }
}
```

2)

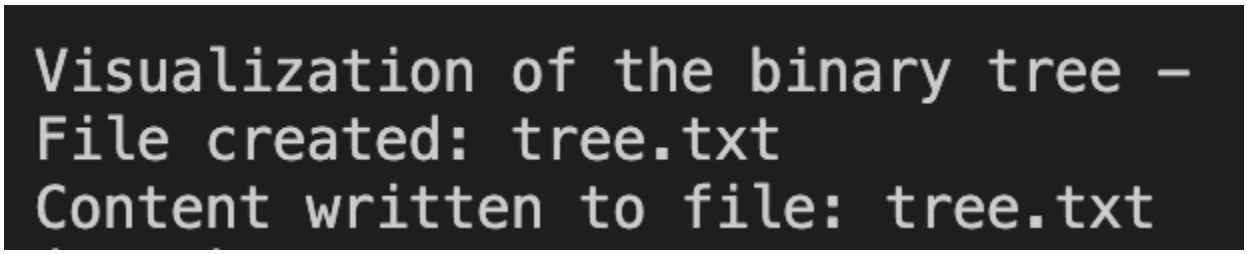
```
//Constructor
@SuppressWarnings("removal")
public BTreeIndex(String[] docs){
    //I have hard coded this list of terms which are not terms that belong to english language
    String[] incorrectToken = {"'", "'90s", "'r'", "'s", "10", "1960's", "1990s", "1997's",
                               "2", "20", "20th", "3", "4", "7", "8", "9", "y2k"};

    //Parsing the documents
    ArrayList<ArrayList<String>> docLists = new ArrayList<ArrayList<String>>();
    termList = new ArrayList<String>();
    docIds = new ArrayList<ArrayList<Integer>>();
    for(int id = 0; id < docs.length; id++){
        String[] tokens = parse("Lab1_Data/" + docs[id]);
        ArrayList<String> subList = new ArrayList<String>();
        for(String word: tokens){
            subList.add(word);
        }
        docLists.add(subList);
    }
```

3)

```
Search Query 1:wisdom
Result:
Term: wisdom  Document IDs: [2]

Search Query 2:plot
Result:
Term: plot  Document IDs: [0, 2, 4]

Search Query 3: strange thing
Term: strange  Document IDs: [0]
Term: thing  Document IDs: [1, 2, 3]
Result:
No Documents!

Search Query 4: film review
Term: film  Document IDs: [0, 2, 3, 4]
Term: review  Document IDs: [0, 1]
Result:
Document IDs: [0]

Search Query 5: a good start
Term: a  Document IDs: [0, 1, 2, 3, 4]
Term: good  Document IDs: [0, 1, 3, 4]
Term: start  Document IDs: [0, 1]
Result:
Document IDs: [0, 1]

Search Query 6: american thrilling chase
Term: american  Document IDs: [0, 2]
Term: thrilling  Document IDs: [0]
Term: chase  Document IDs: [0, 1]
Result:
Document IDs: [0]
```

Task 2: Binary Tree Visualization

Terminal Output:



tree.txt:

```
Level 0 (Root Node) :
Term: living  Postings: [4]

Level 1 :
-------Left Subtree------
Term: dust  Postings: [4]
-------Right Subtree------
Term: she  Postings: [3, 4]

Level 2 :
-------Left Subtree------
Term: cast  Postings: [2, 3, 4]
Term: guess  Postings: [0, 4]
-------Right Subtree------
Term: parts  Postings: [1]
Term: thing  Postings: [1, 2, 3]

Level 3 :
-------Left Subtree------
Term: bad  Postings: [0, 3, 4]
Term: critique  Postings: [0]
Term: fed  Postings: [0, 2]
Term: information  Postings: [2]
-------Right Subtree------
Term: narrator  Postings: [4]
Term: realized  Postings: [3]
Term: states  Postings: [4]
Term: virus  Postings: [1]

Level 4 :
-------Left Subtree------
Term: animated  Postings: [3]
Term: booby  Postings: [3]
Term: colorful  Postings: [3]
Term: didn't  Postings: [0]
```