R·I·T	**Rochester Institute of Technology** **Golisano College of Computing and Information Sciences** **School of Information**

Lab 3
Binary Tree Indexing, Search, and Visualization

Overview

This lab consists of two tasks:

- Building a binary tree to index the term dictionary and using it to answer single term queries and conjunctive queries using the documents from Lab 1.
- Visualizing various aspects of the binary tree.

Resources

- You should have read Chapters 2 and 3 of Introduction to Information Retrieval.
- Carefully read the lecture notes and code from Week 5 and understand the technical details.

Task 1: Binary Tree Index Construction and Searching (70 points)

In this task, you need to construct a binary tree index and use it to answer basic queries.

1. Complete the following two methods in BTreeIndex.java that allow you to insert nodes into a tree and search a term from the tree index.

```
/**
 * insert a node to a subtree
 * @param node root node of a subtree
 * @param iNode the node to be inserted into the subtree
 */
public void add(Node node, Node iNode)
{
        //TO BE COMPLETED
}
/**
 * Search a term in a subtree
 * @param n root node of a subtree
 * @param key a query term
 * @return tree nodes with term that match the query term or null if
no match
 */
public Node search(Node n, String key)
{
        //TO BE COMPLETED
}
```

2. Complete the constructor of BTreeIndex.java:

```
    /**
      * Construct binary search tree to store the term dictionary
      * @param docs List of input strings
      *
      */
     public BTreeIndex(String[] docs)
     {
        //TO BE COMPLETED
     }
```

3. Add test cases into the main method of BTreeIndex.java to test the binary tree index to answer single term and conjunctive (AND) queries. Show two test cases each for a single term query, a two-term conjunctive query, and a three-term conjunctive query.

Task 2: Binary Tree Visualization (30 points)

Add a visualizeTree method to BTreeIndex.java that outputs a level-based visualization of the binary tree for the first 4 levels of your tree to a file called tree.txt. The visualization should indicate the parent-child relationship between the nodes in the tree. Be creative!

Lab Submission Instructions – Create a PDF document showing all of your test cases for Task 1 and the resultant output. Your output should also show the postings for each term involved in a given query so it's obvious that your output is correct. Create a zip file called Lab3.zip that includes the PDF document with your results, the Lab1_Data folder with all documents contained therein, your visualization for Task 2 stored in tree.txt, and your code and submit it to the Lab 3 dropbox prior to the due date. **Please do NOT use file paths that are specific to your machine. Your code should include a hard-coded path to the Lab1_Data directory, which will be in the current directory where the code resides. Failure to do this will result in significant point loss.**