CSCI 36200: Data StructuresProgramming Assignment 2Instructor: Dr. Snehasis Mukhopadhyay

Due date: March 20, 2018

# Project Description

The program builds a cross-reference index by using Binary search tree (or Lexicographic Tree).

Each node of tree contains a word and a linked list of lines that store that word.

Each node in binary search tree is bigger (lexicographical comparison on key or word) than the left node and smaller than the right node. For example:

The linked list of lines is implemented in the LinkedList class. This class stores the LinkedListNode objects that link together to create singly linked list. Each LinkedListNode contains an integer value and a reference to next node. The values of nodes are in ascending order. For example, “fun” appears in line 4, 7, 8, 12

word =“fun”

lines: LinkedList

12

8

7

4

When a line number (integer number) is added to linked list, it iterates from head node to tail node to find the appropriate location to insert to maintain the correct order. If the number is found in the linked list, the algorithm ignores this number.

The Binary search tree uses recursive method to insert word and its line number into tree. It recursively chooses the left node or right node to insert into subtree. If the tree has node that contains the word, it adds the line number to the linked list of that node.

The output is result of the inorder traversal on tree.

# Other comments

FileInputStream. Retrieved from <https://docs.oracle.com/javase/7/docs/api/java/io/FileInputStream.html>

BufferedWriter. Retrieved from <https://docs.oracle.com/javase/7/docs/api/java/io/BufferedWriter.html>

Adamchik, V. S. (2009). Linked Lists. Retrieved from <https://www.cs.cmu.edu/~adamchik/15-121/lectures/Linked%20Lists/linked%20lists.html>

Data structure – binary search tree. Retrieved from <https://www.tutorialspoint.com/data_structures_algorithms/binary_search_tree.htm>