

**CSC 228 Data Structures and Algorithms, Spring 2019**  
**Instructor: Dr. Natarajan Meghanathan**

**Project 6: Binary Search Tree: Average Number of Comparisons for a Successful Search**

**Due:** April 3rd, 2019, 11.59 PM (submission through Canvas)

You are given the following code:

- (1) Code to construct a binary search tree using a sorted randomly generated array of integers
- (2) Code to determine the depth (level numbers) of the vertices in a binary tree

Add a member function `assignLevelNumbers()` to the Binary Search Tree class to determine the level numbers of the vertices in a binary search tree (start Breadth First Search with the `rootNodeID`, which is considered to be at level 0)

Add a member function `getLevelNumber(int nodeid)` that in turn calls the `getLevelNum()` on the BST node object corresponding to the 'nodeid' and returns the level number of node identified by its id.

In the main function, first call the `assignLevelNumbers()` function on the object of class Binary Search Tree before determining the number of comparisons for successful search.

Successful search: In class, we saw that the number of comparisons for the successful search of a key in a binary search tree (BST) is one more than the level number of the vertex representing the key in the BST. In the main function, write a for loop that will go through the individual vertices and extract their level numbers using the `getLevelNumber(int nodeid)` function called on an object of the class Binary Search Tree. Use these level numbers of the vertices to calculate the average number of comparisons for a successful search and print the same.

Test your code with the following values for the number of elements in the array.  
10, 100, 1000, 10000

The maximum value for an element in each case is 2500.

As part of the output, you need not print the contents of the array. Just print the average number of comparisons for a successful search for each of the above four values for the number of elements in the array.

**What to submit:** Include both (a) and (b) in a single word document and submit:

- (a) The complete code for the binary search tree class (including the implementation of the `assignLevelNumbers()` and `getLevelNumber(int)` functions) and the associated classes as well as the extended main function that determines the average number of comparisons for a successful search.
- (b) Tabulate the values for the number of elements and the average number of comparisons for a successful search. Why is that the average number of comparisons increases very slowly with increase in the number of elements?