

Issue Date: 21 Oct, 2021

Due date: 31 Oct, 2021

Instructions:

- Plagiarism is defined as "taking and using the thoughts, writings, and inventions of another person as one's own". IBA has a no-compromise policy on Plagiarism – in case, plagiarism is proved student will be given an **F grade**.
- Your assignment should represent your effort. However, you are not expected to work alone. It is fine to discuss the exercises and try to find solutions together, but each student shall write down and submit his/her solutions separately. It is a good academic standard to acknowledge collaborators, so if you worked together with other students, please list their names.
- Electronic Submission on LMS is compulsory.
- The assignment will be graded based on **timely submission on LMS**.

Deliverables:

Submit the JAVA project as a zip folder.

1. Write a function to print all values in the given range k1 to k2 stored in the binary search tree. Suppose two values k1 and k2 (where $k1 < k2$). A prototype of the function is given below

```
Public void PrintRange(node r, int k1, int k2) { ... }
```

Print all the keys of the tree in range k1 to k2. i.e. print all x such that $k1 \leq x \leq k2$ and x is a key of given BST. Print all the keys available in the given range in increasing order.

2. Write a program to store a word dictionary in a binary search tree using a linked list where each node contains a word and its meaning. Use rules of binary search tree as
 - a. Every word in n's left subtree is less than the word in a node n.
 - b. Every word in n's right subtree is greater than or equal to the word in node n.
 Implement the basic methods of the binary search tree that includes methods such as insert, find, delete, traverse methods.

The dictionary is available as a text file with this assignment. Read the text file line by line and split each line into two parts i) word and ii) meaning. The definition of tree node should be defined as shown below

	Aback	adv. take aback surprise, disconcert. [old english: related to *a2]	
--	-------	---	--

```
Class Node{
    String word;
    String meaning;
    Node left;
    Node right;
    ... }
```

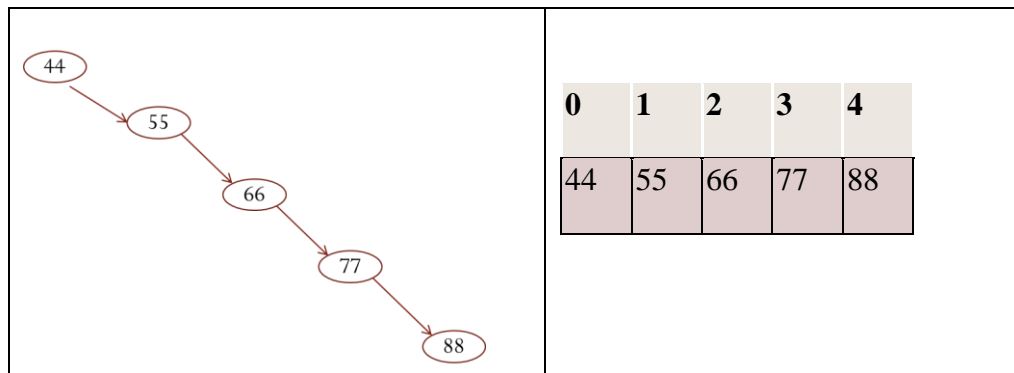
Note:

The list of words in the dictionary is in sorted order and if you insert words in the same order then BST will be created as a degenerated tree. Therefore, you can analyze by reordering the rows in a given dictionary if it can help in making a balanced tree.

3. Balancing the unbalanced BSTs method, you can choose to implement public void `balanceBST()`, which will balance an unbalanced BST. An unbalanced BST contains right and left subtrees with a significantly different number of nodes in each subtree.

Implementing public void `balanceBST()` To balance a BST, you want to get the values into a sorted array. Then you can recursively divide the array into halves, inserting middle values of each of the recursive halves (kind of like a binary search algorithm) into a new tree.

- Use `inorderTraversal()` as you did in the lab, rather than to print, store the values in the array to get BST values in sorted order in array as shown below.



- Now create the BST.
- Call the recursive method `balanceRecursive(int[] array, int first, int last)`, where `first` and `last` are the first and last index of the array, calculate the array mid and insert the return value in BST.

To implement `balanceRecursive(int[] array, int first, int last)` Think about how `binarySearch` worked. You'll want to do something similar. The left child will be the mid-value from the first half and the right child should be set by taking to mid-value from the second half, do this as two recursive calls in the method. The base case should return null (this corresponds to a child of a leaf).

Following should be the final balance tree

