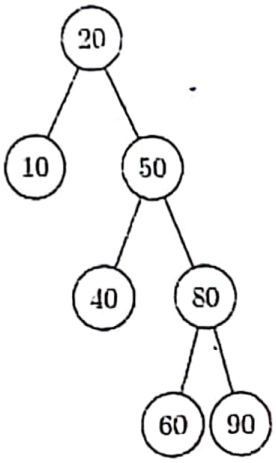
Université Libanaise
Faculté d'Information
Br I

الجامعة اللبنانية
كلية الاعلام
الفرع الاول

الدورة الاولى للعام الجامعي
2024/2023

المرحلة: اجازة	المادة: Data Structure and Algorithms
السنة المنهجية: الثانية	المدة:ساعتان
الدورة الأولى	الاستاذ: د. نادين زبيب

Exercise I: (40 pts)



Question 1: Define a Binary Tree data structure
Define the classes needed for a Binary Search Tree

Question 2: Recursion
Write a **recursive** method in the class BST that returns the sum of all values in the leaf nodes (leaves) in a Binary Search Tree that are smaller than the value in the root (these are the nodes in the left subtree of the root).

Question 3: Insertion into a Binary Search Tree
Remember the rules of a Binary Search Tree, and implement a function which accepts an integer and inserts it into the tree. Test your code by inserting some nodes into the above tree, and make sure to try various edge cases.

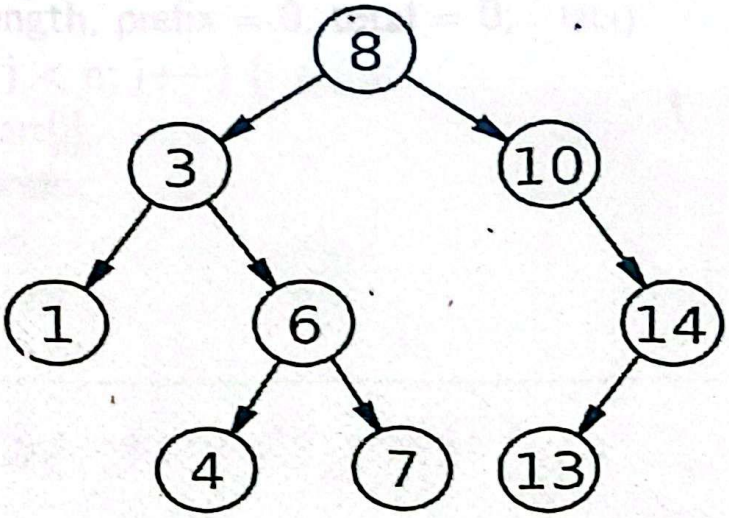
Question 4: Deletion from a Binary Search Tree
Implement a function that accepts an integer, and deletes one node with that value, if it exists. Remember to cover the 3 different cases of tree-deletion.

Question 5: Searching a Binary Search Tree
Both insertion and deletion rely on being able to find a node in a BSTree. Simplify the above two functions by extracting the "Search" logic to a standalone function. You may choose to implement two different functions, one for each of Insertion / Deletion (and any particular needs they may have)

Exercise II: (25 pts)

Part 1 [15 pts]

Consider the binary search tree pictured in the following tree. What is the output of Post-Order traversal, pre-order and in-order?



Part 2 [10 pts]

1- Show the result of inserting 105, 113, 96,1 8, 21, 4, 9, 10, 77, 87, 12, 1, and 90, one at a time, in an initially empty **Binary Search Tree. [5 pts]**

2- What will the tree look like after deleting: 105? **[5 pts]**

Exercise III. (25 pts)

Write the extractPartFromStack method which takes the original Stack, a start index, and an end index as input and returns a Stack which contain all values from the original stack between start index and end index. **For example:**

If the original stack was

12
4
7
-1
6
4
-5
8

And start index = 1 and end index =4

The resulting Stack will contain

4
7
-1
6

Exercise IV: Algorithms Analysis (15 pts)

Give a big-Oh characterization, in terms of n, of the running time of the below method.

```
26  /** Returns the sum of the prefix sums of given array. */
27  public static int example4(int[ ] arr) {
28    int n = arr.length, prefix = 0, total = 0;   Θ(1)
29    for (int j=0; j < n; j++) {   Θ(n)
30      prefix += arr[j];   Θ(n)
31      total += prefix;   Θ(n)
32    }
33    return total;   Θ(1)
34  }
```