Université Libanaise
Faculté d'Information

Br I

الجامعة اللبنانية
كلية الاعلام
الفرع الاول

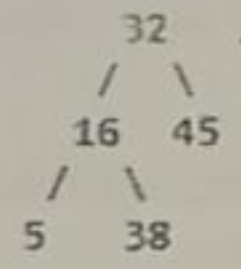
الدورة الاولى للعام الجامعي
2025/2024

المرحلة: اجازة
السنة المنهجية: الثانية
الدورة الاولى

المادة: Data Structure and Algorithms
المدة: ساعتان
الاستاذ: د. نادين زبيب

Exercise I: (20 pts)

1- This is not a valid binary search tree. What's wrong with it?



2- Use the Binary Search Tree insertion algorithm to insert the keys 1, 2, 3, 4, 5, 6, 7 into a initially empty BST (in that order!).

a- What is the height of the resulting tree?

b- Write an algorithm (recursion) that count the number of leaves in the tree. That is the number of nodes that have their left and right nodes both null.

c- Write an algorithm that calculate the height (number of levels) of the tree

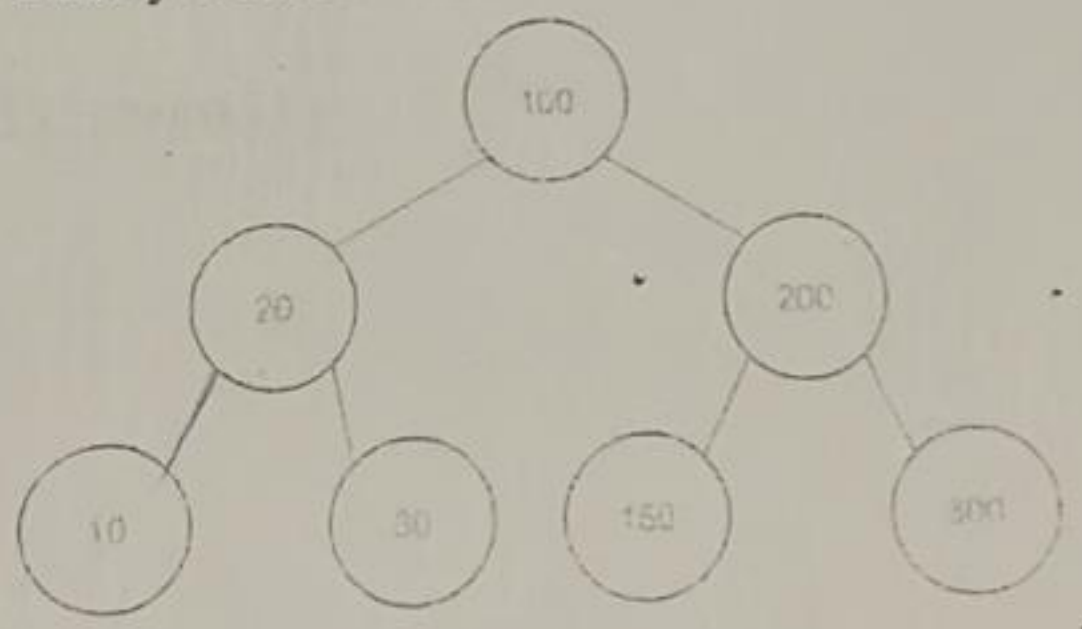
Exercice II: (35 pts)

Part 1

Given a BST in pre-order as {13,5,3,2,11,7,19,23}, draw this BST and determine if this BST is the same as one described in post-order as {2,3,5,7,11,23,19,13}

Part 2

1- Given a Binary Search Tree. The task is to print the elements in in order, preorder, and post order traversal of the Binary Search Tree.



Binary Search Tree

2- Show the result of inserting 105, 113, 96,18, and 90, one at a time, in an initially empty **Binary Search Tree.**

3- What will the tree look like after deleting: 105?

Exercise III. (35 pts)

Note: To solve the exercise below use the LinkedList implementation (used in class) which contains the below classes:

- NodeData
- Node
- LinkedList

Add a method in the application named divisibleByN(LinkedList list1, int n) that takes as parameters from the main a linked list list1 and and an integer n, and returns the number (counter) of nodes that hold a value that is divisible by n.

Create an application in which you:

a- Create a linked list and add to it the values 6, 4, 10, 9, 1.

b- Read an integer n from the user.

Call the method divisibleByN to output the number (counter) of nodes that hold a value that is divisible by n.

For example, for the above list if n is 3 then the method will return 2 since only 6 and 9 are divisible by 3.

Exercise IV: Algorithms Analysis (10 pts)

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

```
public static int example5(int[] first, int[] second) { //
assume equal-length arrays
int n = first.length, count = 0;
for (int i = 0; i < n; i++) { // loop from 0 to n-1
int total = 0;
for (int j = 0; j < n; j++) // loop from 0 to n-1
for (int k = 0; k <= j; k++) // loop from 0 to j
total += first[k];
if (second[i] == total) count++;
}
return count;
}
}
```