Faculty of Mathematical Economics

Data Structures and Algorithms

Instructor: **Nguyen Thanh Tuan**
DSEB Class of 2021 - 2024

Homework Assignment Week 8

Topic: Tree
Date Created: March 16, 2023

Problem 1: BinaryTree class

a. Implement a `Node` class and a `BinaryTree` class.

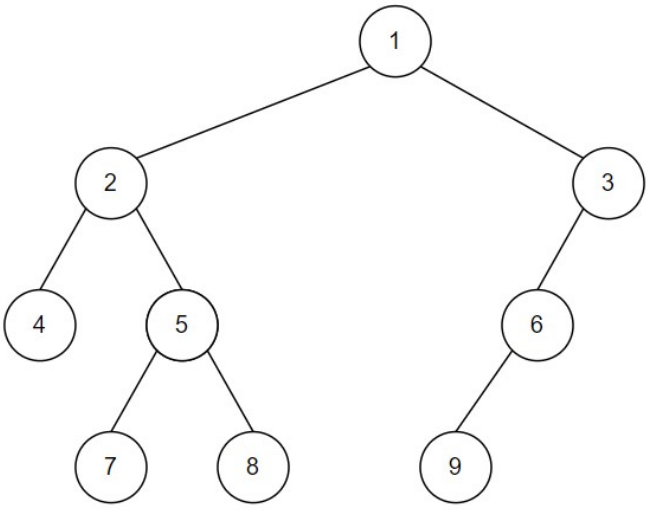
- Each node contains its value, parent, left and right children.
- The `BinaryTree` class should have common methods: `add_left`, `add_right`, `is_root`, `is_leaf`, `check_ancestor` (method to check if a node is parent of other node).
- Build `height` and `depth` methods to find the height and depth of a random node.
- Using `__str__` or `__repr__` to print out all elements in the tree.

Note:

- In `add_left`, `add_right` methods, you should raise appropriate Exception if the node already has a left or right child.
- **Optional:** You will get bonus points if you print out elements in a friendly format.

b. Check your implementation by performing these tasks:

- Create a tree as below:



- Find the height and depth of node 5.
- Using overloading to print out all elements in the tree.

Problem 2: Leaf Nodes

Implement a function to return the sum of all the leaf nodes that are the right child of their parent of the given binary tree.

Example 1:

```
'''Input:
          1
        /   \
      2       3
    /  \     /
  4     5  6
'''
>>> 5
```

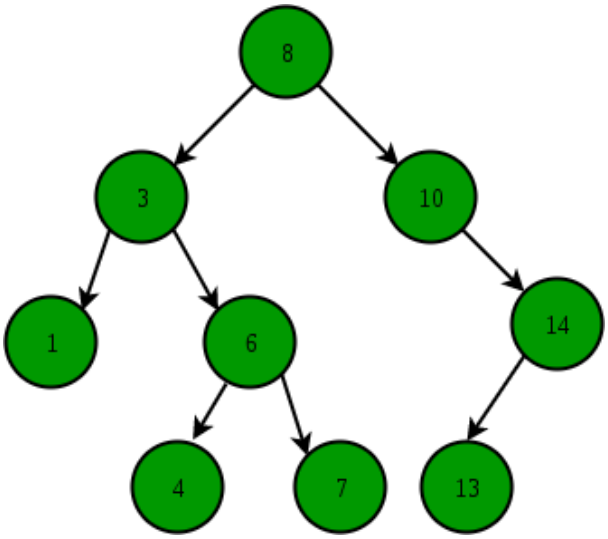
Example 2:

```
'''Input:
          1
        /   \
      2       3
    /  \       \
  4     5        6
   \            / \
    7          8    9
'''
>>> 5 + 7 + 9 = 21
```

Problem 3: Binary Search Tree

A Binary Search Tree (BST) is a node-based binary tree data structure which has the following properties:

- The left subtree of a node contains only nodes with keys lesser than the node's key.
- The right subtree of a node contains only nodes with keys greater than the node's key.
- The left and right subtree each must also be a binary search tree.
- There must be no duplicate nodes.



a. Do the following tasks:

- Build function `insert` to add a new value into a BST.
- Implement a function `iterative_search` to check if a given number is in a BST or not without recursion.

b. Check the implementation:

- Create a BST as above using `insert`.
- Check if 0, 3, 4 and 12 are in the BST.

Guidelines for submission

- Your submission must be under the `.ipynb` format.
- Your submission will be graded and it is likely that homework grade will contribute as a component in your GPA.
- If your submission is later than the due date without special consideration approval, you will receive a penalty on your mark.