Assignment 5

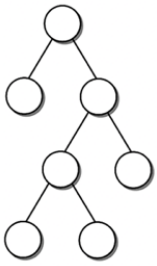
Advanced Programming (INFO135)

Published at: 13:00, Friday, 25.03.2022
Deadline: 13:00, Friday, 01.04.2022

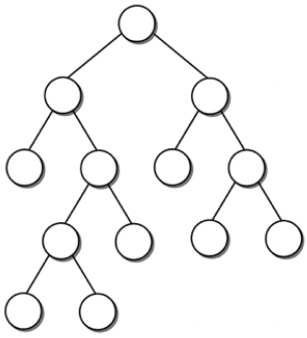
---

1. Which of the following Trees are Full Binary Tree.

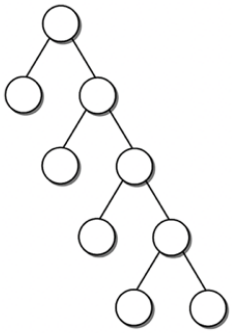
Binary Tree 1



Binary Tree 2

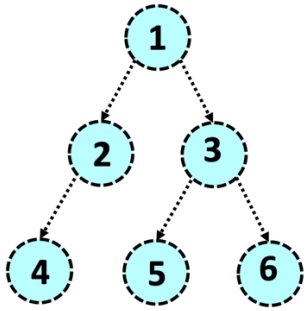


Binary Tree 3



---

2. Use the implementation of **Binary Tree** (based on **List of Lists**) provided in Lectures and write a function called **`make_tree()`** that builds the following tree and prints it in the output.

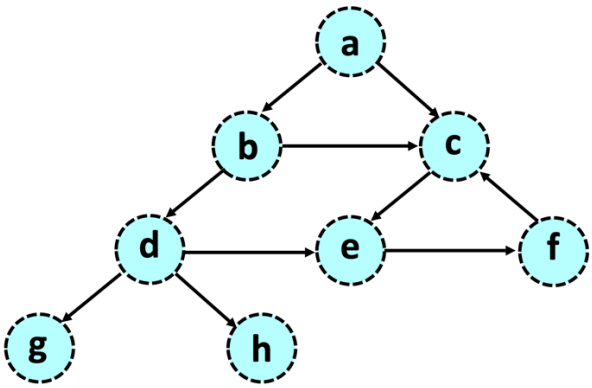


---

3. Write a function **`build_my_graph2()`** that:
   a) creates the following Graph.
   b) runs Depth First Search (DFS) algorithm starting from node 'a' and prints all the visited nodes.

What is printed in the output when you run the function?

Note: you can use the implementation of Graph class and the DFS algorithm (provided in Lecture notes).



---

4. Use the **Binary Search Tree (BST)** class (provided in Lecture notes) and write two new methods that are described in the following:
   a) `compute_sum()` that computes the sum of all the node values in the BST
   b) `compute_count()` that computes the total number of nodes

   Note: you can assume the values of the nodes (vertices) within the tree are all numerical.

```
class BinarySearchTree:

    def __init__(self, value=None):
        …

    def is_empty(self):
        …

    def insert(self, value):
        …

    def compute_sum(self):
        <your code here>

    def compute_count(self):
        <your code here>


my_tree = BinarySearchTree()

my_tree.insert(2)
my_tree.insert(4)
my_tree.insert(6)
my_tree.insert(8)
my_tree.insert(10)
```

```
print('sum:', my_tree.compute_sum())
print('number of nodes:', my_tree.compute_count())
```

[Output]

```
sum: 30
number of nodes: 5
```