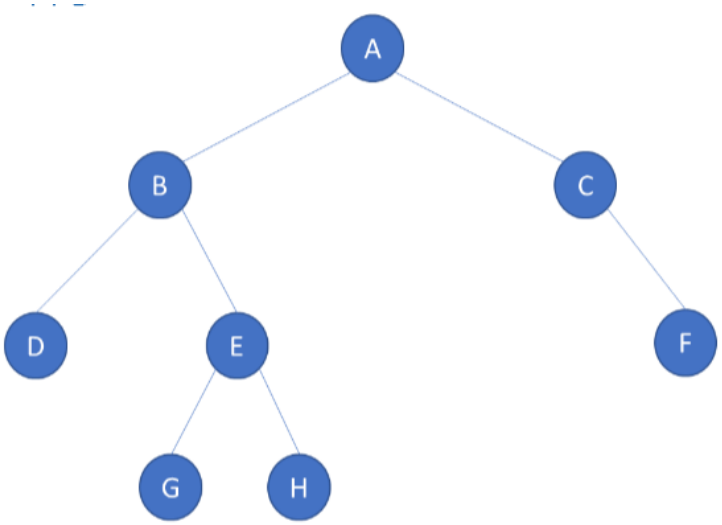
You will understand how tree structures work and learn how to use them.

Approval

This exercise can be delivered in groups of two students. The exercise must normally be approved by a student assistant during the lab hours.

Q1: Task 1: Understanding Tree structures



Given the tree above

a) Which node is the root node?

b) Which nodes are leaf nodes?

Fill in the table:

Node	Leaf	Sibling	Depth
A			
B			
C			
D			
E			
F			
G			
H			

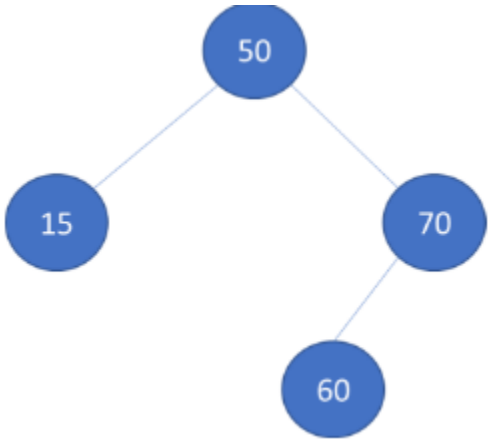
Write the nodes in the tree for the given patterns:

a. Preorder order

b. Inorder sequence

c. post order sequence

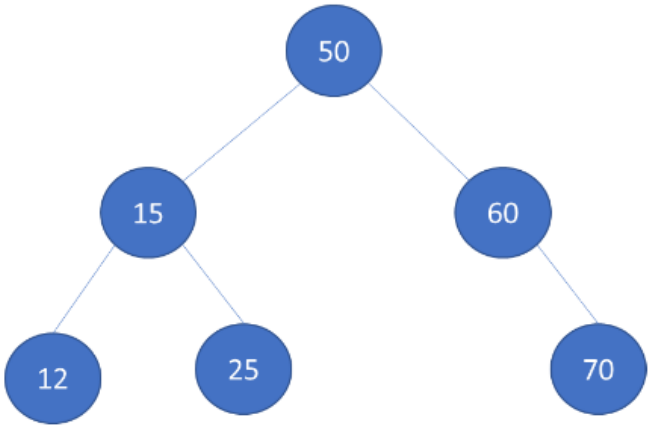
Q2: AVL tree



A) Given the AVL tree above, which you can assume is balanced. Insert a new node with value 25.

1. Draw the AVL tree

2. Is the AVL tree balanced after this insertion? If not, show each step of the process to turn it into a balanced AVL tree

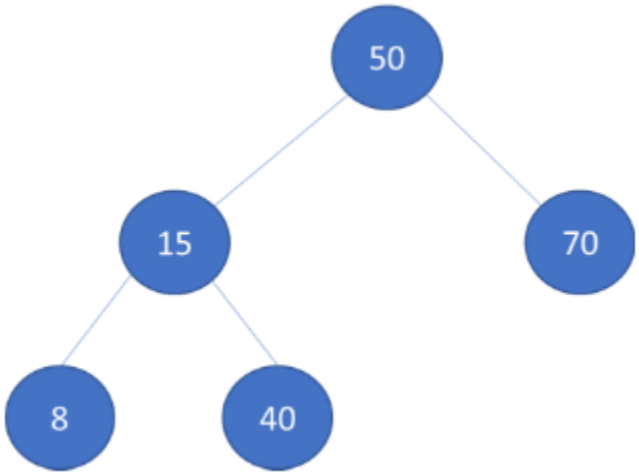


B) Given the AVL tree above, which you can assume is balanced. Insert the new node 85.

1. Draw the AVL tree

2. Is the AVL tree balanced after this insertion? If not, show each step of the process

to turn it into a balanced AVL tree



C) Given the AVL tree above, which you can assume is balanced. Insert the new node 35.

1. Draw the AVL tree

2. Is the AVL tree balanced after this insertion? If not, show each step of the process

to turn it into a balanced AVL tree

Q3: Use of Tree structures

a) Starting from the given code for a binary tree (binaertre.py), write a recursive method that calculates the number of leaf nodes in the tree.
b) Create a class for preorder iterator for the given code for a binary tree (binarytree.py). You can follow the postorder iterator which is already in the given code. Note that the preorder iterator can be made significantly simpler than the postorder iterator.

c) Given the file vindmaalinger_redusert_mer.txt as well as distributed code to read in this file (vindmaaling.py). Use the given AVL-tree implementation (avl_tre.py) to create a map

over wind measurements to make the following queries, as well as implement the queries:

a. Find how strong the wind was at midday (at 12:00) on 4 September 2010. That keeps returning the nearest wind measurement after the specified time.

b. Find how strong the wind was at its strongest on 7 December 2011. You can do this by asking about all wind measurements on this date (from 7 December 2011 at 00:00 to December 8, 2011 at 0:00), and then find the maximum by going through this list.

c. Optional: Extend the code so that the user can specify the times for tasks a and b. Note that the file only contains measurements from the years 2009 to 2012. You can, for example, use the date 2011-12-25 (the time when Dagmar ravaged the northwest and Trøndelag)