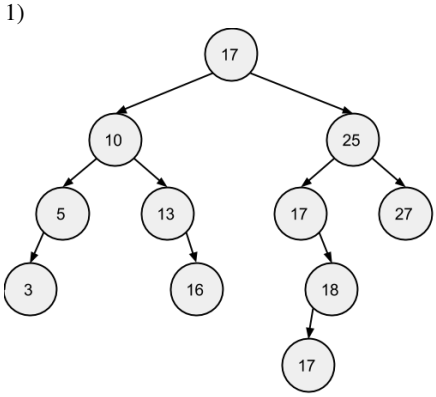
Joanna Klukowska
joannakl@cs.nyu.edu

DNHI Homework 5
Trees

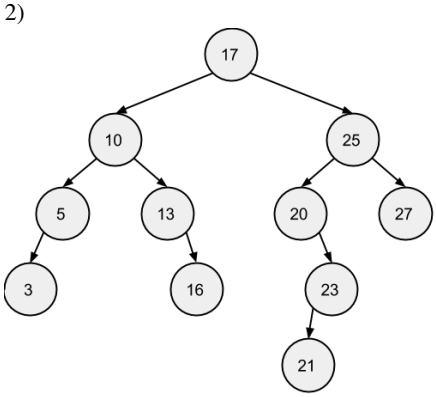
Problem 1

For each of the following trees state what kind of a tree it is (check all that apply).

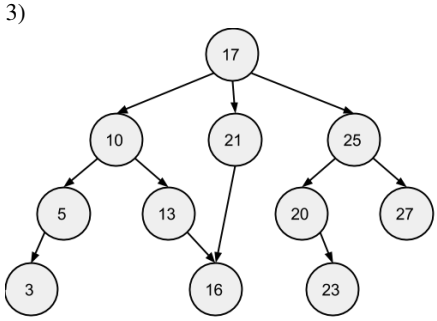
1)



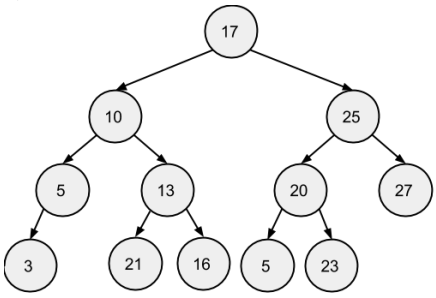
2)



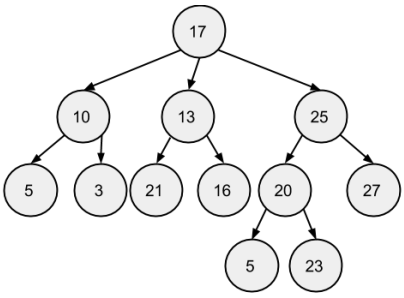
3)



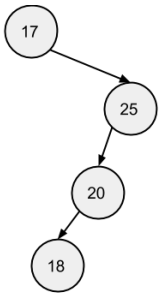
4)



5)



6)



Tree #	Not a tree	General tree	Binary tree	Binary search tree
1				
2				
3				
4				
5				
6				

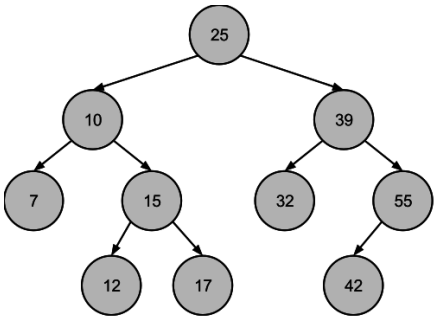
Problem 2

Specify inorder, preorder and postorder traversals of the fourth tree in Problem 1 and the original tree in Problem 3.

Joanna Klukowska
joannakl@cs.nyu.edu

Problem 3

Starting with the binary search tree shown below, show what the tree will look like after each of the following operations. Assume that remove method uses the predecessor when applicable. For each step modify the tree that results from the previous step (NOT the original tree).



1. `insert(21)`
2. `insert(8)`
3. `insert(30)`
4. `insert(35)`
5. `remove(17)`
6. `remove(7)`
7. `remove(39)`
8. `insert(60)`
9. `remove(25)`

Problem 4

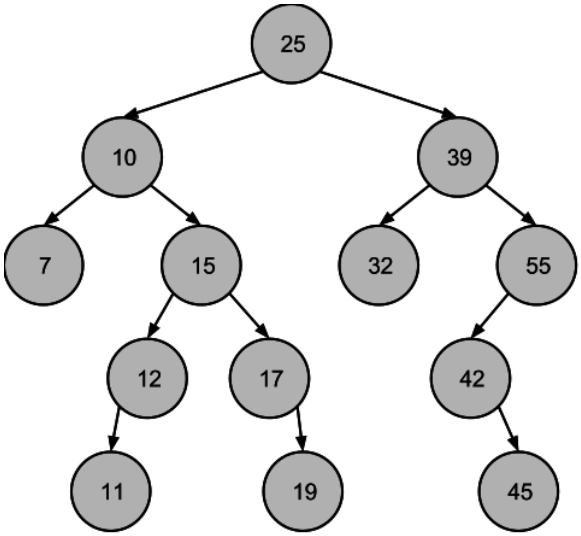
Implement an inorder traversal of a binary tree (this method should work for binary search tree as well) that uses iterative approach. Your method should be a method of a binary tree class. You can assume that there is a private data field called root that points to the root of the tree. You may specify this method using pseudocode, but make sure you are specific. You can assume that on visiting the node you print its content to the standard output.

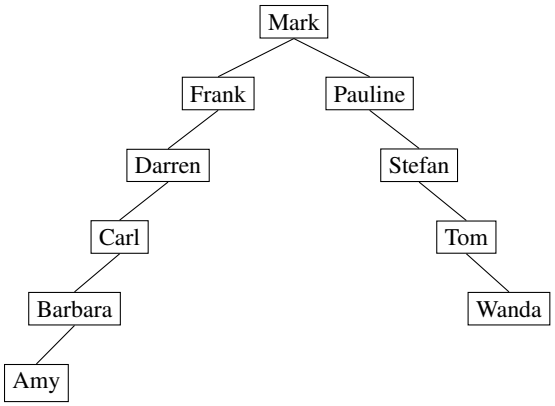
What changes would you have to make to convert this into a postorder traversal?

Problem 5

Given the following binary search trees, show the structure of the tree after a balancing operation has been performed on it. Assume that when selecting the middle, we always round down (or perform the integer division).

Joanna Klukowska
joannakl@cs.nyu.edu





Problem 6

Given the trees in Problem 5, show their preorder and postorder traversals.

Problem 7

Given the left tree in Problem 5, show the tree after the following operations. Assume that remove operations use the successor to replace a removed node when appropriate.

1. `insert(14)`
2. `insert(33)`
3. `insert(35)`
4. `remove(39)`
5. `remove(10)`
6. `insert(16)`
7. `remove(25)`

Problem 7

Write a method of a binary tree that determines the size of the tree. You can write pseudocode. You cannot assume that there is a data field storing the size of the tree.

Problem 8

Write a method of a binary tree that determines the number of leaves in the tree. You can write pseudocode.

Joanna Klukowska
joannakl@cs.nyu.edu

Problem 9

Draw a binary tree for which the inorder and preorder traversals are as follows:

inorder: F E D B A C

preorder: B E F D C A

Problem 10

Given a node in a binary tree, write a recursive method that computes the height of that node. The nodes do not store any height information. You may use pseudocode. The height of a node is the number of edges from the node to the deepest leaf.

Problem 11

Given a binary tree with 3 levels (level 0, level 1 and level 2) what is the largest number of nodes that the tree may contain? what is the smallest number of nodes that the tree may contain?

Problem 12

Write a method of a binary (search) tree class that returns the sum of all the numbers stored in the nodes. Write another method that returns the sum of the numbers stored in the leaf-nodes. Write another method that returns the sum of the numbers stored at even numbered levels (assume that the root is at level 0, which is even).

Problem 13

Write a method of a binary search tree class that converts the tree to its mirror image (i.e., swaps left and right child for each node). Is the resulting tree a binary search tree?

Problem 14

Given a sorted array (increasing order) of integers, write an algorithm that creates a binary search tree of minimal height.