Faculty of
Information Technology
King Mongkut's Institute of Technology Ladkrabang

Name:..

Student ID:...

Assignment # 5: Trees

1. In your own words, describe the definition of a tree and its properties.

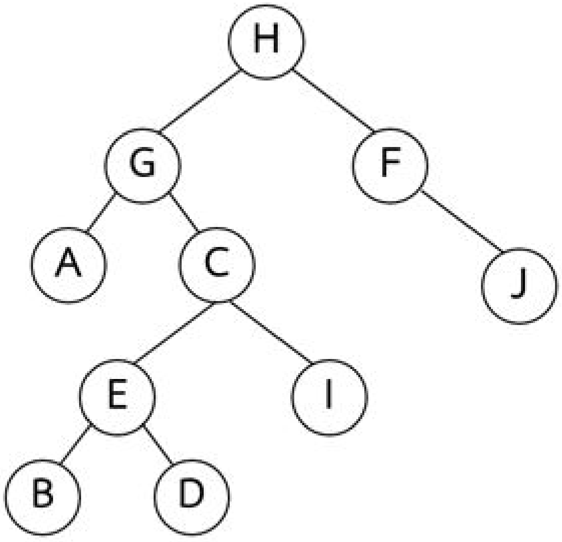
2. Describe the definition of a binary tree.

3. Please provide your answer on the following questions.

3.1 Given a binary tree of height 5, what are the minimum and the maximum number of nodes this tree can contain? Please also include the calculation steps.

3.2 Suppose there are 16 nodes in a binary tree, what are the minimum and the maximum height this tree can have? Please also include the calculation steps.

4. Given a binary tree below, please provide your answer on the following questions.



4.1 Give the node values corresponding to the following basic elements of a tree.

Element	Value
Root	
Parents	
Children	
Siblings	
Leaves	
Degree of tree	
Degree of node ‘E’	
Degree of node ‘J’	
Height of tree	
Depth of node ‘B’	
Depth of node ‘H’	
Ancestors of ‘D’	

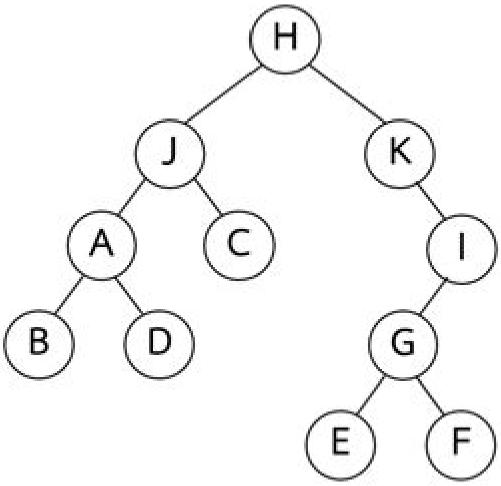
4.2 Write the sequence of nodes accessed using the preorder traversal algorithm.

4.3 Write the sequence of nodes accessed using the inorder traversal algorithm.

4.4 Write the sequence of nodes accessed using the postorder traversal algorithm.

4.5 Write the sequence of nodes accessed using the breadth-first traversal algorithm.

5. Given a binary tree below, please provide your answer on the following questions.



5.1 Give the node values corresponding to the following basic elements of a tree.

Element	Value
Root	
Parents	
Children	
Siblings	
Leaves	
Degree of tree	
Degree of node 'E'	
Degree of node 'J'	
Height of tree	
Depth of node 'B'	
Depth of node 'H'	
Descendants of 'J'	

5.2 Write the sequence of nodes accessed using the preorder traversal algorithm.

5.3 Write the sequence of nodes accessed using the inorder traversal algorithm.

5.4 Write the sequence of nodes accessed using the postorder traversal algorithm.

5.5 Write the sequence of nodes accessed using the breadth-first traversal algorithm.

6. Given the mathematical expressions in the following questions below, draw the (arithmetic) **expression tree** (infix form) representation and provide the sequence of nodes accessed using preorder and postorder traversal algorithms.

6.1 $b + cd$ where operator '+' is assigned as the root node.

6.2 $(2b^2 + c) / a$ where operator ‘/’ is assigned as the root node.

6.3 $ab^2 + 2cd - e^2$ where operator ‘+’ is assigned as the root node.

6.4 $e + (b^2 - 4ac) / 2d$ where operator ‘/’ is assigned as the root node.

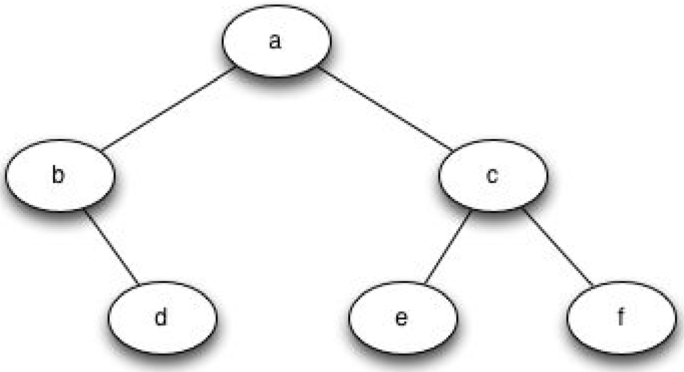
7. Given a class of **BinaryTree** and its insertion methods below:

```
def BinaryTree(r):
  return [r, [], []]

""" Insertion methods """
def insertLeft(root, newBranch):
  t = root.pop(1) #Obtain list that corresponds to the current
left child of the root
  if len(t) > 1:    #If the left child is not empty, push the
old left child as the left child of the new node.
    root.insert(1, [newBranch, t, []])
  else:              #If the left child is empty
    root.insert(1, [newBranch,[],[]])
  return root

def insertRight(root, newBranch):
  t = root.pop(2) #Obtain list that corresponds to the current
right child of the root
  if len(t) > 1:   #If the right child is not empty, push the
old right child as the right child of the new node.
    root.insert(2, [newBranch, [], t])
  else:             #If the right child is empty
    root.insert(2, [newBranch, [], []])
  return root

""" Accessor methods """
def getRootVal(root):
  return root[0]

def setRootVal(root, newVal):
  root[0] = newVal

def getLeftChild(root):
  return root[1]

def getRightChild(root):
  return root[2]
```

7.1 Draw the binary tree and provide the list form of the binary tree after executing the operations provided below.

```
x = BinaryTree('A')
insertLeft(x,'B')
insertRight(x,'C')
insertRight(getRightChild(x),'D')
insertLeft(getRightChild(getRightChild(x)),'E')
```

7.2 Write a Python code or pseudocode to implement a binary tree as depicted in the figure below using **lists** and provided insertion operations.



7.3 Using the same tree as in Question 7.2, draw an updated tree after the following operations:

```
insertLeft(getLeftChild(x),'R')
setRootVal(getRightChild(x), 'S')
insertRight(getLeftChild(getRightChild(x),'T')
insertLeft(getLeftChild(x),'U')
```