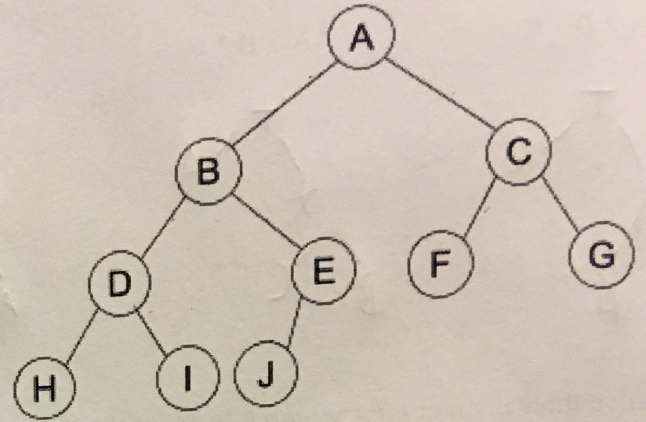
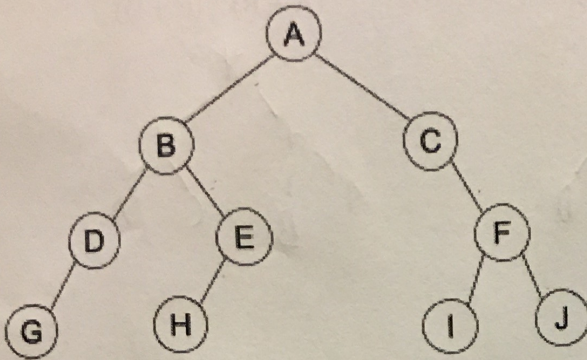


# Trees

Put your written answers in the comment section of BinaryNode.java.

Use the following trees to fill in the table.



Quantity	Left tree	Right Tree
List the leaf/leaves	G H I J	H I J
List the root/roots	A	A
Number of edges	9	9
List child/children of node D	G	H I
List parent(s) of node I	F	D
Height of tree	4	4
Is this a full tree?	No	No
Is this a complete tree?	No	Yes
Is this a balanced tree?	No	Yes
Post-order traversal	G D H E B I J F C A	H I D J E B F G C A
Pre-order traversal	A B D G E H C F I J	A B D H I E J C F G
In-order traversal	G D I B H E A C I F J	H D I B J E A F C G

1. Determine postorder traversal, given the following:

a. Preorder = BCAD

Inorder = CBAD

Postorder = C D A B

b. Preorder = DBACEGF

Inorder = ABCDEFG

Postorder = A C B F G E D

2. How many nodes are there in a full binary tree of the following heights? The first one is done for you:

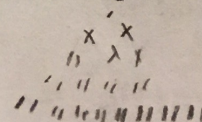
a. Height = 1 1 Node

b. Height = 2 3

c. Height = 3 7

d. Height = 4 15

e. Height = 5 31



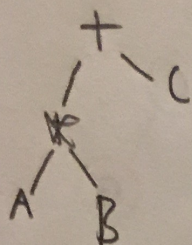


3. What formula describes the number of nodes in a full binary tree of height  $n$ .

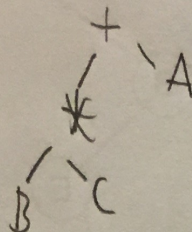
$$2^{n+1} - 1$$

4. Give parse trees that corresponds to the following formulas:

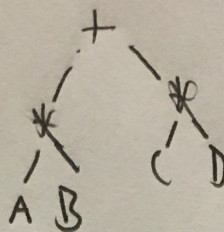
$A * B + C$



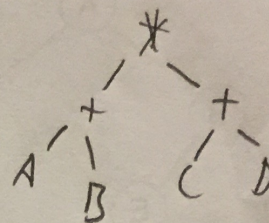
$A + B * C$



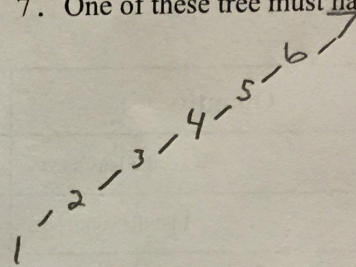
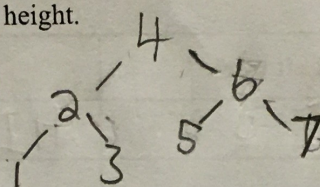
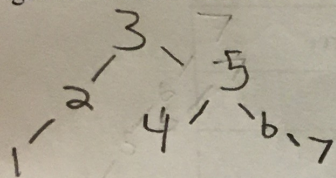
$A * B + C * D$



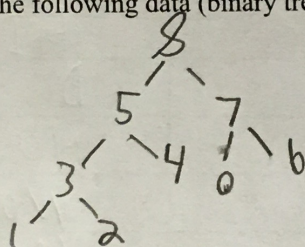
$(A + B) * (C + D)$



5. Give three distinct *binary search trees* for the following data: 1 2 3 4 5 6 7. One of these tree must have minimum height and another must have maximum height.



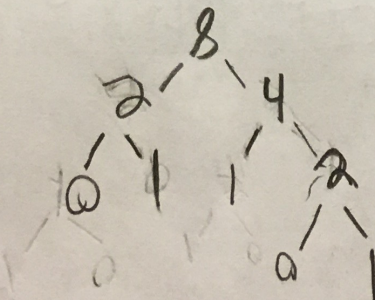
6. Draw a picture of a *max heap* for the following data (binary trees): 0 1 2 3 4 5 6 7 8



7. Determine the tree produced by `mystery(8)`, where `mystery` is a method in the `BinaryNode` class.

```

public BinaryNode mystery(int N) {
    BinaryNode result = null;
    if (N > 0)
        result = new BinaryNode(N, mystery(N/2), mystery(N/4));
    return result;
}
  
```



8. Complete/add the following **bold** functions in the following methods that are in the `BinaryTree<T>` class. Implementing these might necessitate adding methods to the `BinaryNode<T>` class.

```

public int getNumberOfLeaves()
public boolean isCompleteTree()
public boolean isMaxHeap()
  
```

```

public boolean isFullTree()
public boolean isBalancedTree()
public boolean isBST()
  
```