

Part A - coding

1. Write a method called `inToTree` of `ExpressionTree` class which takes a string of infix expression (fully parenthesized), resets the root of the expression tree to such tree which constructed from the expression. The root set to null of the expression contains characters other than value, operator, and parenthesis or if it's not an in-fix full parenthesized expression. You may assume each value is single digit only.

```
public void inToTree(String expression){
```

You may use a JCF Deque class as a stack to keep track of the parenthesis. It may call other helper method like recursive function that takes a string and return a binary tree node, if it calls, you have to write such method.

Extra Credit:

Write the expression tree constructor which takes an algebraic expression (infix but non-fully parenthesized expression) and construct the tree.

```
public ExprTree(String expression){
```

2. Write a print method of `ExpressionTree` class which print the expression in fully parenthesized infix. If it calls `isLeaf` method, then you should finish such method.

```
public void print();
```

3. Write `levelOrder` method of `Tree` class method which starts at the root of a tree and prints all elements of the tree in level-order (in one line).

```
public void levelOrder();
```

It may call a private `levelOrder` method which takes a queue of nodes waiting to be printed. If so, please implement the following method as well.

```
private void levelOrder(List<TreeNode<E>>)
```

Extra Credit: Write a general `Tree` class method which takes another tree, and check whether the parameter tree that starting at root is a subtree of current tree.

```
public boolean isSubTree(Tree);
```

Submit: Please download the given assignment `ExpressionTree`

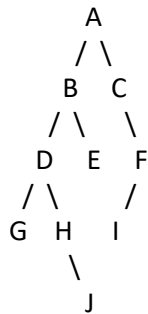
<https://venus.cs.qc.cuny.edu/~yang/sp20/cs313/assignment/A2ExpressionTree.java>

Change the file and class name to include your initials at the end. eg, if your name is John Smith, please change the file to `A2ExpressionTreeSJ` and include your name at the very top of the program as comments.

Email your java file to junbinliang66@gmail.com. Write CS313, section number (26-Tue/Thu 12:15 class, 37-Tue/Thu 6:30 class, 51-Fri 1:40 class), time of the class, your name, and A2 in the subject line of your email.

Part B – written (Submit as an pdf file, or hand-in in class.)

1. Consider following tree



- What is the height of the tree?
 - What is the height of node E?
 - What is the depth of the node E?
 - What is the output of post-order traversal?
 - What is the output of in-order traversal?
2. Consider following pre-order expression, draw the expression tree.
*+-123-4+56

2. Consider following post-order expression, draw the expression tree.
123*+4-56-/

3. Draw the following math algorithm as expression tree.
 $(3-(5+1) - 2) * (2+3*6) / 3$

4. Consider following function, it should remove even values in list,

```
public static void removeEven(List<Integer> list){  
    for (int i : list){  
        if (i % 2 == 0) list.remove(i);  
    }  
}
```

Does this one work? If not, write the statement that doesn't work.