

Lab 09

Jeong Wonil, Yoo Yeonil

Practice 1

Given Lab09Tester.java and Lab09Node.java, finish Lab09Tree

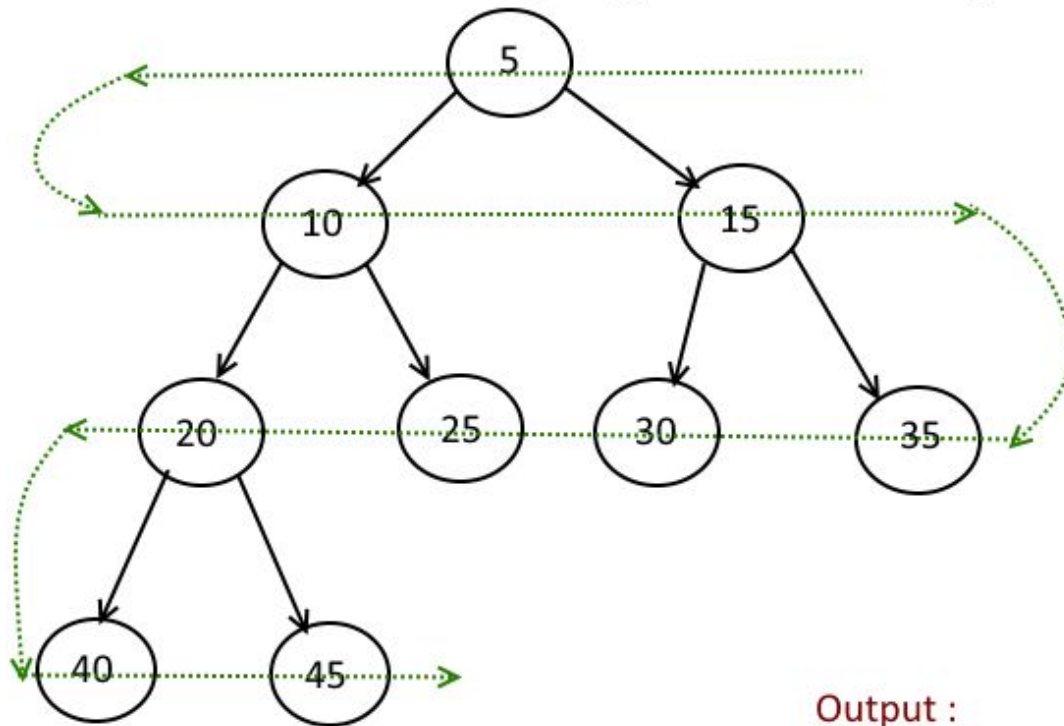
- `void insert(int v)` calls recursive method `void insert(int v, Lab09Node n)` and it travels left if $v \leq$ node's val, else right until cannot travel farther. Once it travels end, create new node of value of `v` and link current node's left or right to new node
- `void remove(int v)` removes node with `v` value

Practice 1

- Create three report method `String preorder()`, `String inorder()`, `String postorder()`, `String levelorder()` **each method calls recursive method** `String preorder(Lab09Node n)`, `String inorder(Lab09Node n)`, `String postorder(Lab09Node n)`, `String levelorderR()`
- `preorder` reports current node, then left path then right path
- `inorder` reports left path, current node, then right path
- `postorder` reports left path, right path, current child
- `levelorder` reports current level, next level from left child to right child

- `String printSideways()` returns string the tree in a sideways indented format, with right nodes above roots above left nodes, with each level 4 spaces more indented than the one above it.

Level Order Traversal in Zig Zag pattern OR Print in Spiral



Output :

5
10 15
35 30 25 20
40 45

Practice 1 expected result

```
preorder    : 10 6 2 7 8 15 13 11 20
inorder     : 2 6 7 8 10 11 13 15 20
postorder   : 2 8 7 6 11 13 20 15 10
levelorder  : 10 6 15 2 7 13 20 8 11
print sideways
```

```
      20
    15
  13
    11
10
      8
    7
  6
    2
```

Practice 2

A bottle: [Weight(kg) , capacity(L), volume(L)]

Given 3 empty bottles and a weight condition, you should examine how to meet the weight condition by completely draining out water in the bottles or filling in the bottles either from water faucet or **from other bottles**. **Each filling of bottles from water faucet should maximize volume of a bottle and draining should make a bottle empty**. All bottles(either empty or not)'s weight should be counted to meet the weight condition. Also, your program should find out whether the weight condition can be satisfied or not. (Assume a liter weighs 1 kg)

weight, capacity, volume => (int) [0 ~ 100] ;

weight condition => (int)[0 ~ 1,000]

Practice 2

[input from console]

[weight condition]

[Bottle1's weight] [`` 's capacity]...

[] []

8

2 5

2 2

1 2

Practice 2

[output to console]

[true / false]

true