CS310
Data Structures

K. Raven Russell
krusselc@gmu.edu
George Mason University

Today

- **Last Lectures**
  - Tree Introduction
  - Recursion Review
- **Today**
  - Tree Traversals (using trees and recursion)

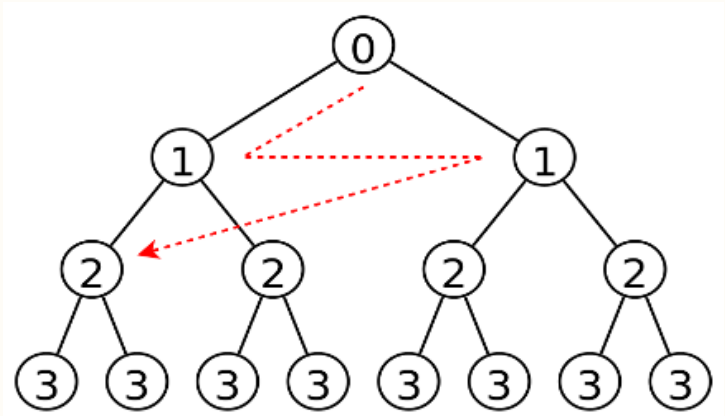
Common Tree Operations

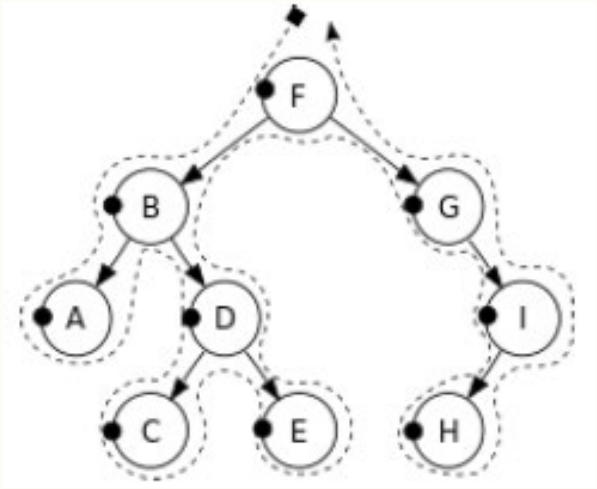
- Searching for an item
- Adding items
- Deleting items (synonymous with removing)
- Balancing
- Iterating = mention things one by one
  - all the items (in some order)
  - a section of a tree

Tree Traversals

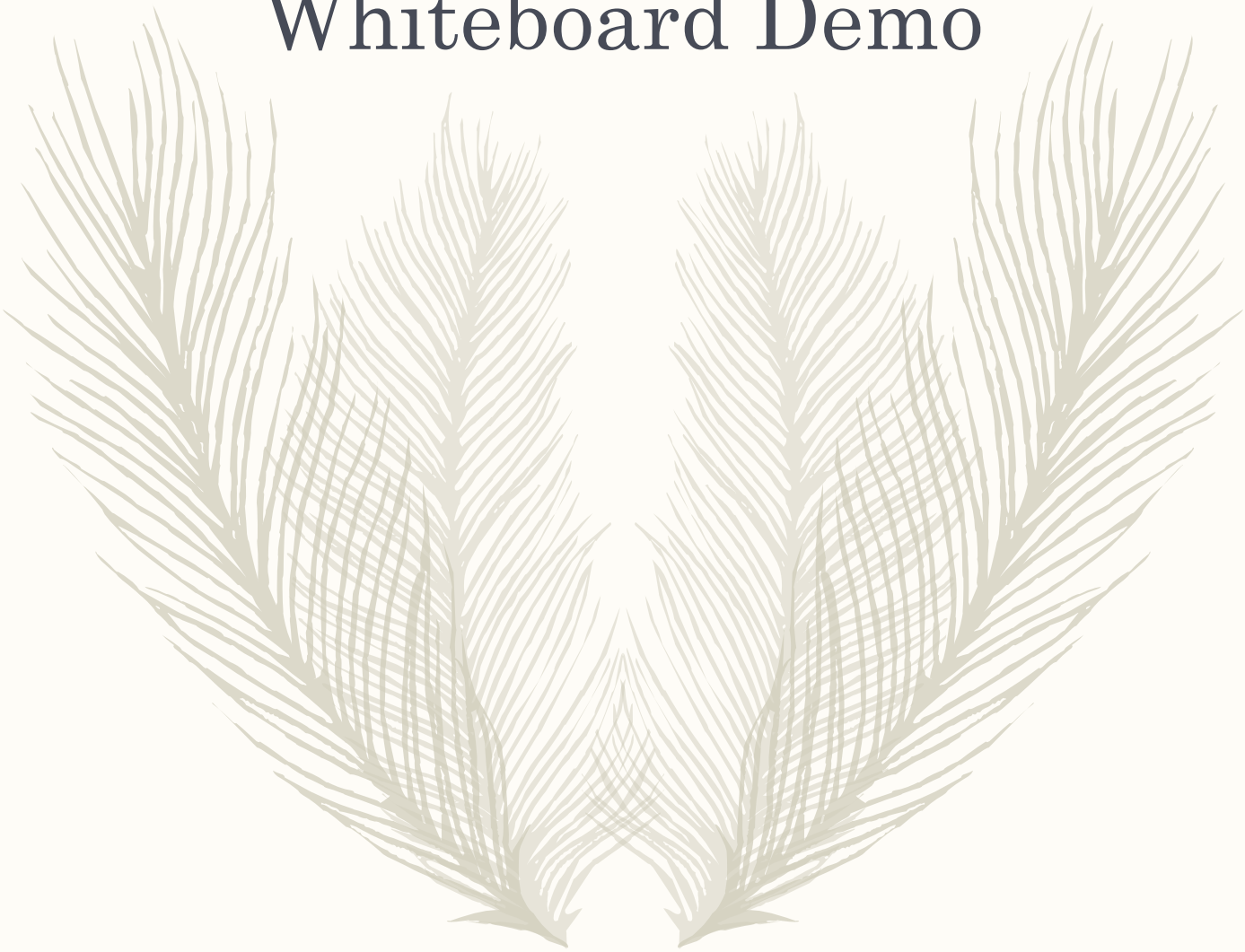
Weiss 18.4 describes *iterators*, Today we describe *operations*

- Two common types
  - breadth first - process things closest to the root first
  - depth first - follow a path all the way to its end and backtrack



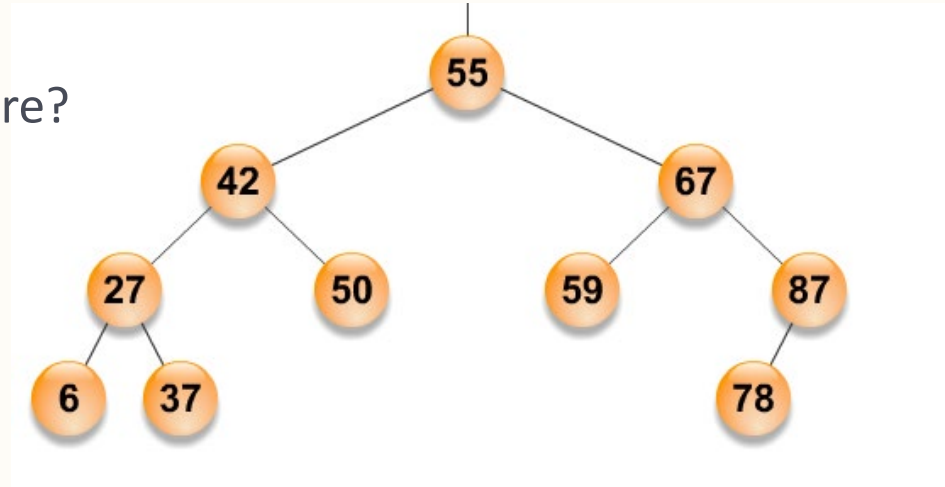


Whiteboard Demo



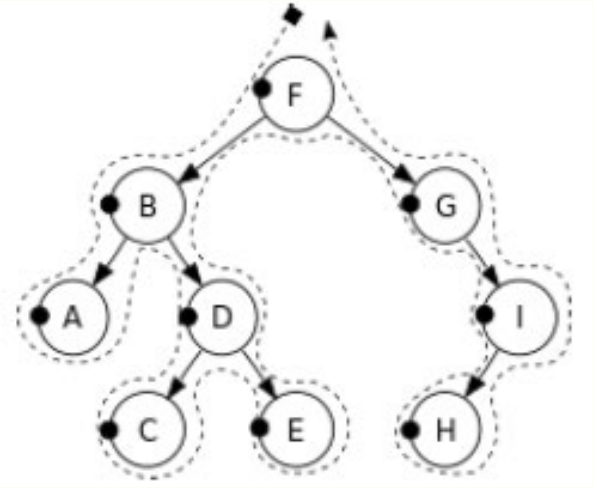
Tree Traversals: Breadth First

- Breadth first
  - process things closest to the root first
  - in trees, sometimes called "level-order"
- Stored in an array?
  - walk the array!
- Stored in a linked structure?
  - use a queue!



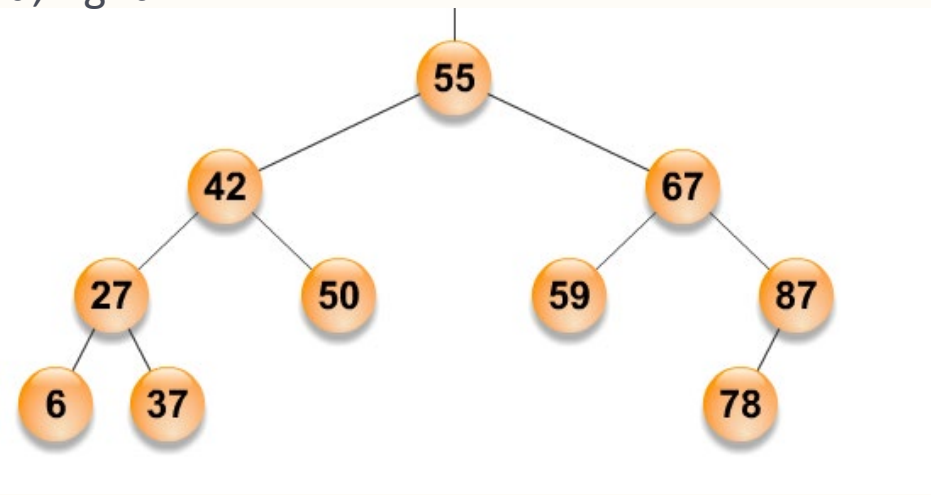
Tree Traversals: Depth First

- Three common depth first
  - Pre-, Post-, and In-Order
- Pre-order
  - self, left child, right child
- Post-order
  - left child, right child, self
- In-order
  - left child, self, right child



Tree Traversals: Depth First

- **Pre-order**
  - process order: self, left child, right child
  - Example: 55, 42, 27, 6, 37, 50, 67, 59, 87, 78
- **Post-order**
  - process order: left child, right child, self
  - Example: 6, 37, 27, 50, 42, 59, 78, 87, 67, 55
- **In-order**
  - process order: left child, self, right child
  - Example: 6, 27, 37, 42, 50, 55, 59, 67, 78, 87
  - search tree? sorted items!



Recursive Implementations

```
class Node<T> {
    T data;
    Node<T> left, right;
}

//use call stack instead
//of local stack!

inOrder(Node t) {
    if(t == null) return;
    inOrder(t.left);
    print(t.data);
    inOrder(t.right);
}
```

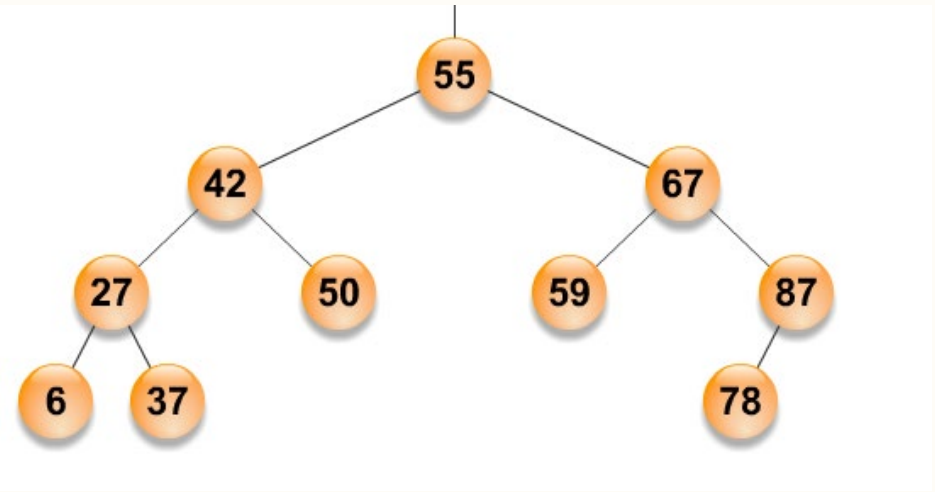
```
preOrder(Node t) {
    if(t == null) return;
    print(t.data);
    preOrder(t.left);
    preOrder(t.right);
}

postOrder(Node t) {
    if(t == null) return;
    postOrder(t.left);
    postOrder(t.right);
    print(t.data);
}
```

Whiteboard Tracing
(see next slides)

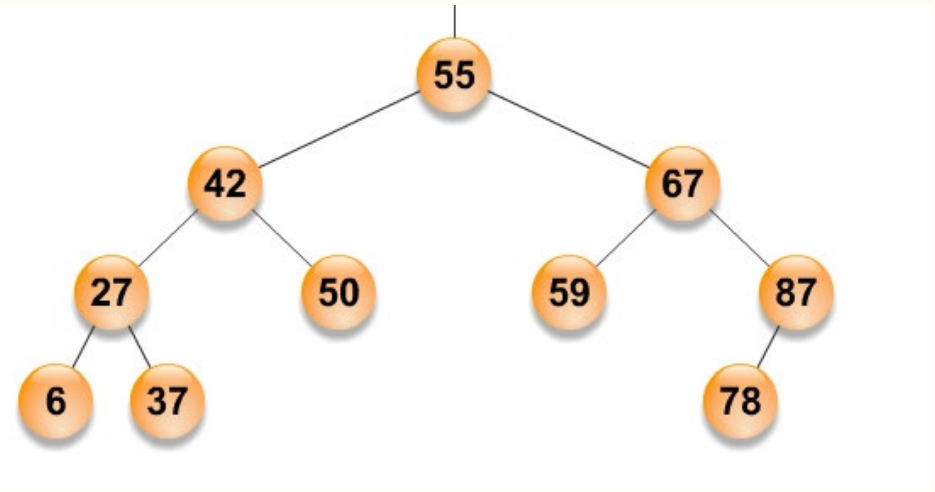
Tracing Pre-order Walk w/ Code

```
preOrder(Node t) {
    if(t == null) return;
    print(t.data);
    preOrder(t.left);
    preOrder(t.right);
}
```



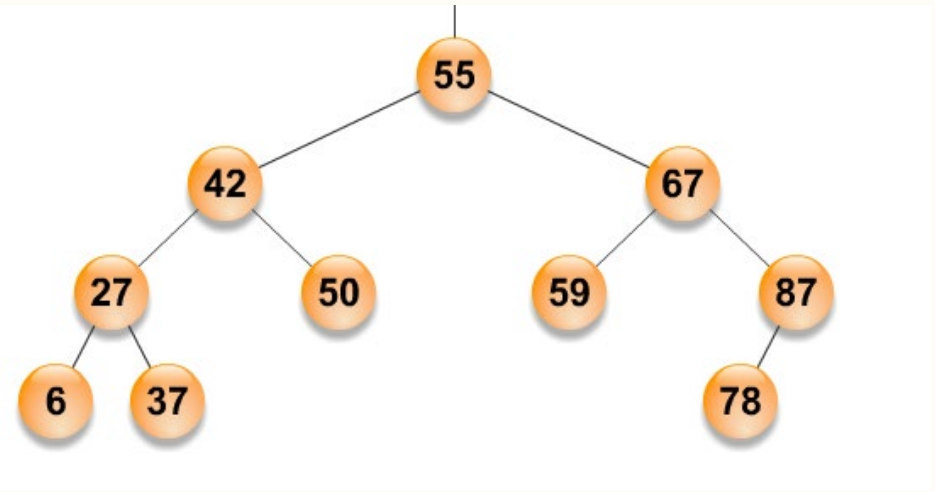
Tracing Post-order Walk w/ Code

```
postOrder(Node t) {
    if(t == null) return;
    postOrder(t.left);
    postOrder(t.right);
    print(t.data);
}
```



Tracing In-order Walk w/ Code

```
inOrder(Node t) {
    if(t == null) return;
    inOrder(t.left);
    print(t.data);
    inOrder(t.right);
}
```



Questions?

Which Traversal Should I Use?

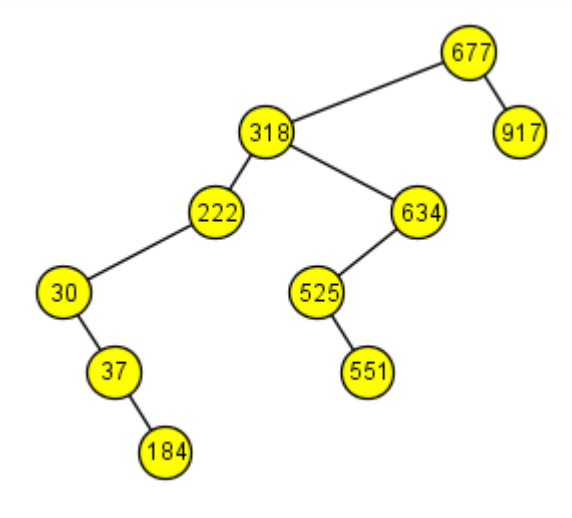
- If you want to push information **down** the tree…
  - Use pre-order
  - Idea: calculate something "here" and "give it" to the children.
- If you want to push information **up** the tree…
  - Use post-order
  - Idea: calculate something "below" and "give it" to me.
- If you want to push information from the **left to the right** in the tree…
  - Use in-order
  - Idea: calculate something based on information "to the left", pass it to the parent, then they can pass it "to the right"

Example: Pushing Info Down

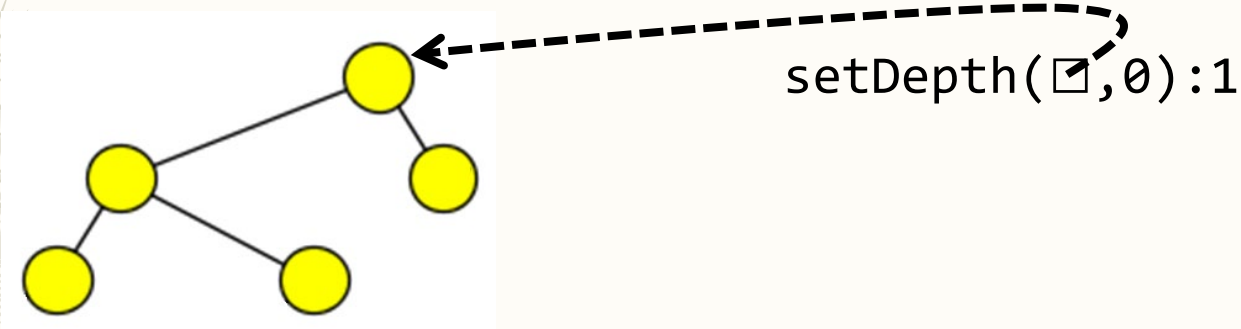
```
class Node<T> {
    T data;
    int depth;
    Node<T> left, right;
}

setDepth(Node t, int i) {
    if(t == null) return;
    t.depth = i;
    setDepth(t.left, t.depth+1);
    setDepth(t.right, t.depth+1);
}

setDepth(root, 0)
```

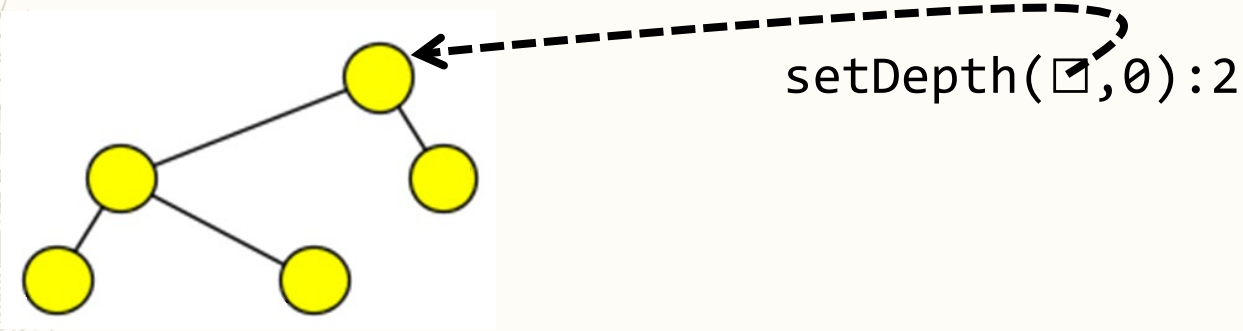


Example: Pushing Info Down



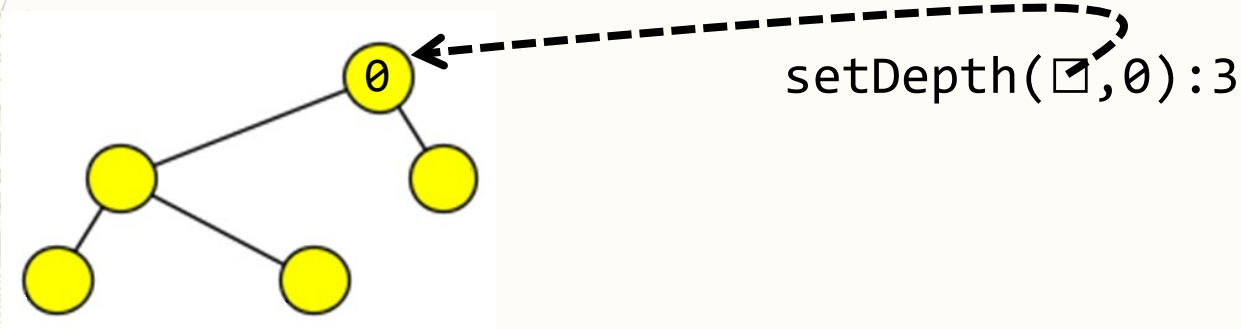

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



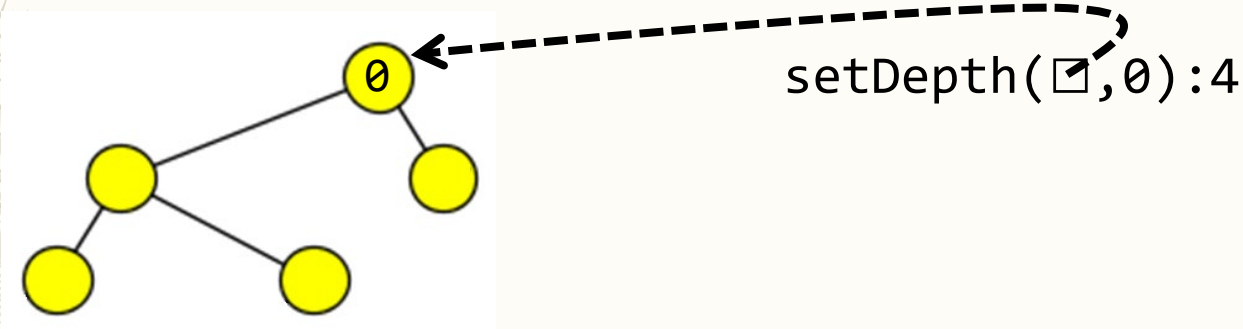

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



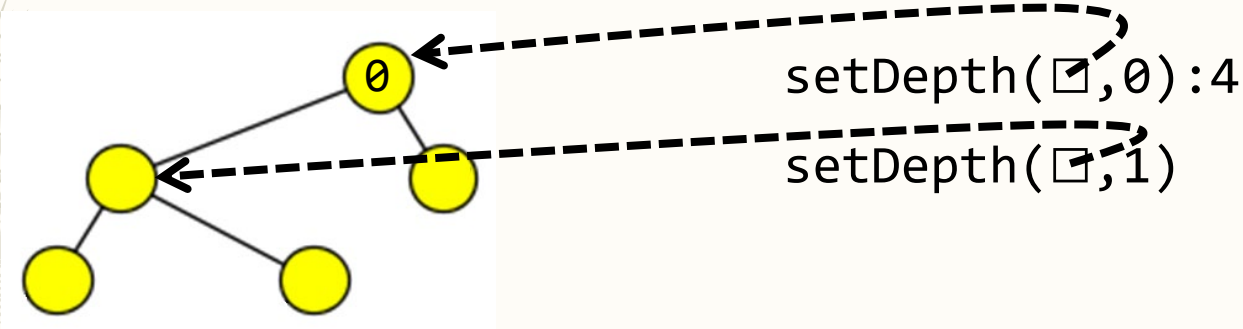

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



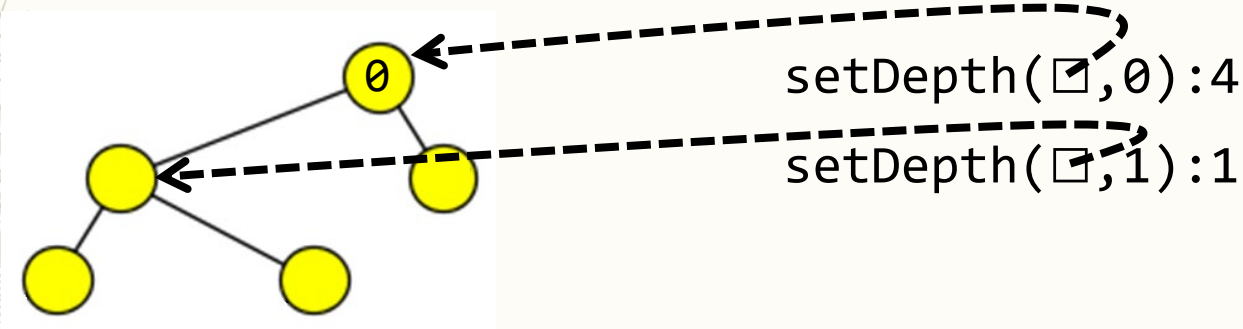

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



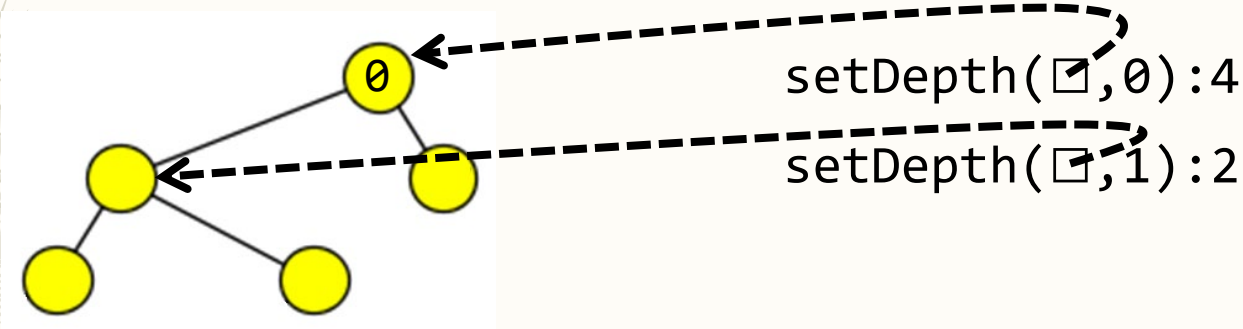

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



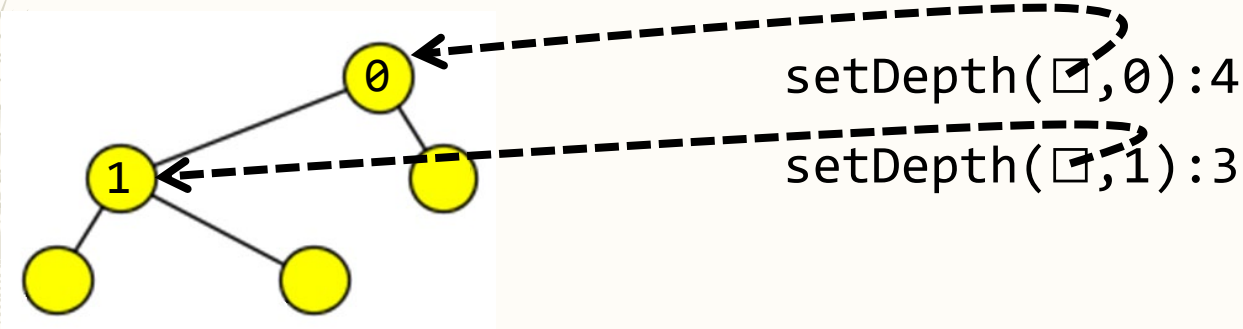

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



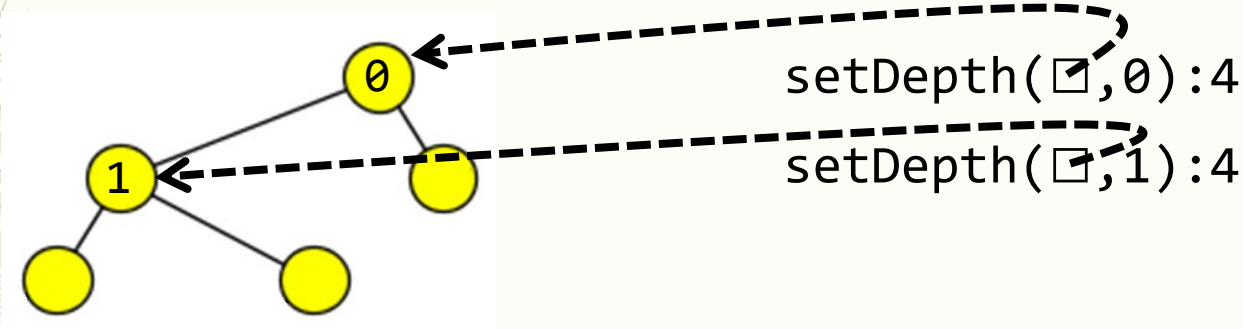

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



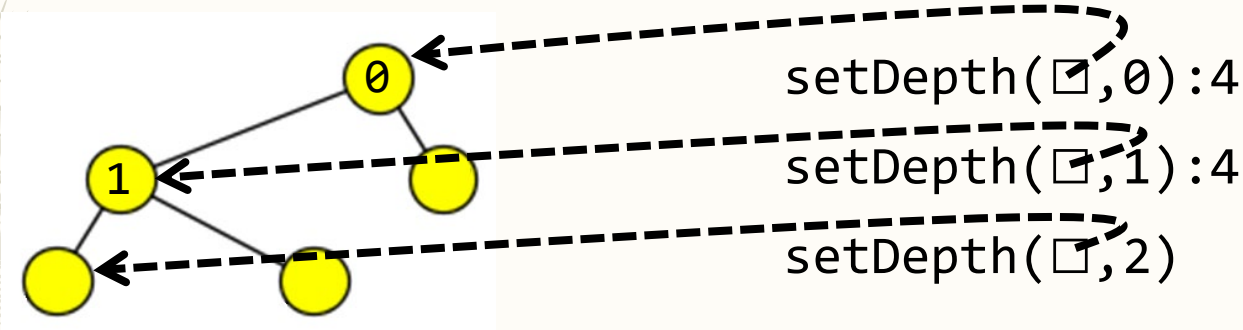

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



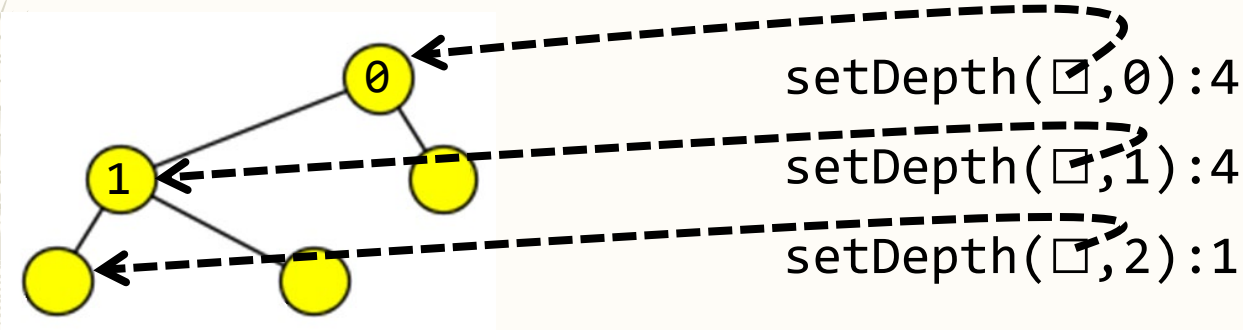

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



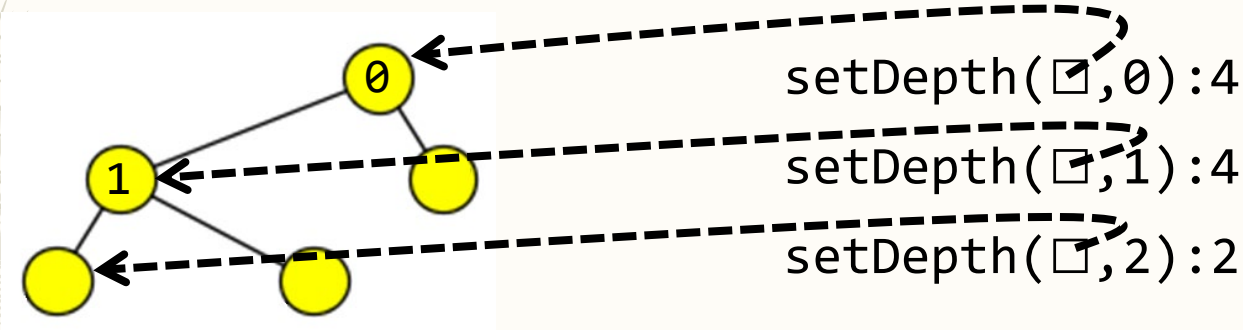

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



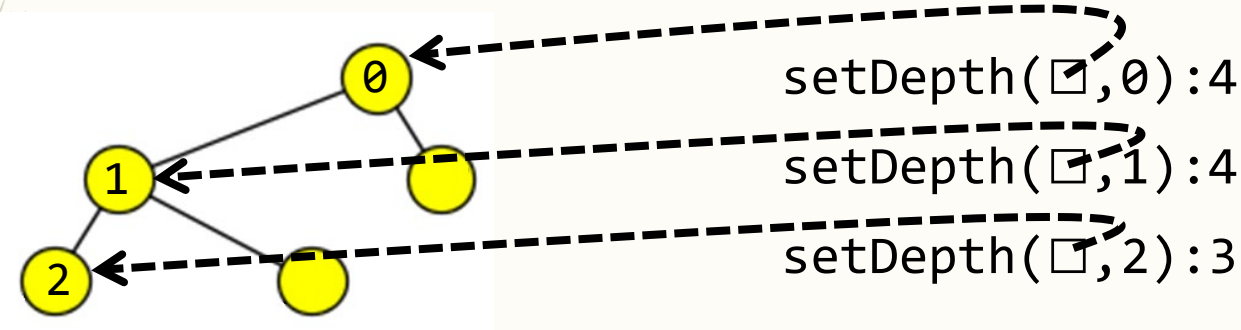

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



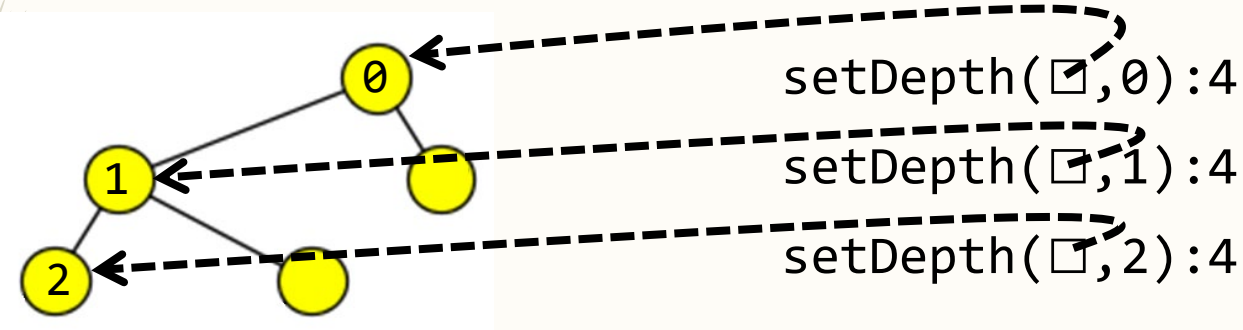

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



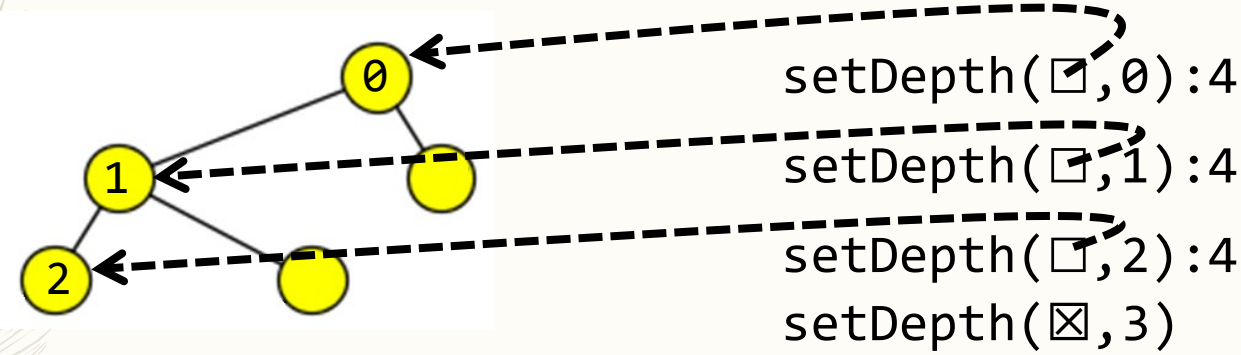

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



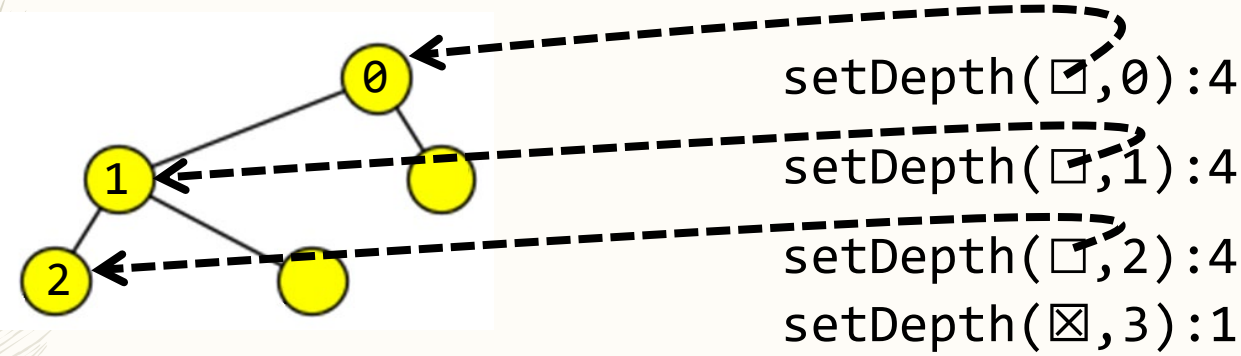

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



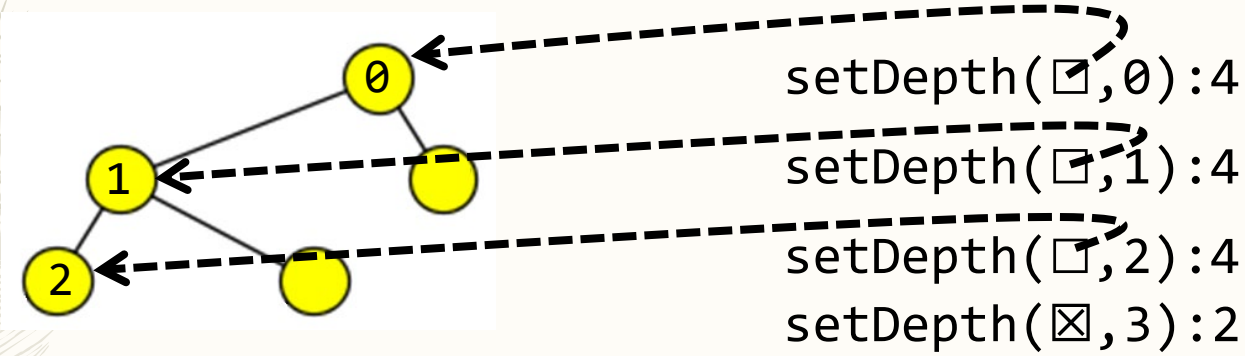

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



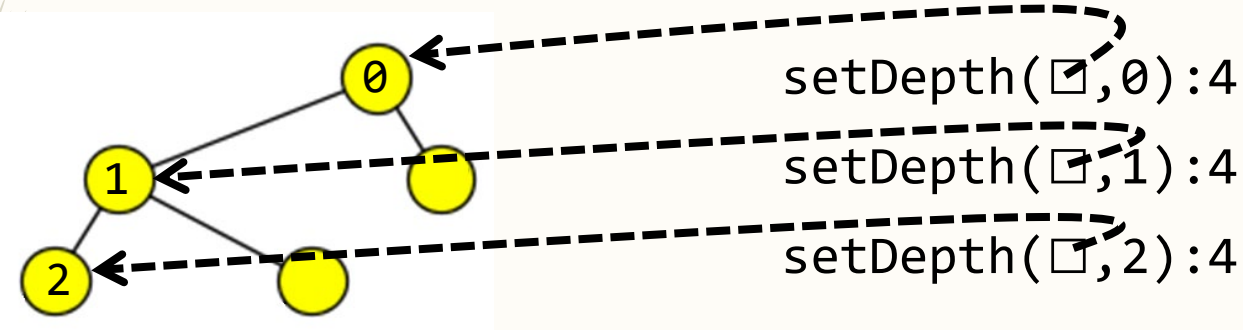

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



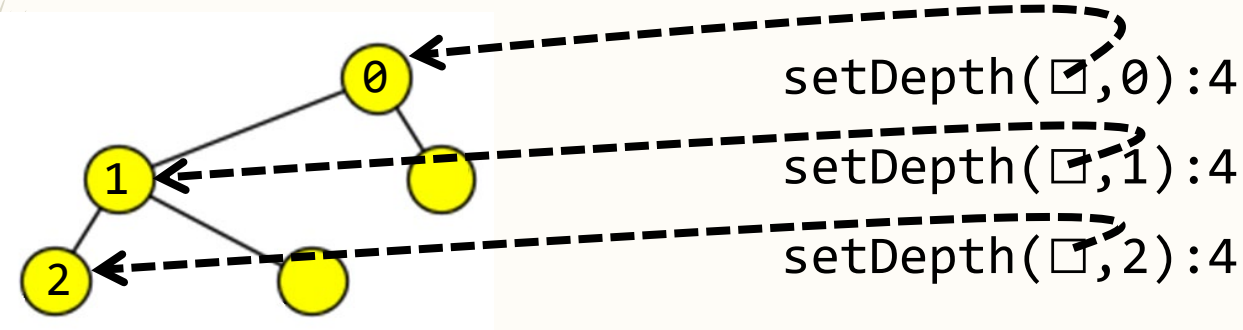

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



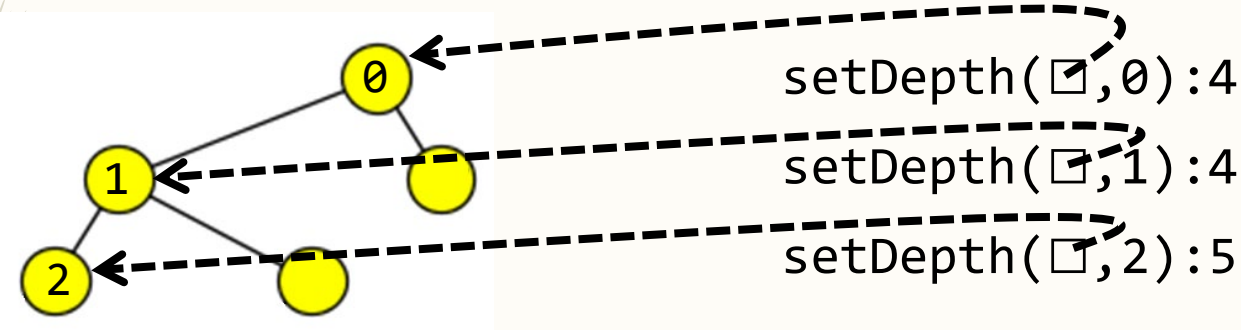

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



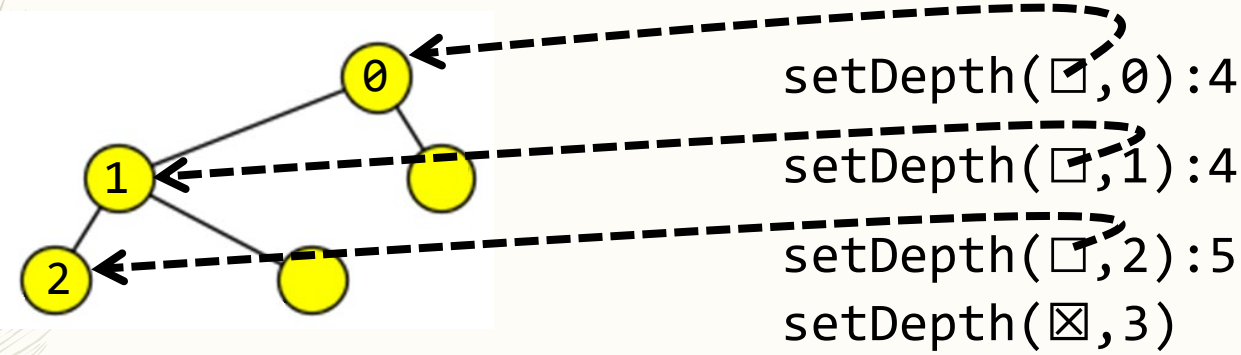

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



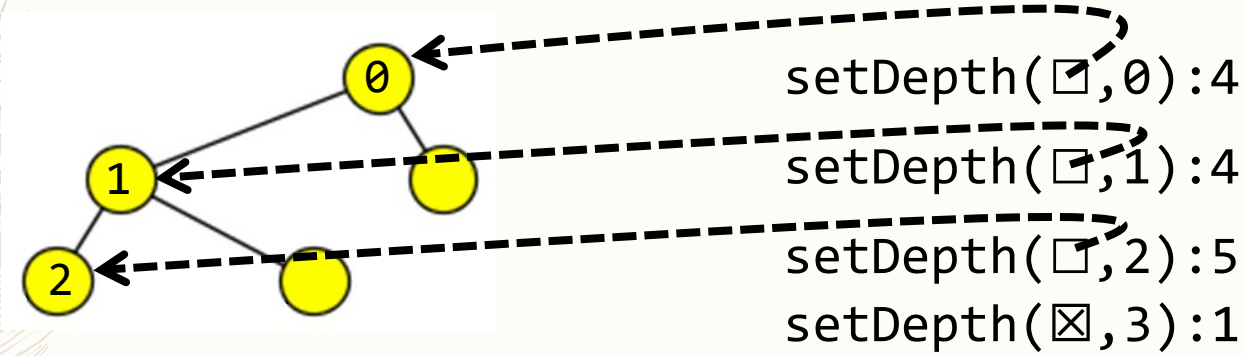

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



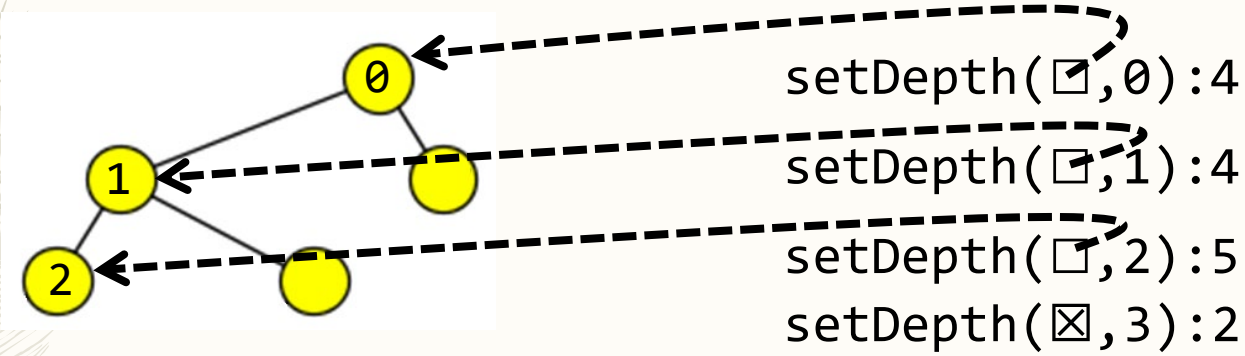

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



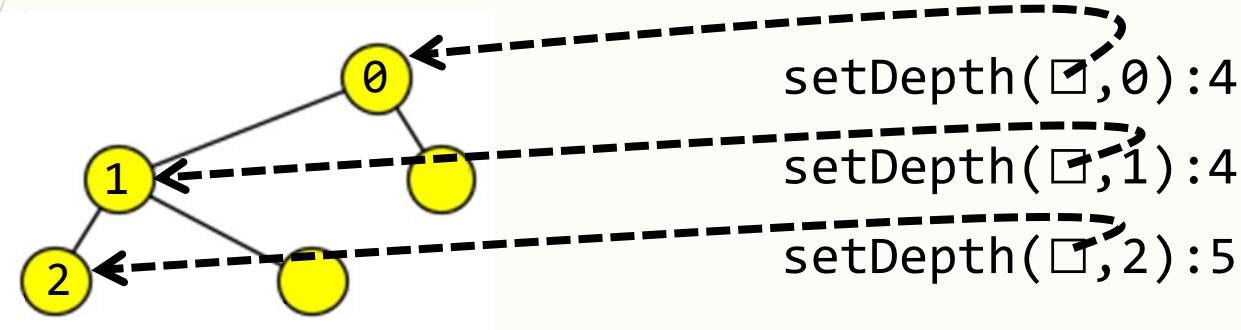

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



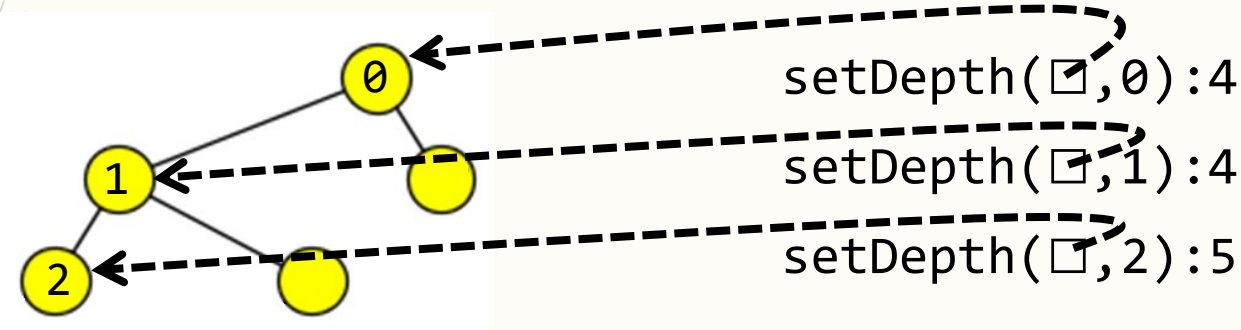

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



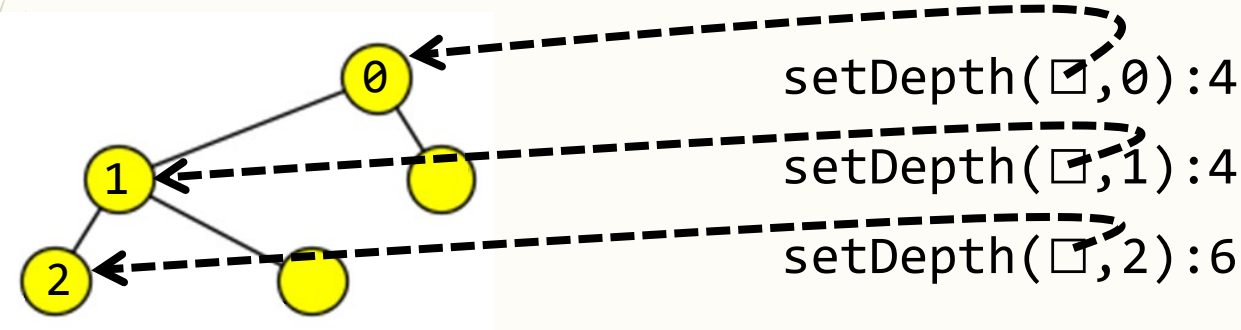

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



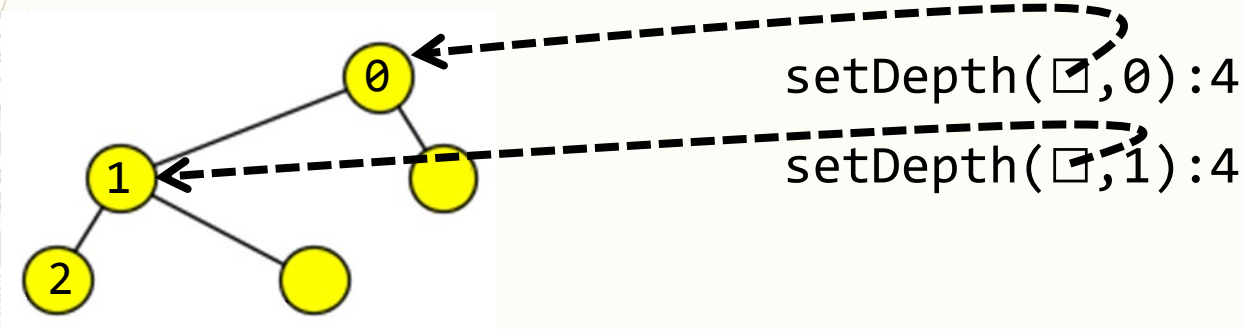

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



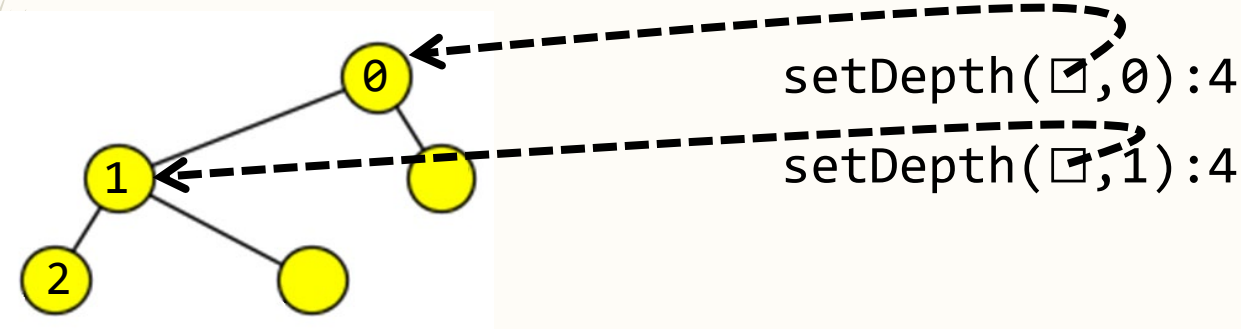

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



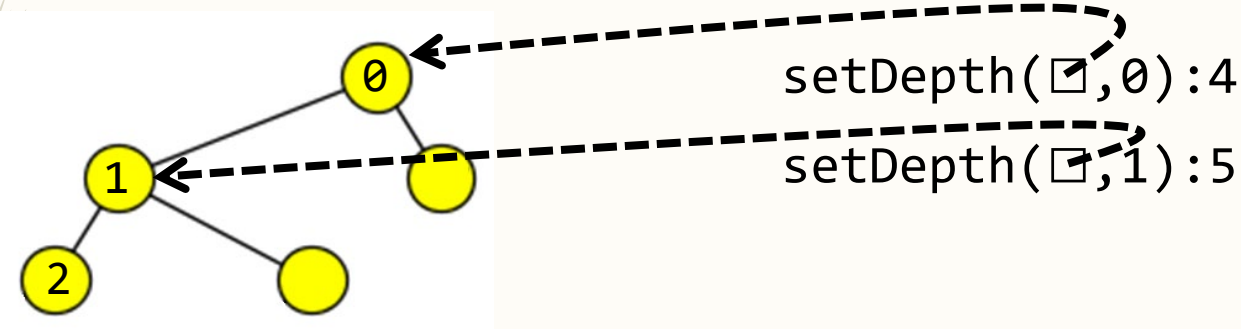

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



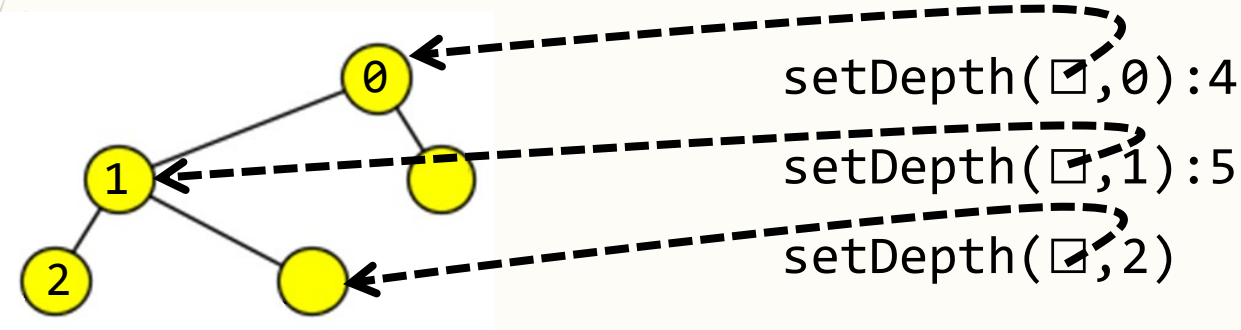

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



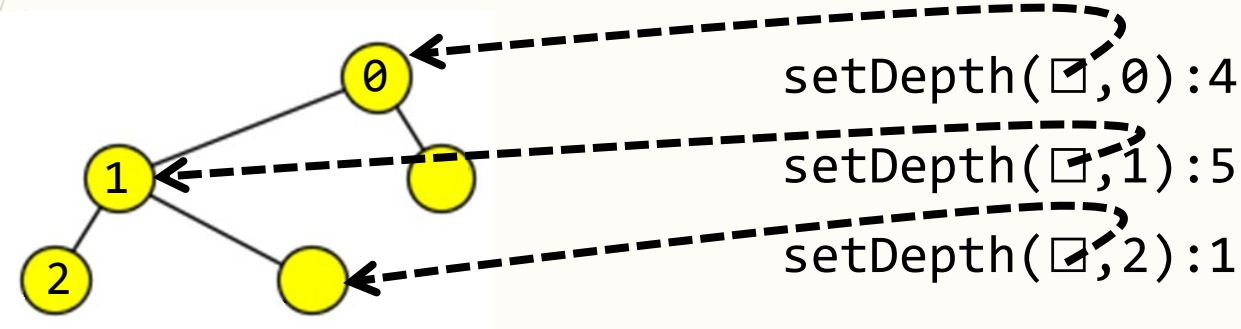

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



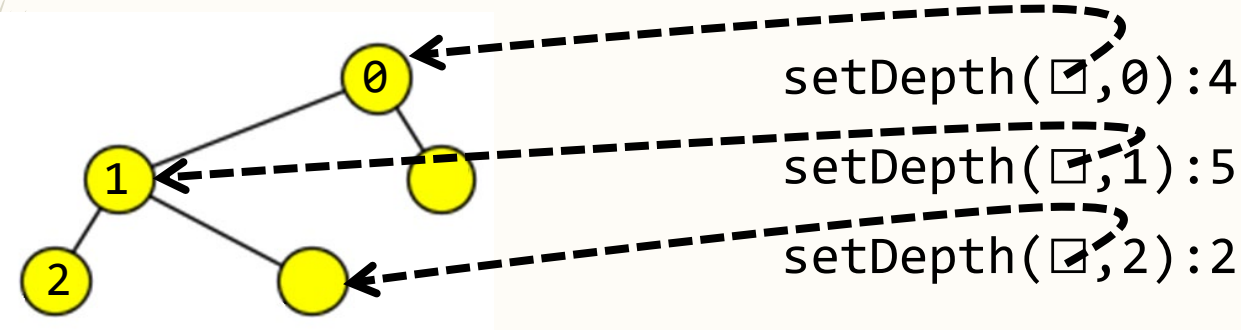

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



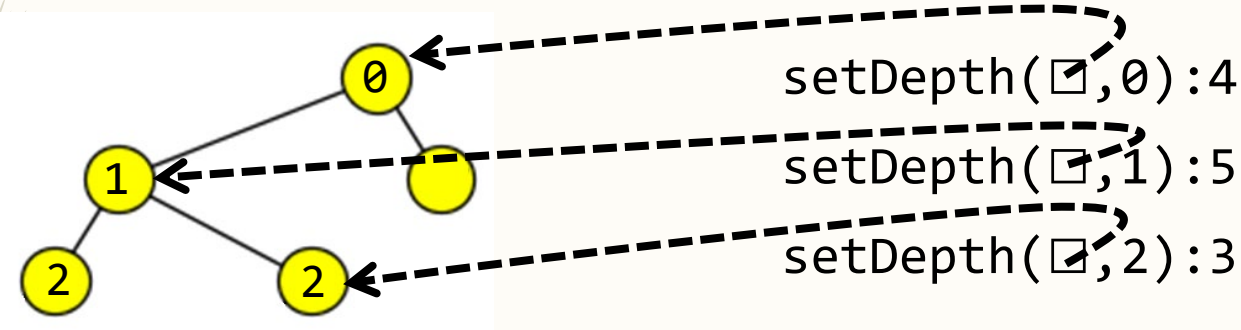

```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down




```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Down



• • •

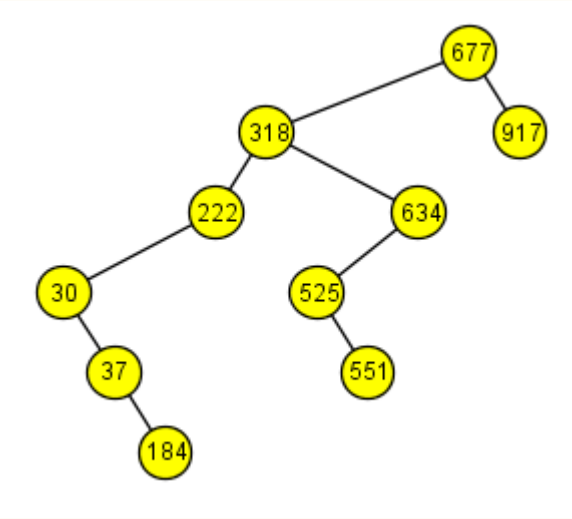
```
1: setDepth(Node t, int i) {
2:     if(t == null) return;
3:     t.depth = i;
4:     setDepth(t.left, t.depth+1);
5:     setDepth(t.right, t.depth+1);
6: }
```

Example: Pushing Info Up

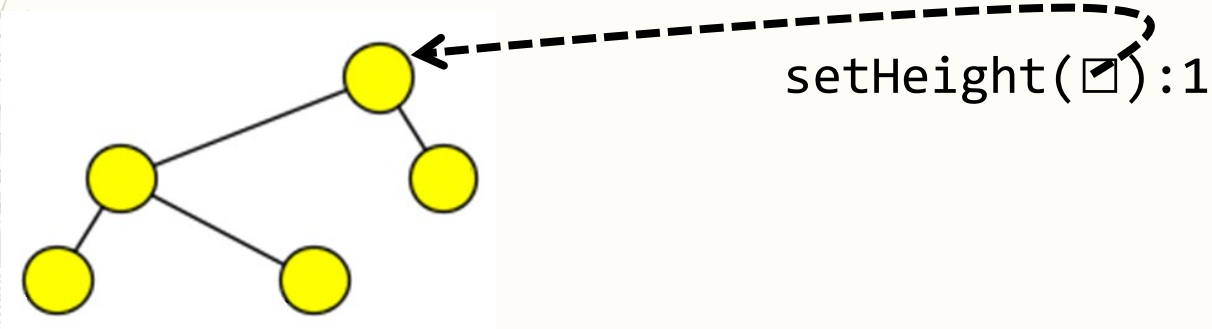
```
class Node<T> {
    T data;
    int height;
    Node<T> left, right;
}

int setHeight(Node t){
    if(t == null) return -1;
    int h1 = setHeight(t.left);
    int h2 = setHeight(t.right);
    t.height = max(h1, h2) + 1;
    return t.height;
}

setHeight(root)
```

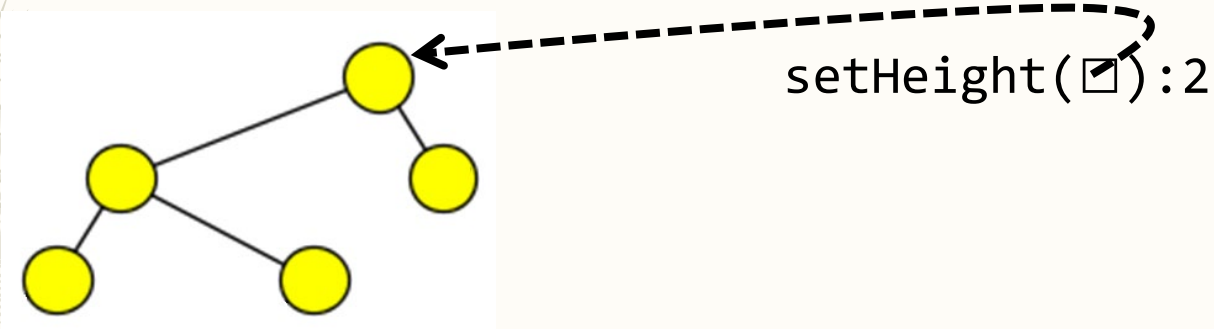


Example: Pushing Info Down



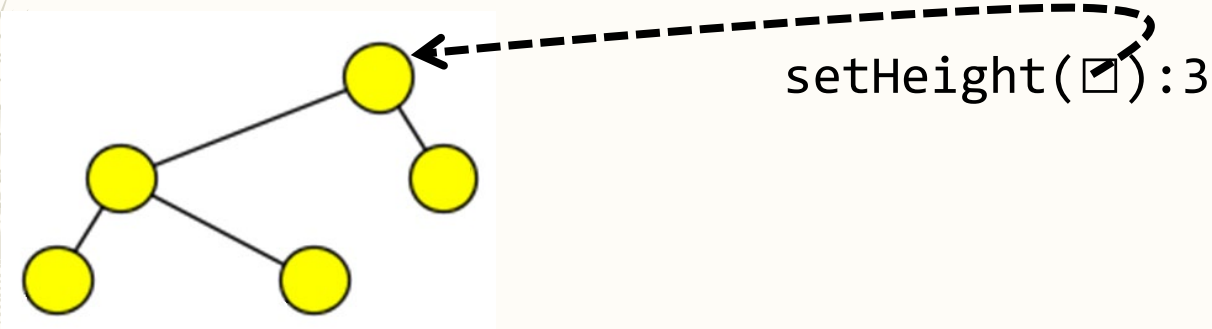

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



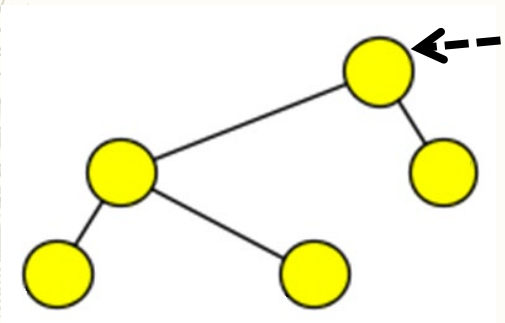

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down




```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

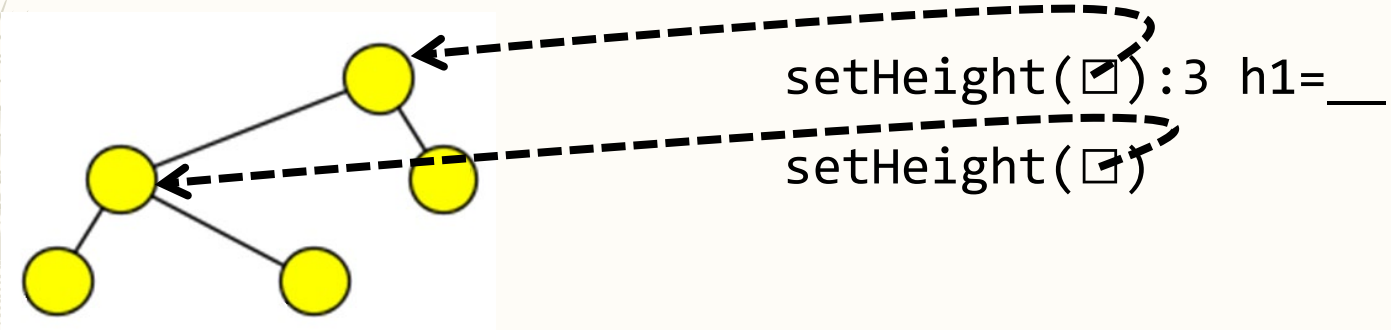
Example: Pushing Info Down



setHeight(☑):3 h1=__

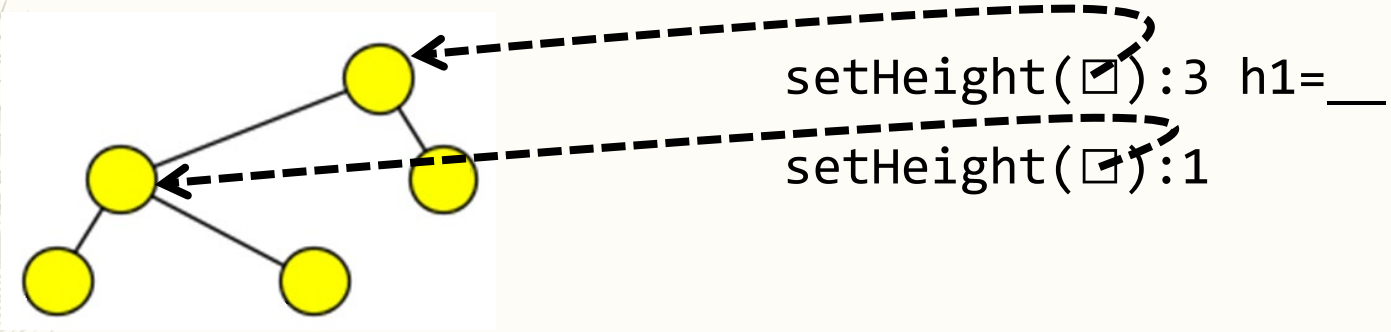
```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



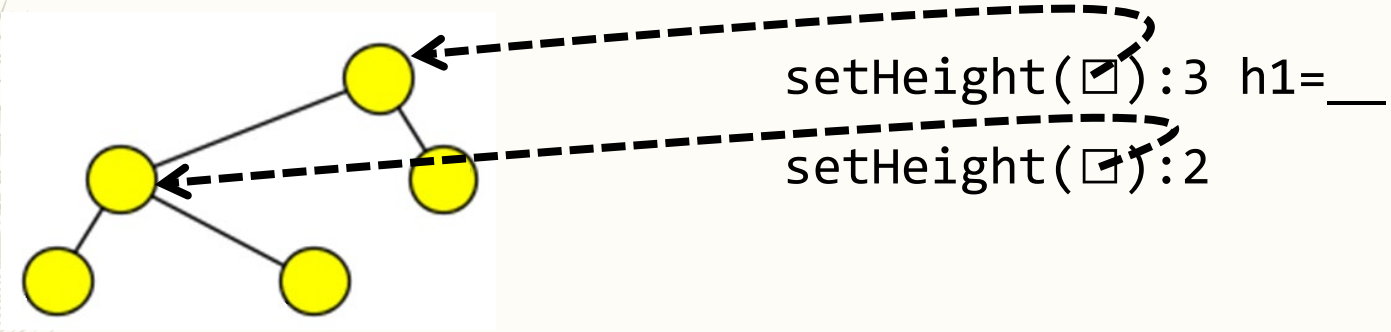

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



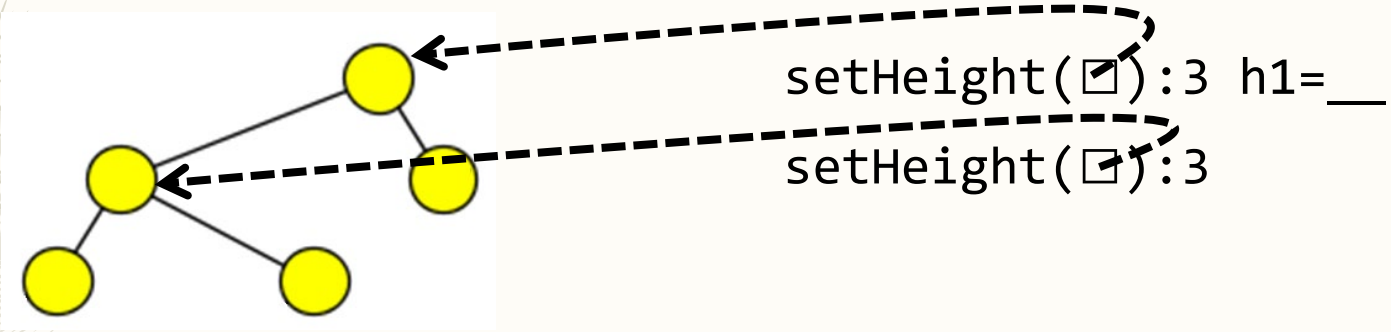

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



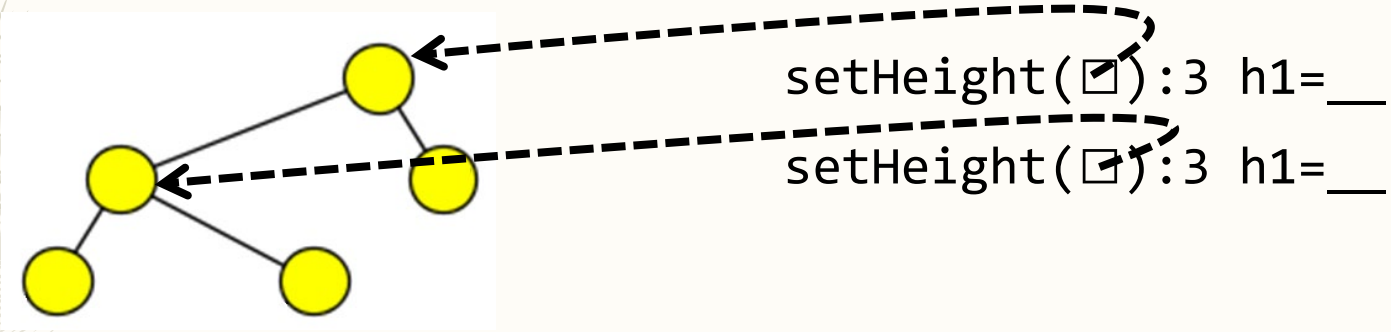

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



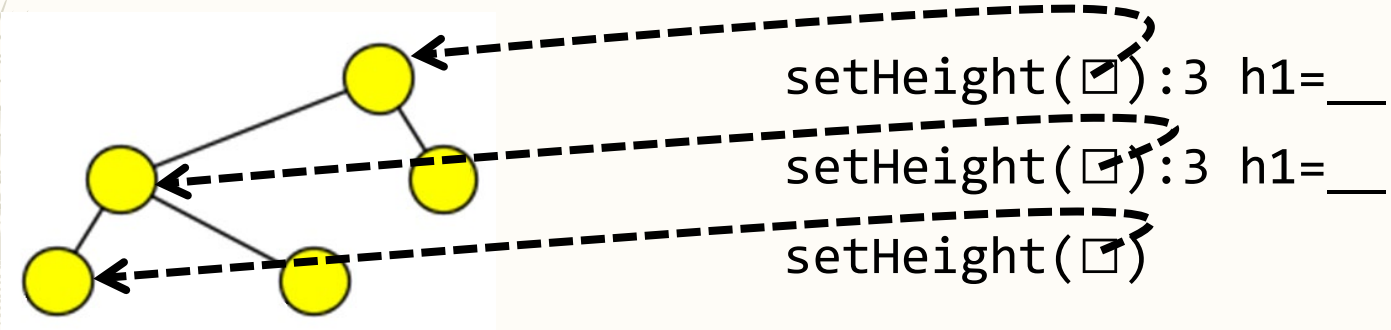

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



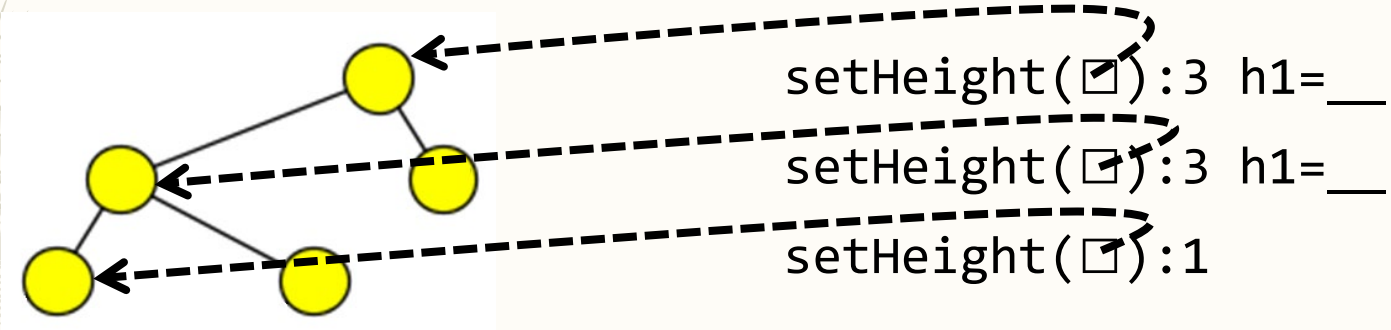

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



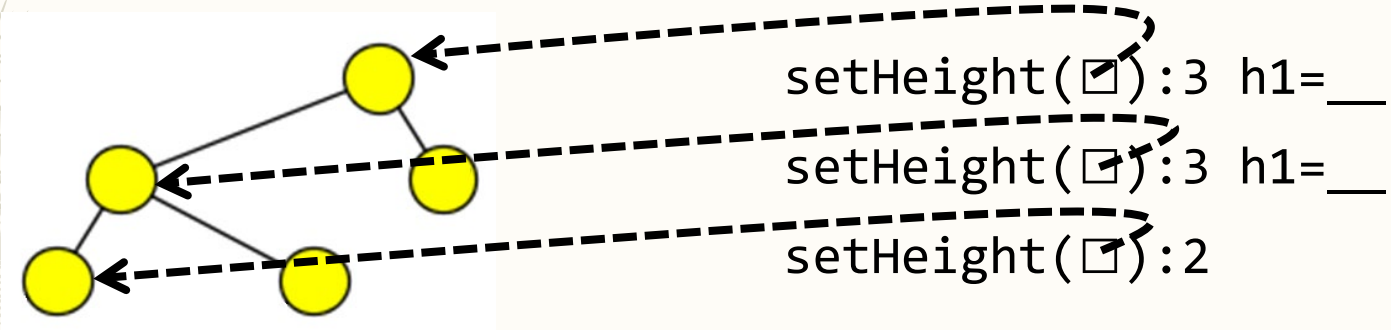

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



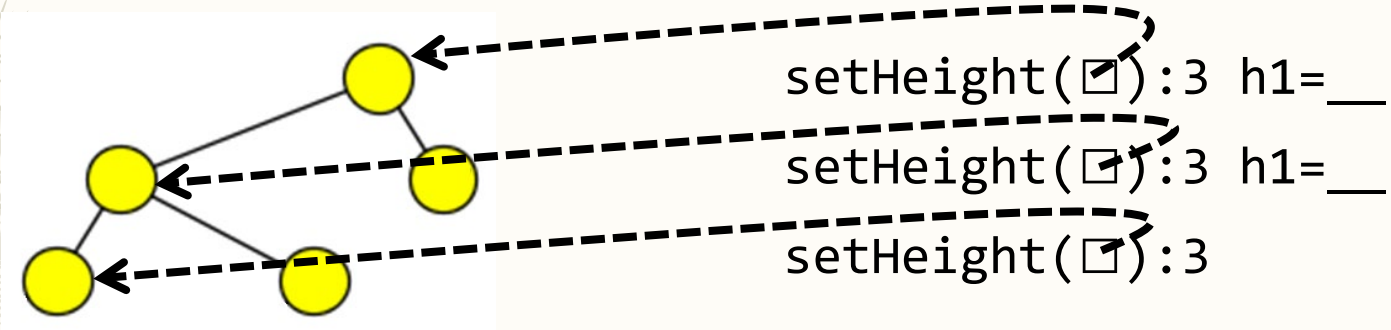

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



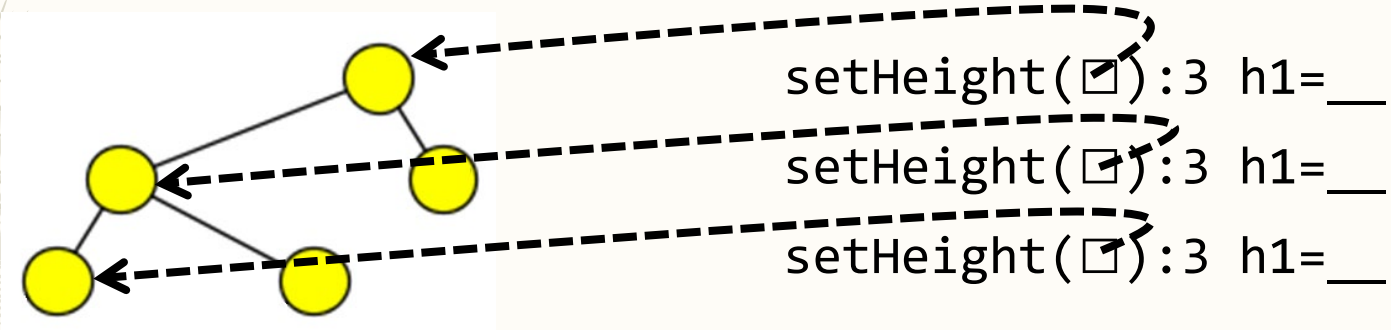

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



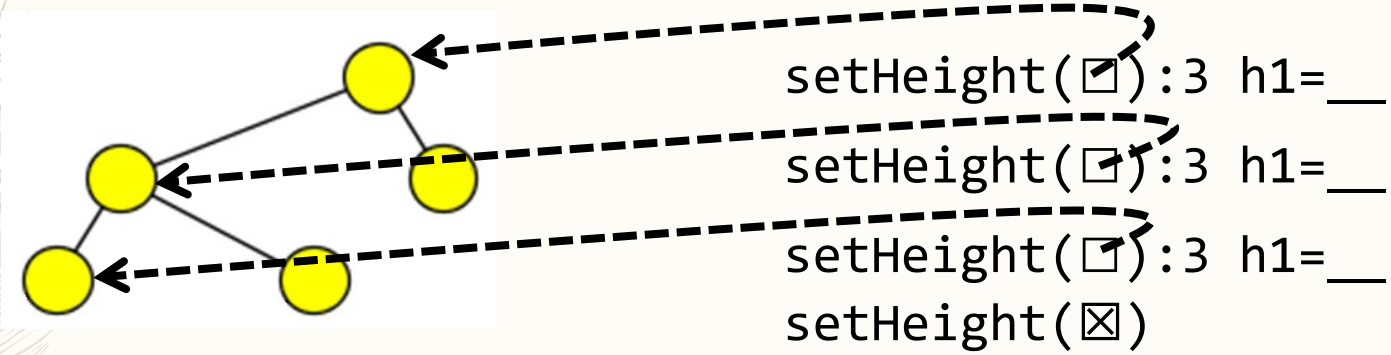

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



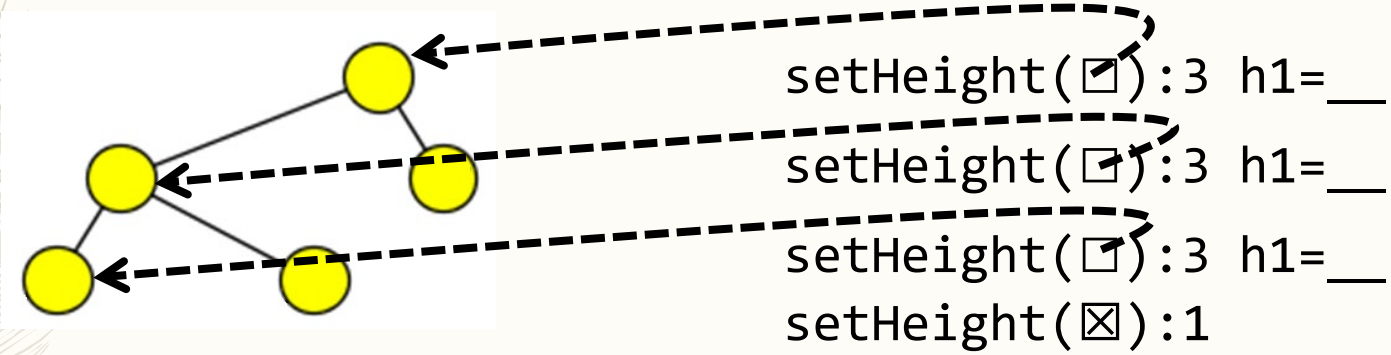

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



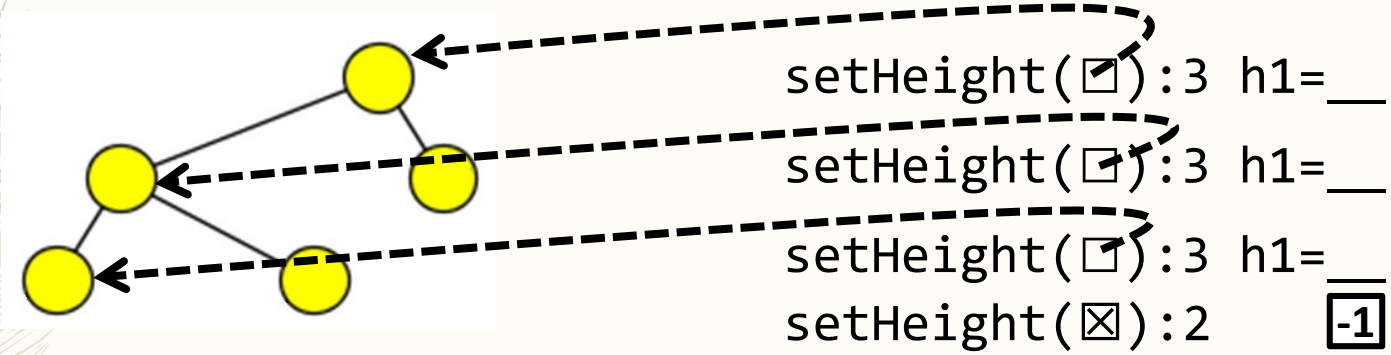

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



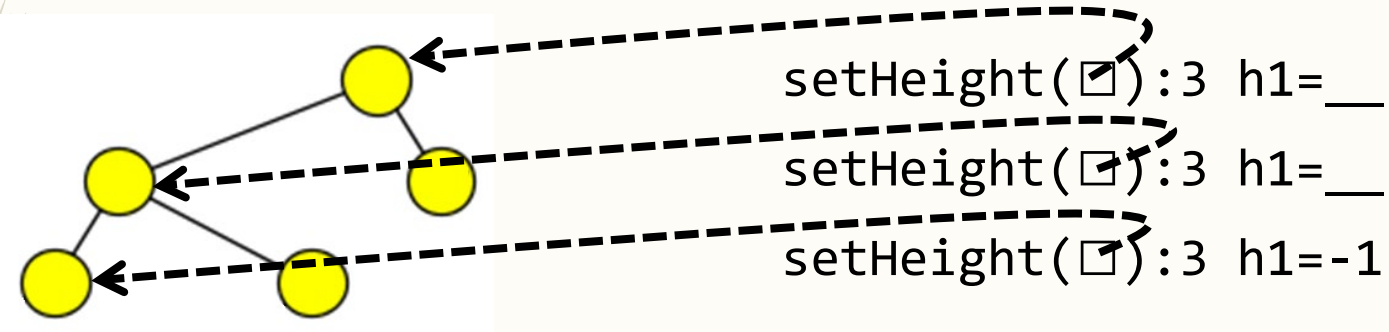

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



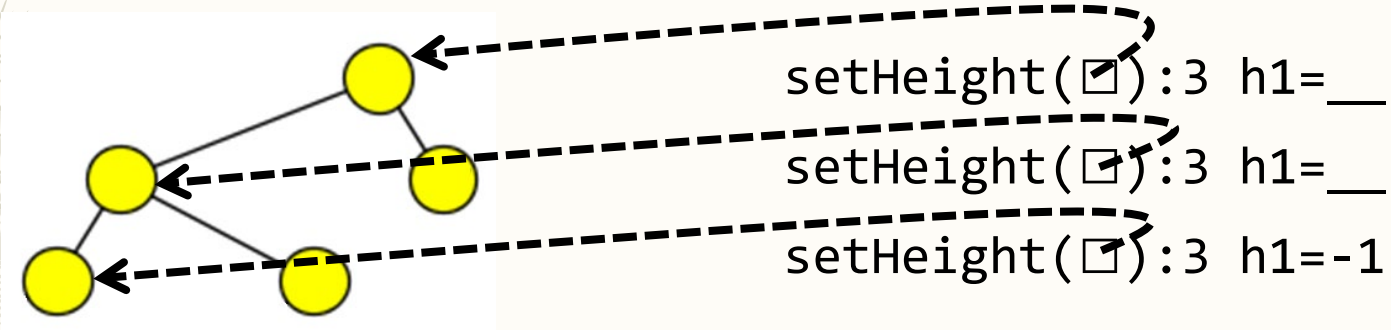

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



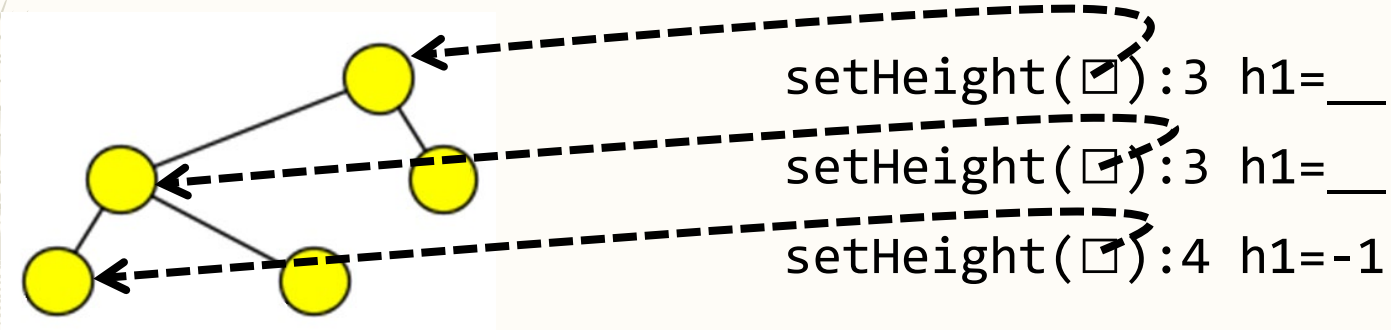

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



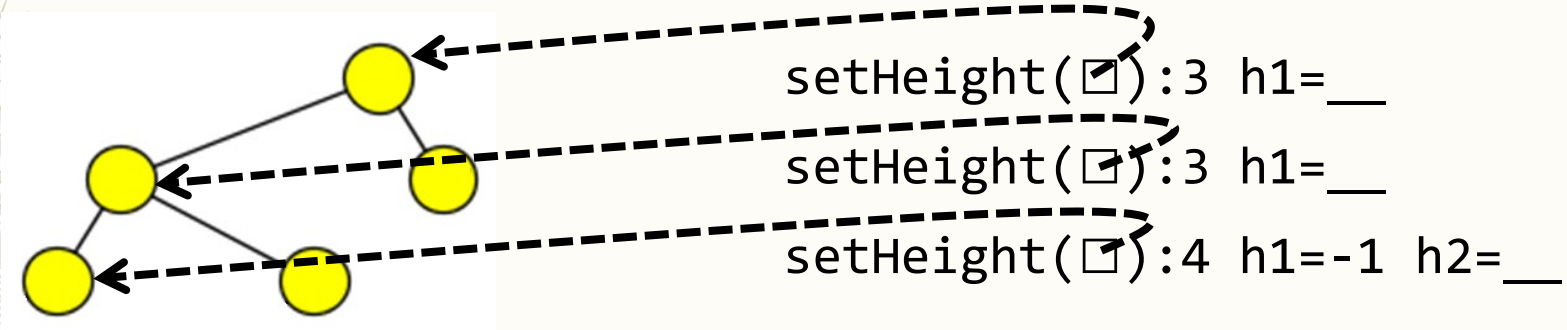

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



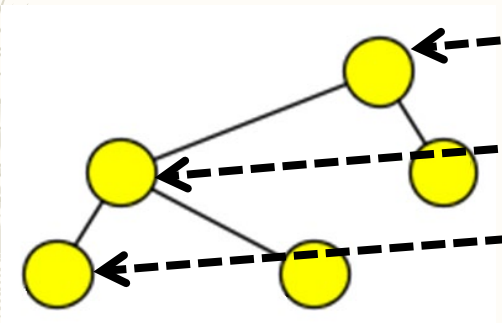

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down




```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



setHeight(☐):3 h1=__

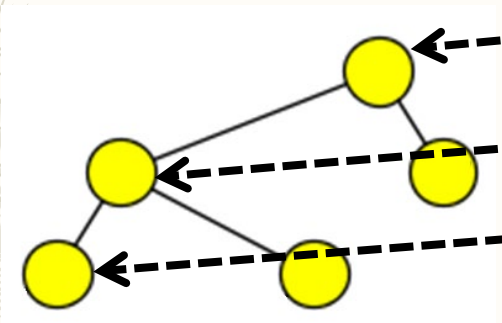
setHeight(☐):3 h1=__

setHeight(☐):4 h1=-1 h2=__

setHeight(☒)

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



setHeight(☐):3 h1=__

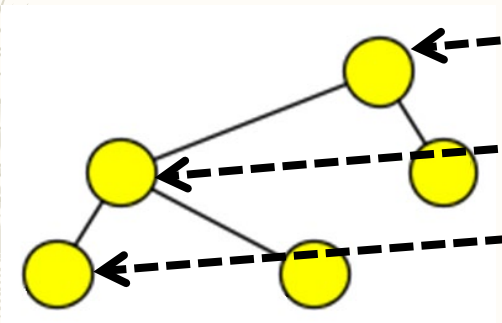
setHeight(☐):3 h1=__

setHeight(☐):4 h1=-1 h2=__

setHeight(☒):1

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



setHeight(□):3 h1=__

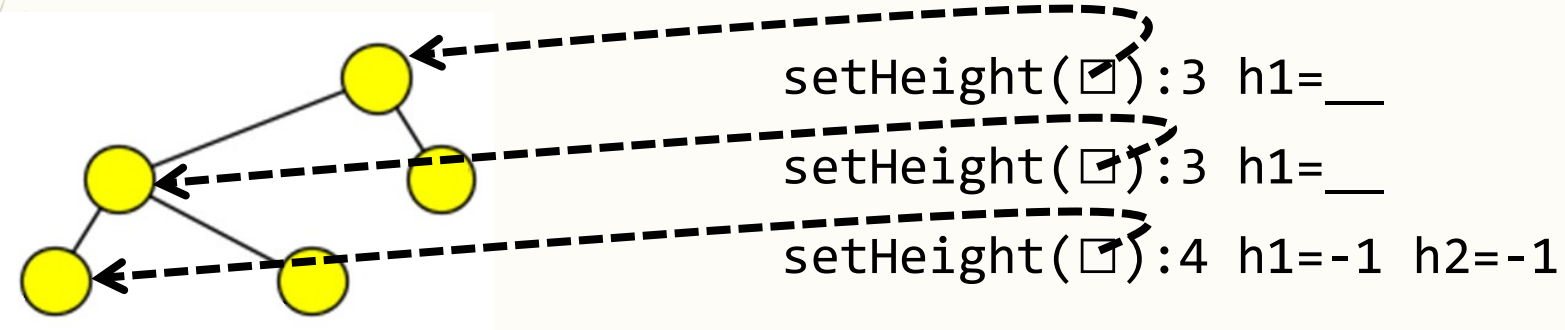
setHeight(□):3 h1=__

setHeight(□):4 h1=-1 h2=__

setHeight(☒):2 -1

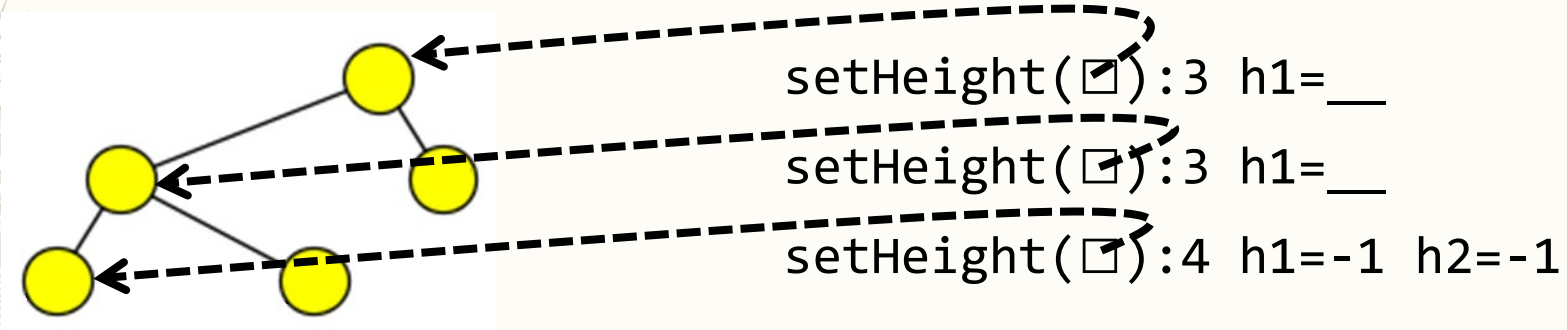
```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



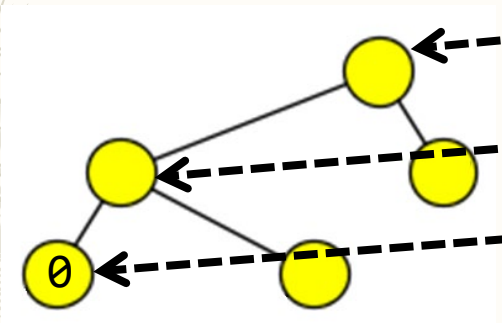

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down




```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



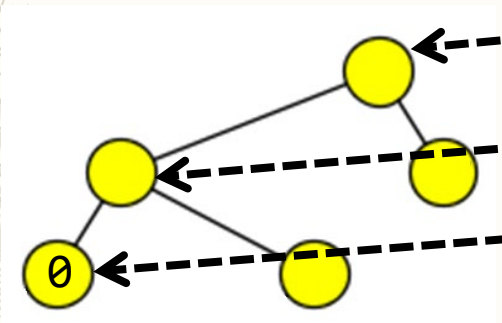
setHeight(□):3 h1=__

setHeight(□):3 h1=__

setHeight(□):5 h1=-1 h2=-1

```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



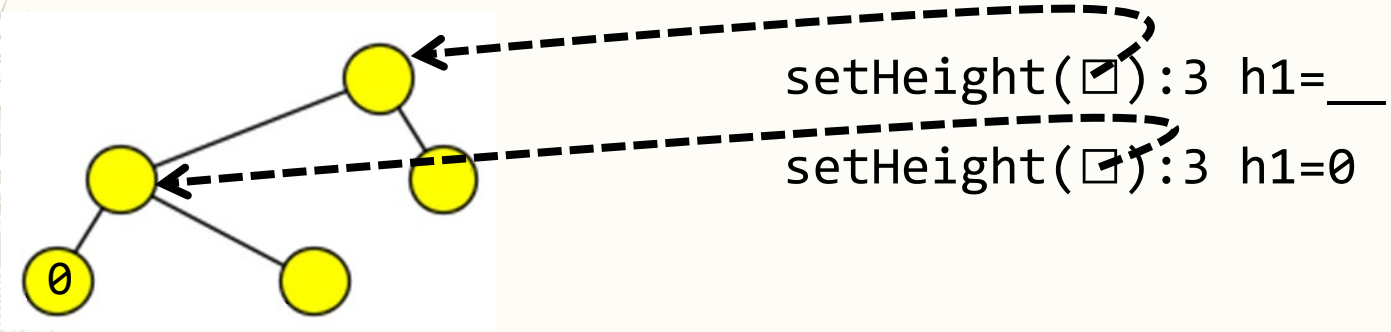
setHeight(□):3 h1=__

setHeight(□):3 h1=__

setHeight(□):6 h1=-1 h2=-1 **0**

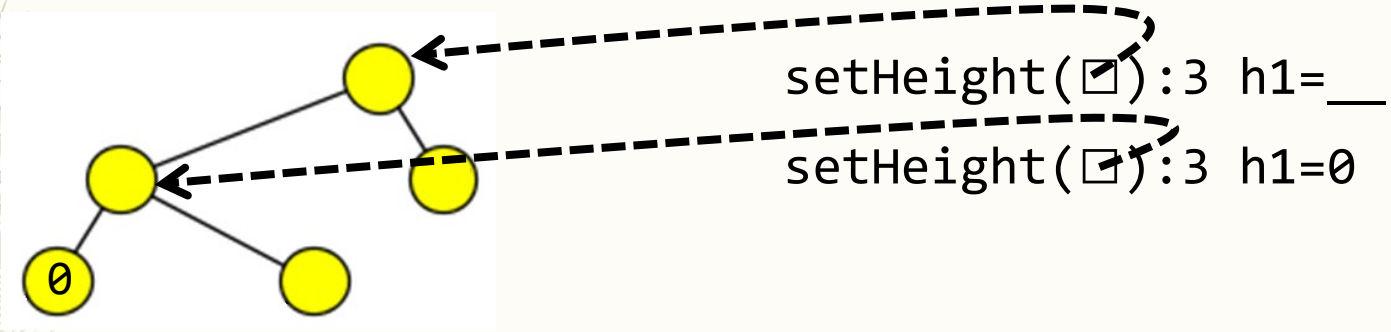
```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down




```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Down



• • •

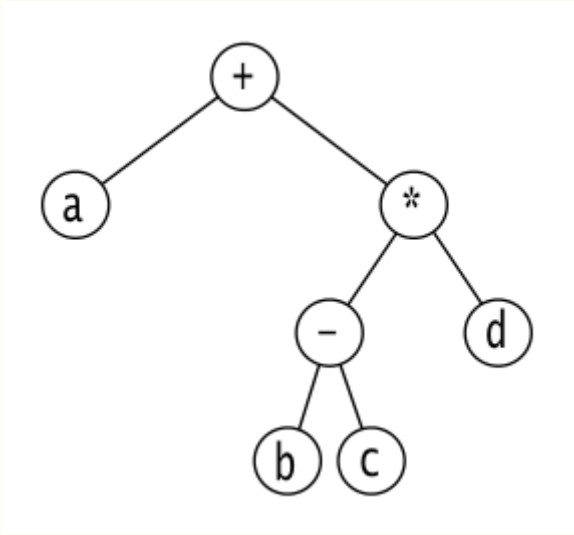
```
1: int setHeight(Node t){
2:     if(t == null) return -1;
3:     int h1 = setHeight(t.left);
4:     int h2 = setHeight(t.right);
5:     t.height = max(h1, h2) + 1;
6:     return t.height;
7: }
```

Example: Pushing Info Over

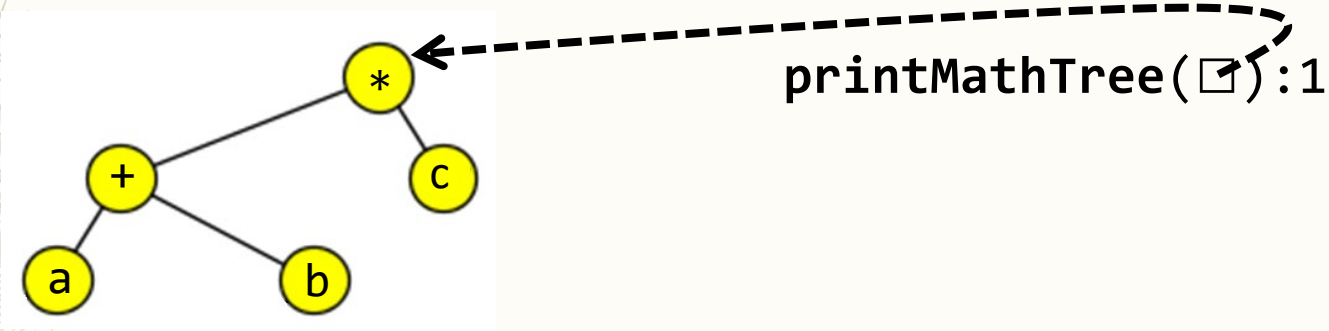
```
class Node<T> {
    T data;
    Node<T> left, right;
}

printMathTree(Node t){
    if(t == null) return;
    print("(");
    printMathTree(t.left);
    print(t.data);
    printMathTree(t.right);
    print(")");
}

printMathTree(root) //infix notation
```



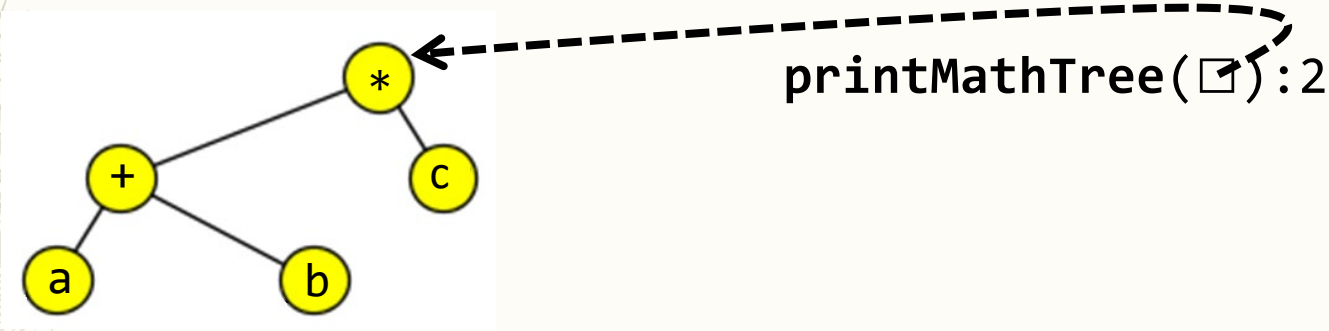
Example: Pushing Info Down



OUTPUT:

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

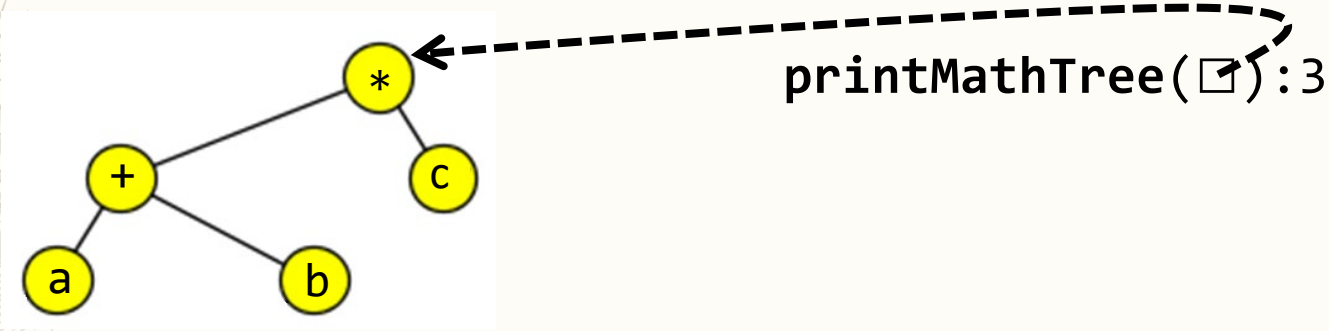
Example: Pushing Info Down



OUTPUT:

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

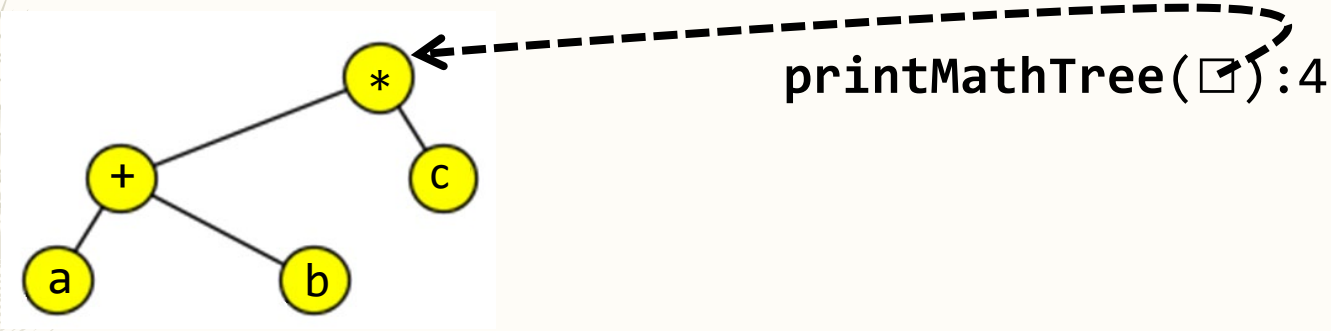
Example: Pushing Info Down



OUTPUT: (

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

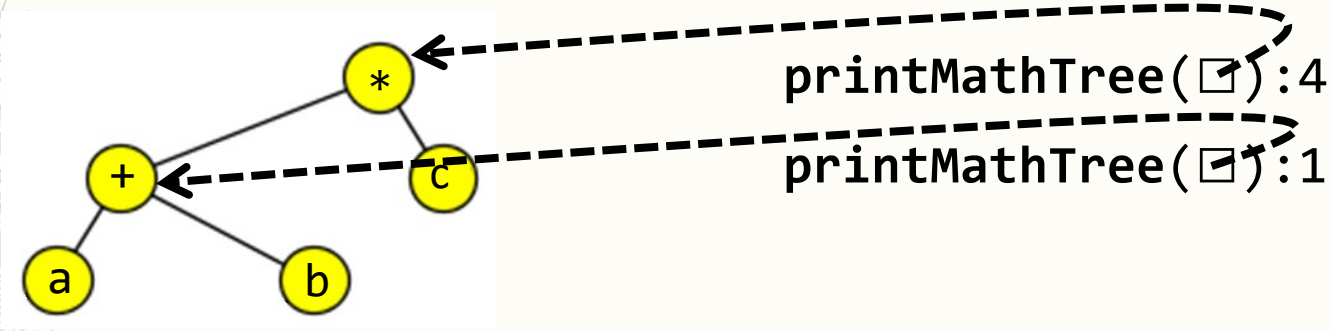
Example: Pushing Info Down



OUTPUT: (

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

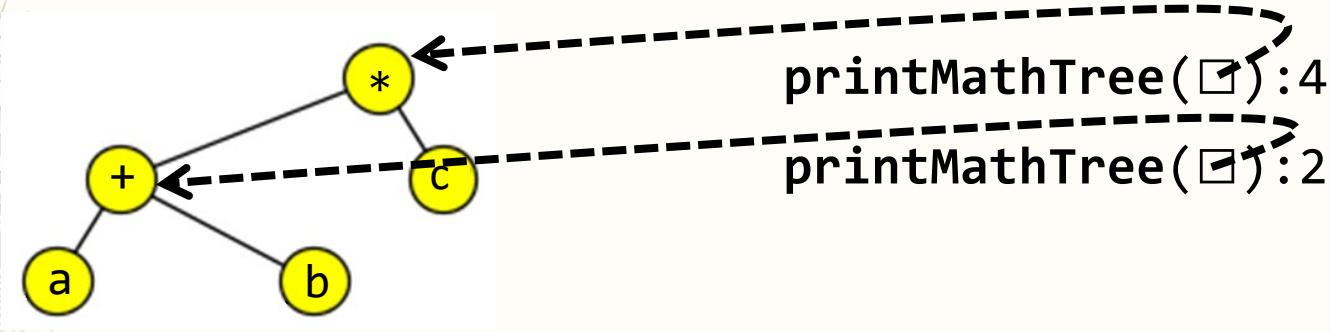
Example: Pushing Info Down



OUTPUT: (

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

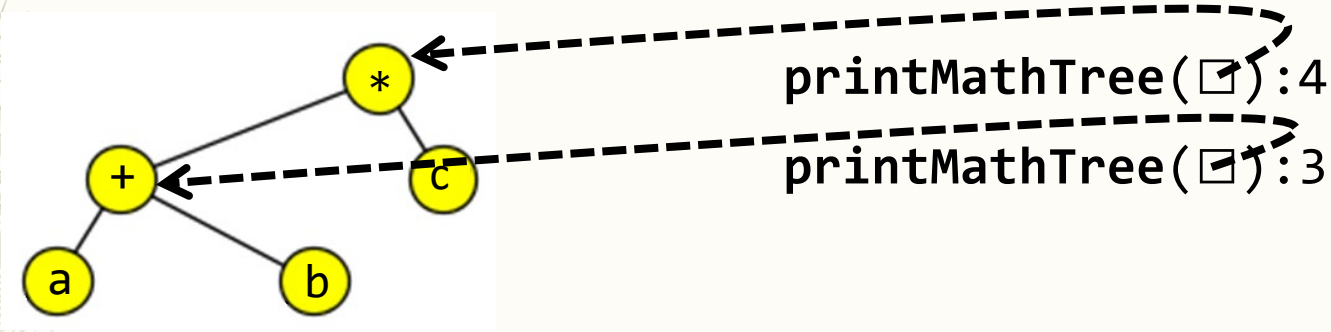
Example: Pushing Info Down



OUTPUT: (

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

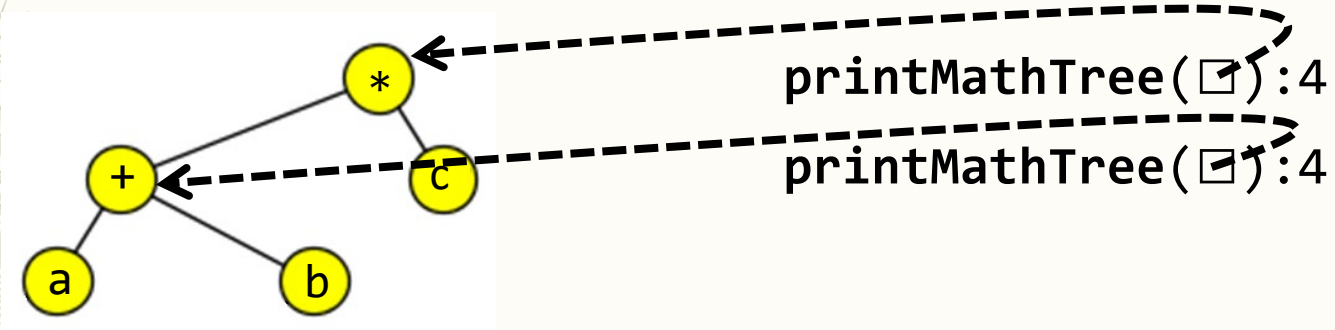
Example: Pushing Info Down



OUTPUT: ((

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

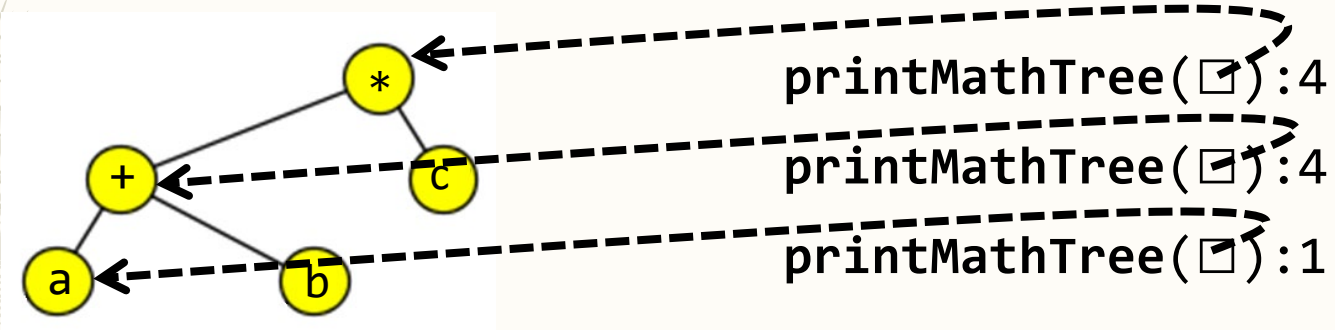
Example: Pushing Info Down



OUTPUT: ((

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

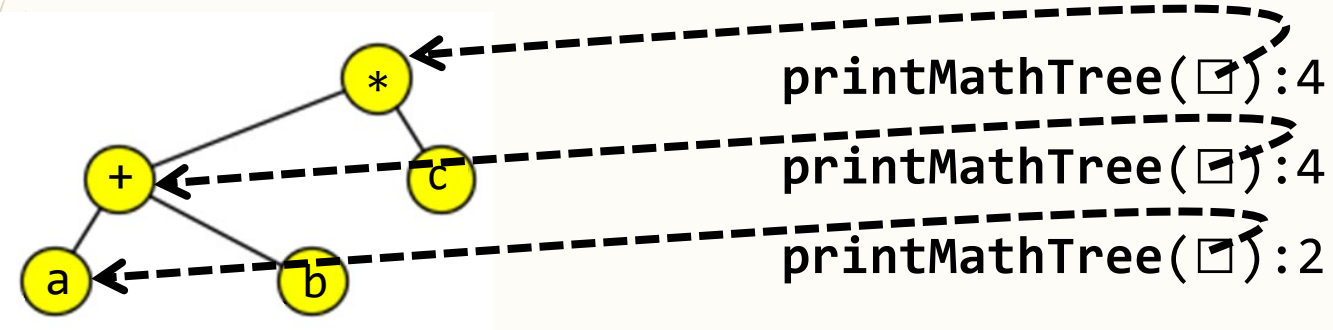
Example: Pushing Info Down



OUTPUT: ((

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

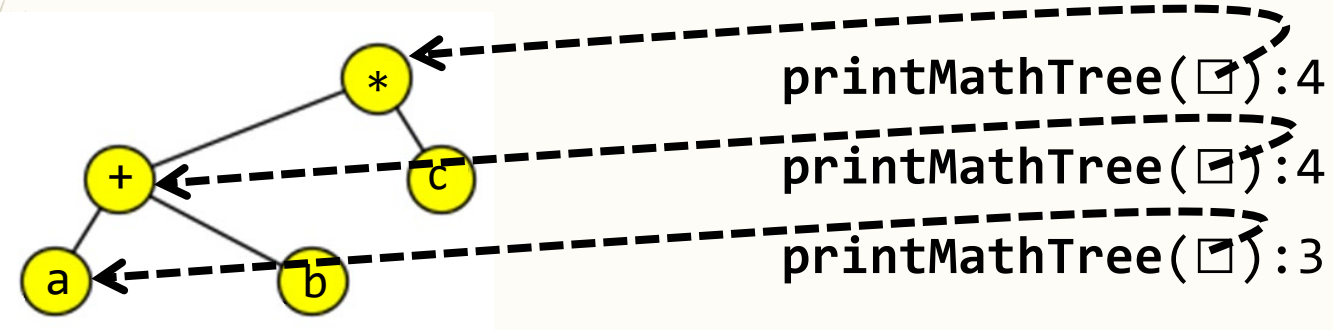
Example: Pushing Info Down



OUTPUT: ((

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

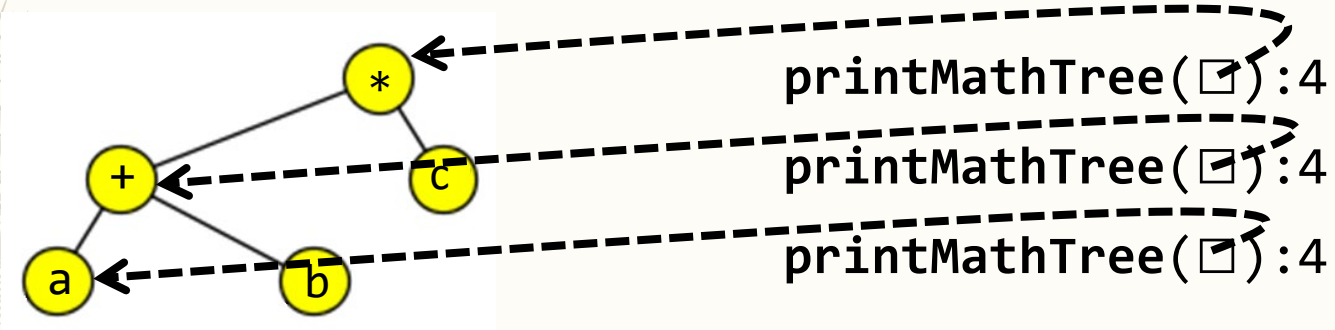
Example: Pushing Info Down



OUTPUT: (((

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

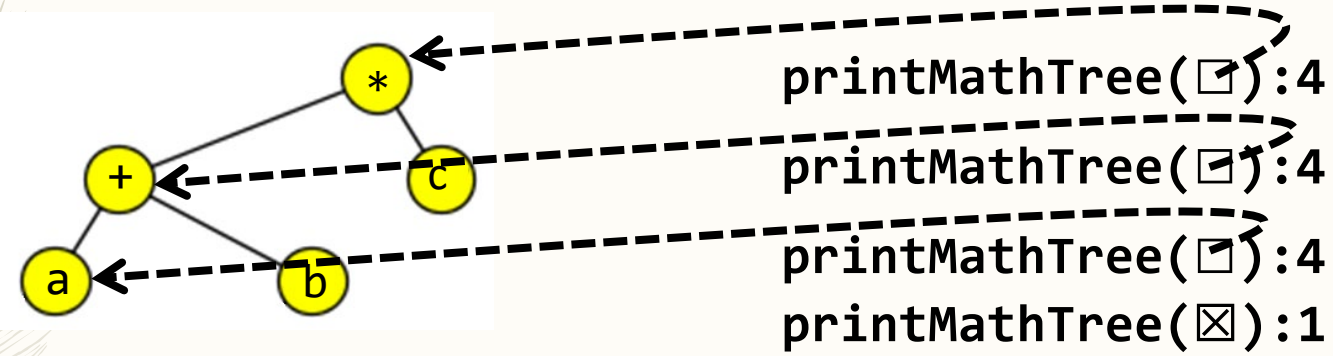
Example: Pushing Info Down



OUTPUT: (((

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

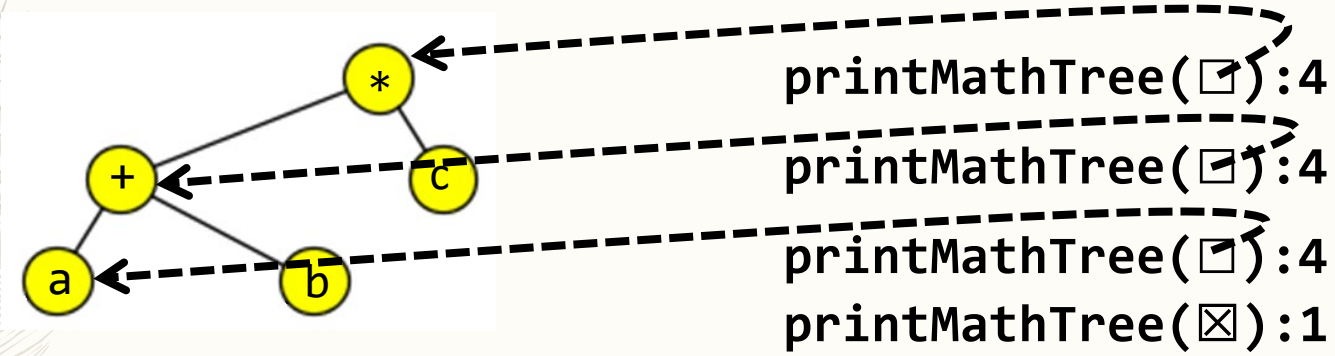
Example: Pushing Info Down



OUTPUT: (((

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

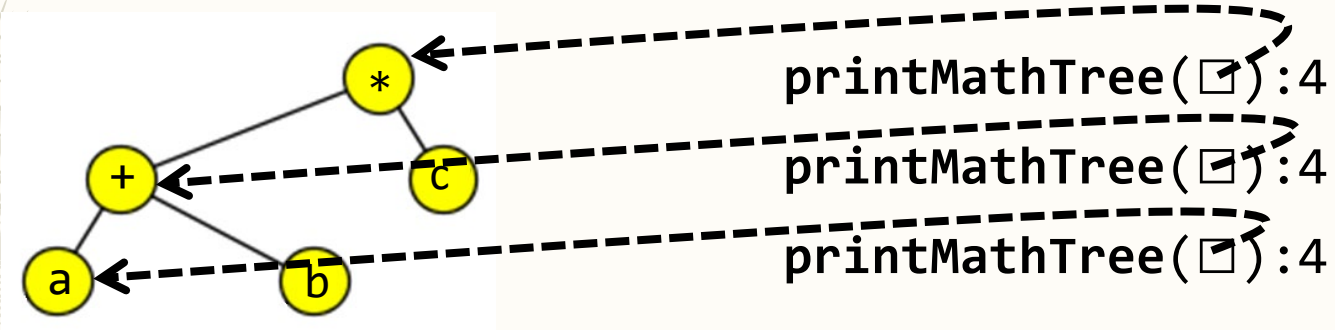
Example: Pushing Info Down



OUTPUT: (((

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

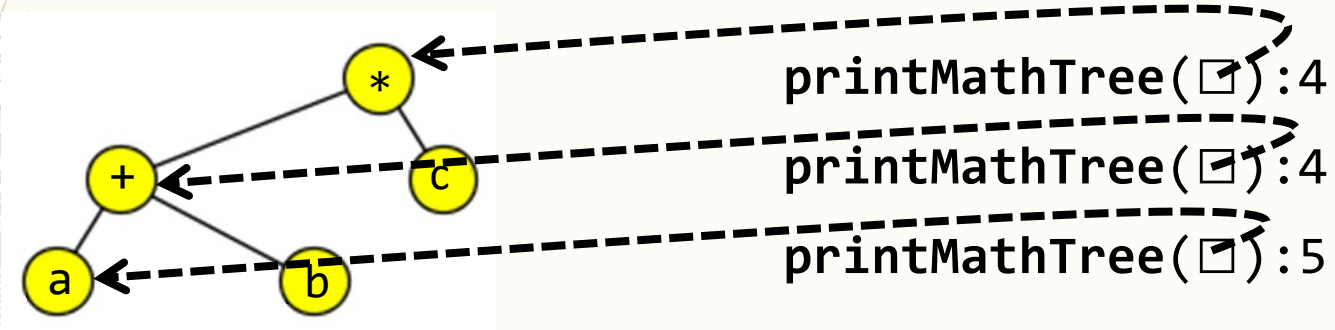
Example: Pushing Info Down



OUTPUT: (((

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

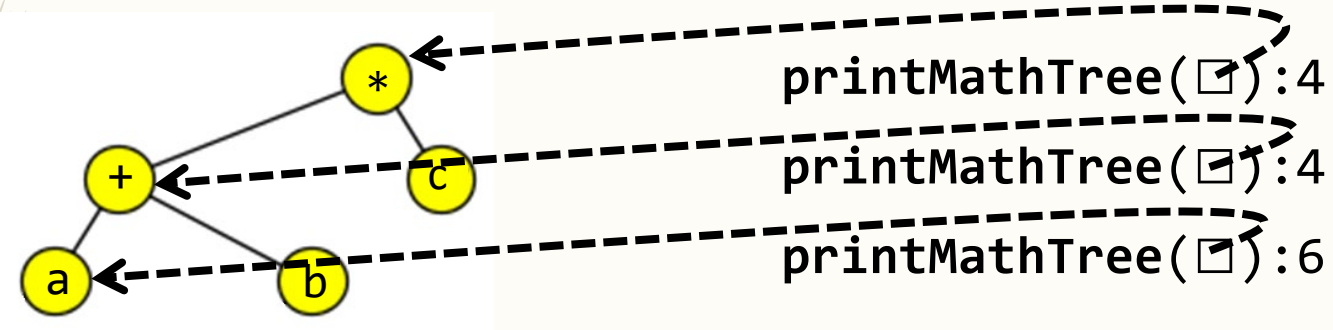
Example: Pushing Info Down



OUTPUT: (((a

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

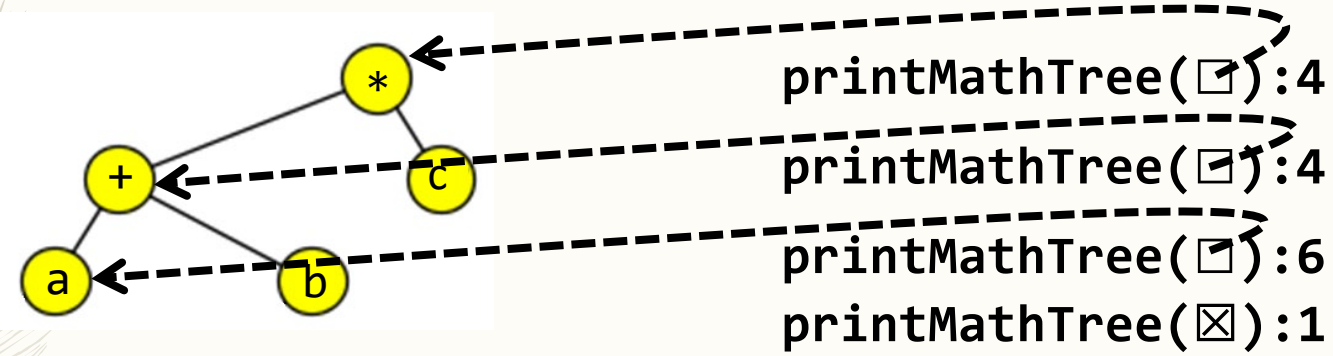
Example: Pushing Info Down



OUTPUT: (((a

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

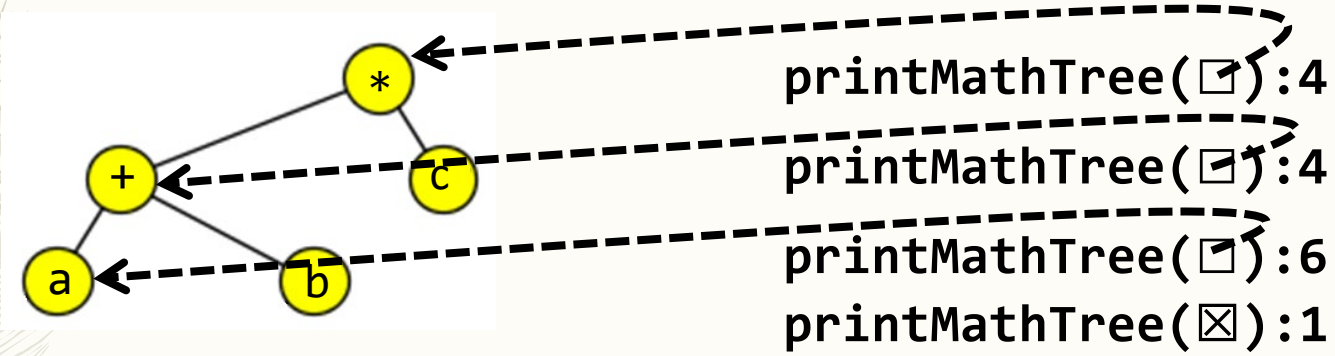
Example: Pushing Info Down



OUTPUT: (((a

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

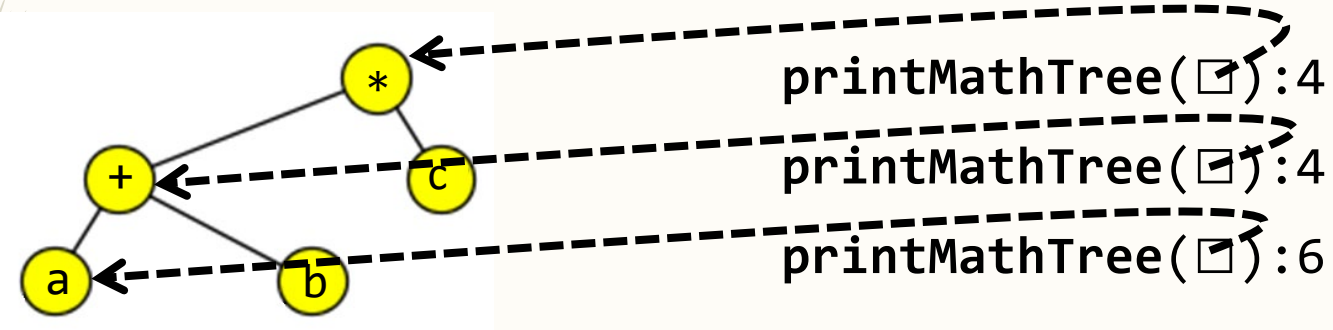
Example: Pushing Info Down



OUTPUT: (((a

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

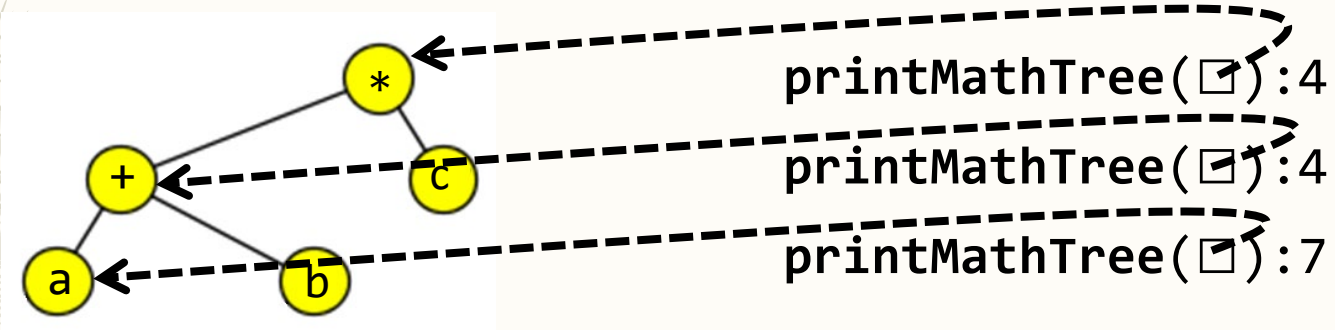
Example: Pushing Info Down



OUTPUT: (((a

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

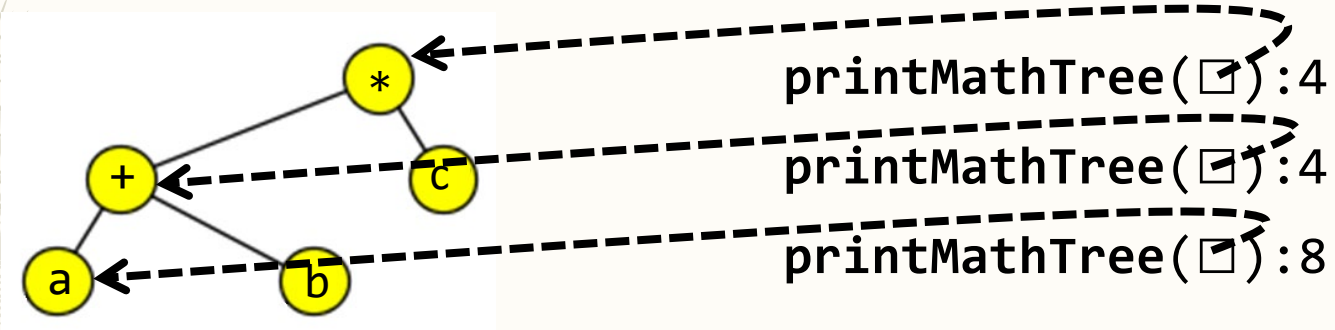
Example: Pushing Info Down



OUTPUT: (((a)

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

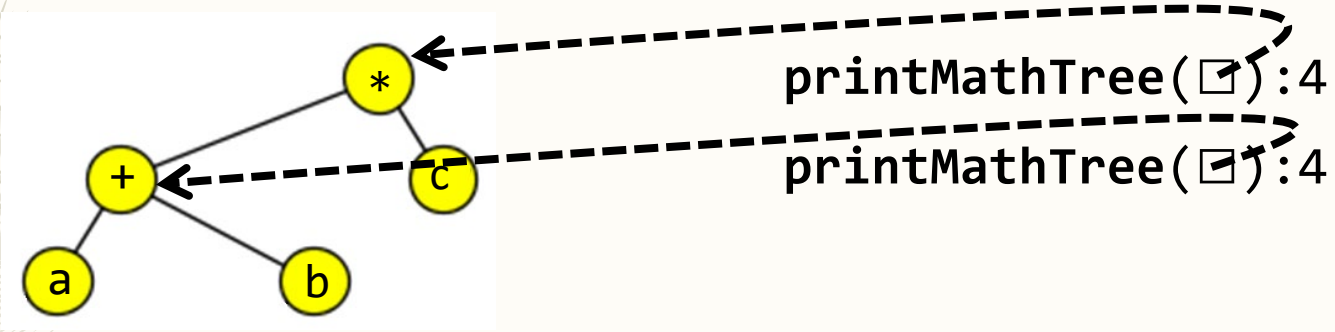
Example: Pushing Info Down



OUTPUT: (((a)

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

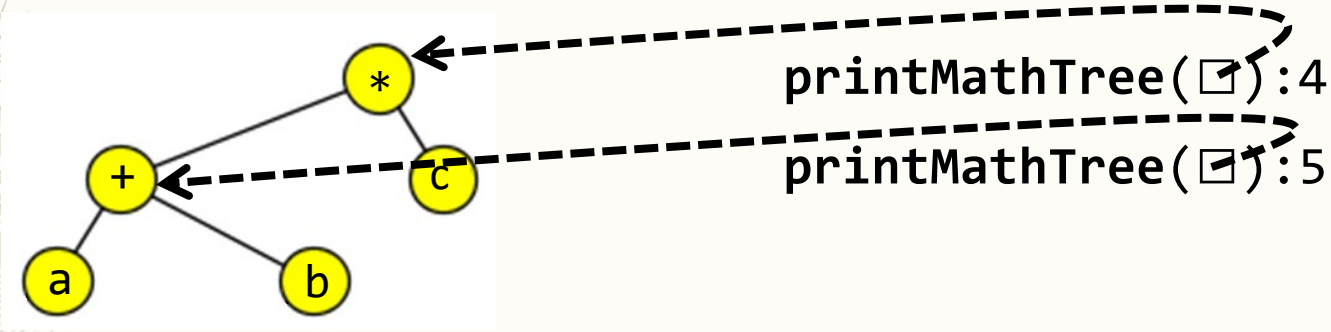
Example: Pushing Info Down



OUTPUT: (((a)

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

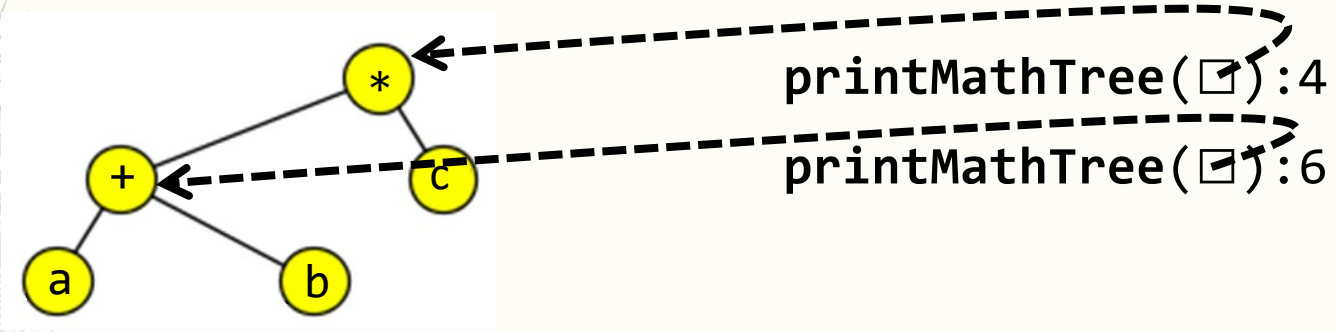
Example: Pushing Info Down



OUTPUT: (((a)+

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

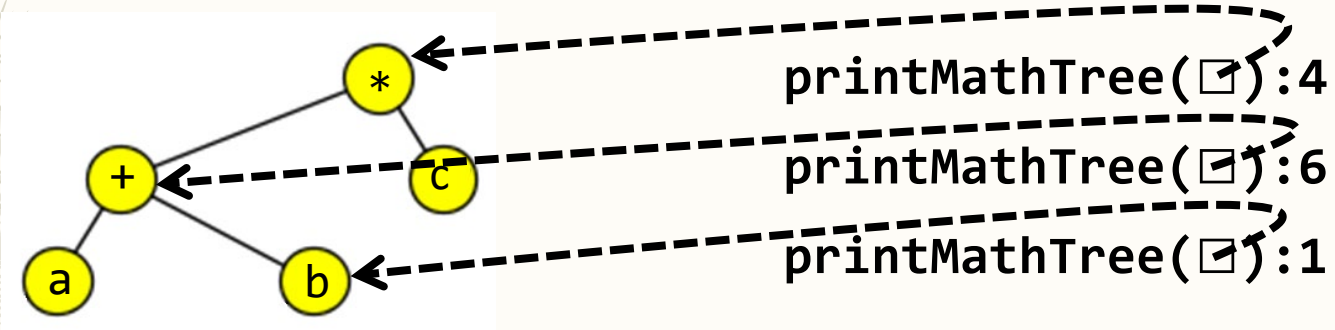
Example: Pushing Info Down



OUTPUT: (((a)+

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```

Example: Pushing Info Down



OUTPUT: (((a)+ . . .

```
1: printMathTree(Node t) {
2:     if(t == null) return;
3:     print("(");
4:     printMathTree(t.left);
5:     print(t.data);
6:     printMathTree(t.right);
7:     print(")");
8: }
```