# Data Structures
# Chapter 5 Tree

1. introduction
2. Binary tree
   - Definition and Properties
   - Traversal
   - **Coding - Quizzes**
3. Binary search tree
4. Tree balancing

*Prof. Youngsup Kim, idebtor@gmail.com, CSEE Dept., Grace School Rm204, Handong Global University*

On my honor, I pledge that I have neither received nor provided improper assistance in the completion of this assignment.
서명: ***강동인*** 학번: 21500002___

# Operations: size()

```
// returns the number of nodes in the binary tree
int size(tree node) {
  if (empty(node)) return 0;
  return size(node->left) + size(node->right) + 1;
}
```

- Q1. What is the total number of the function calls to complete with the tree and how many returns each?

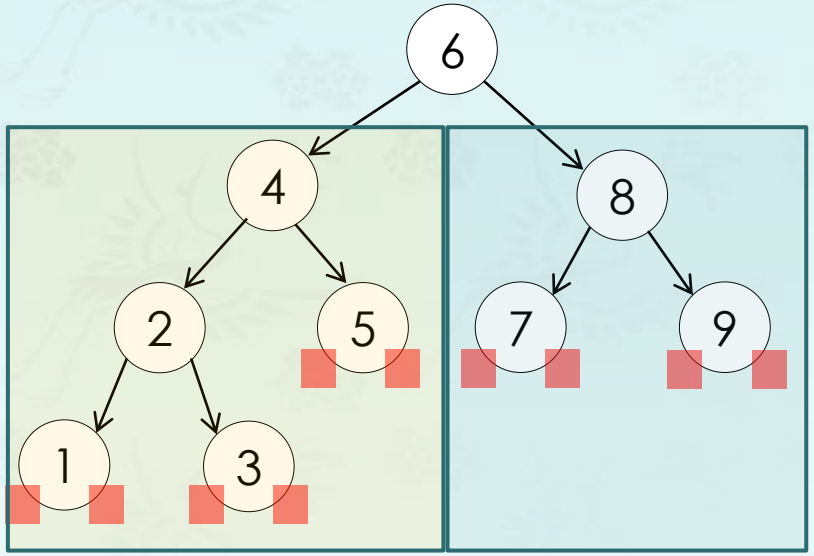
  **17 (return 0 * 8 + return size * 9)**

- Q2. Which node invokes the last function call?

  **Node 9**

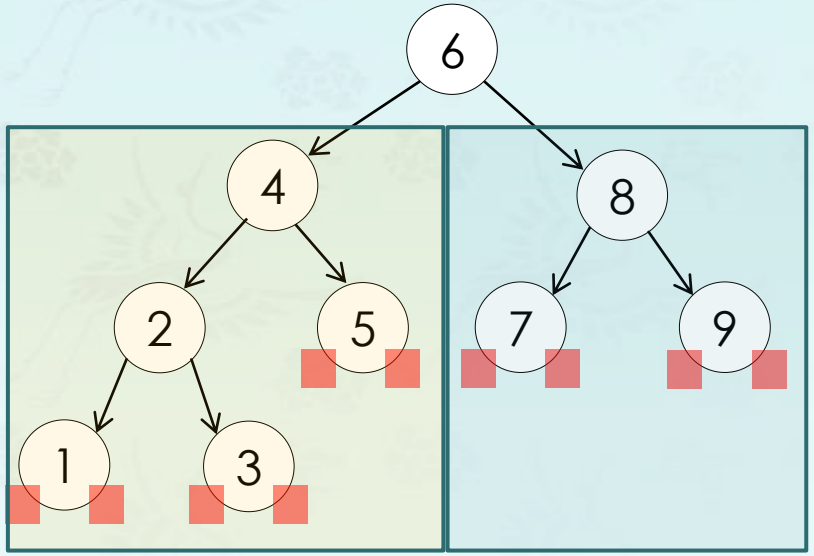
- Q3. Which node finishes its size function call and returns size = 1 for the first time?

  **Node 1**



# Operations: height()

```
// returns the max depth of a tree.
// height = -1 for empty tree, 0 for root only tree
int height(tree node) {
  if (empty(node)) return -1;
  int left  = height(node->left);
  int right = height(node->right);
  return max(left, right) + 1;
}
```




- Q1. What is the total number of the function call to complete with the tree below?

  **17 (return -1 * 8 + return height * 9)**

- Q2. What is the return value of the 10th and 12th function call?

  **-1, 1 (node 5's left empty node, node 8)**

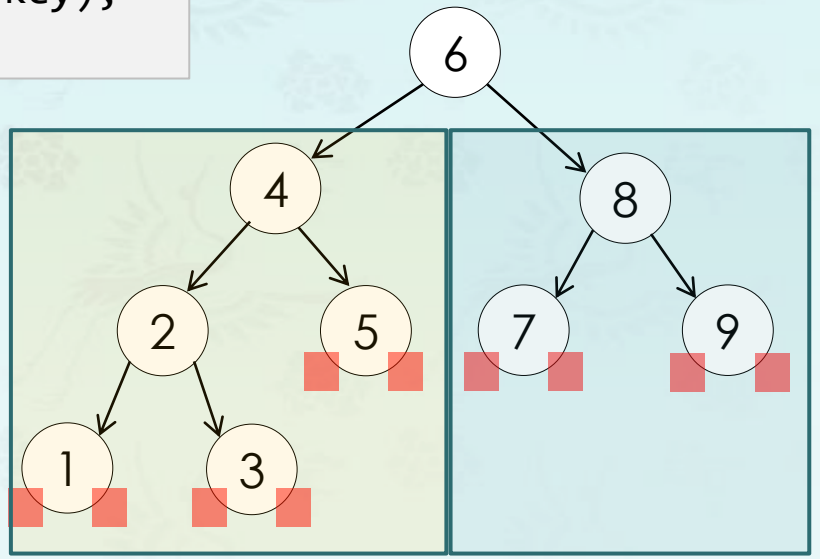
- Q3. What is the return value of the node 2?

  **1**

# Operations: containsBT(), findBT()

```
// returns true if key is in a given binary tree, false otherwise.
bool containsBT(tree root, int key) {
  if (empty(root)) return false;
  if (key == root->key) return true;

  return containsBT(root->left, key) || containsBT(root->right, key);
}
```




- Q1: Which node invokes **containsBT(root->right, key)** for the first time?

  **Node 1**

- Q2: Which node will invoke **return false** for the first time?

  **Node 1**

- Q3: How many function calls are made to reach the node **key=5**?

  **10**

- Q4: How many function calls still remains in the system stack to finish after key=5 is found and what are they?

  **0, containsBT(node 6, 5) returns true and exit, because constainsBT(root->left, 5) returns true**