

# CIS 23: Data Structures and Algorithms

Homework 11  
Prof. Sana Vaziri

Cody Vig

## Problem 1

Consider the following binary search tree

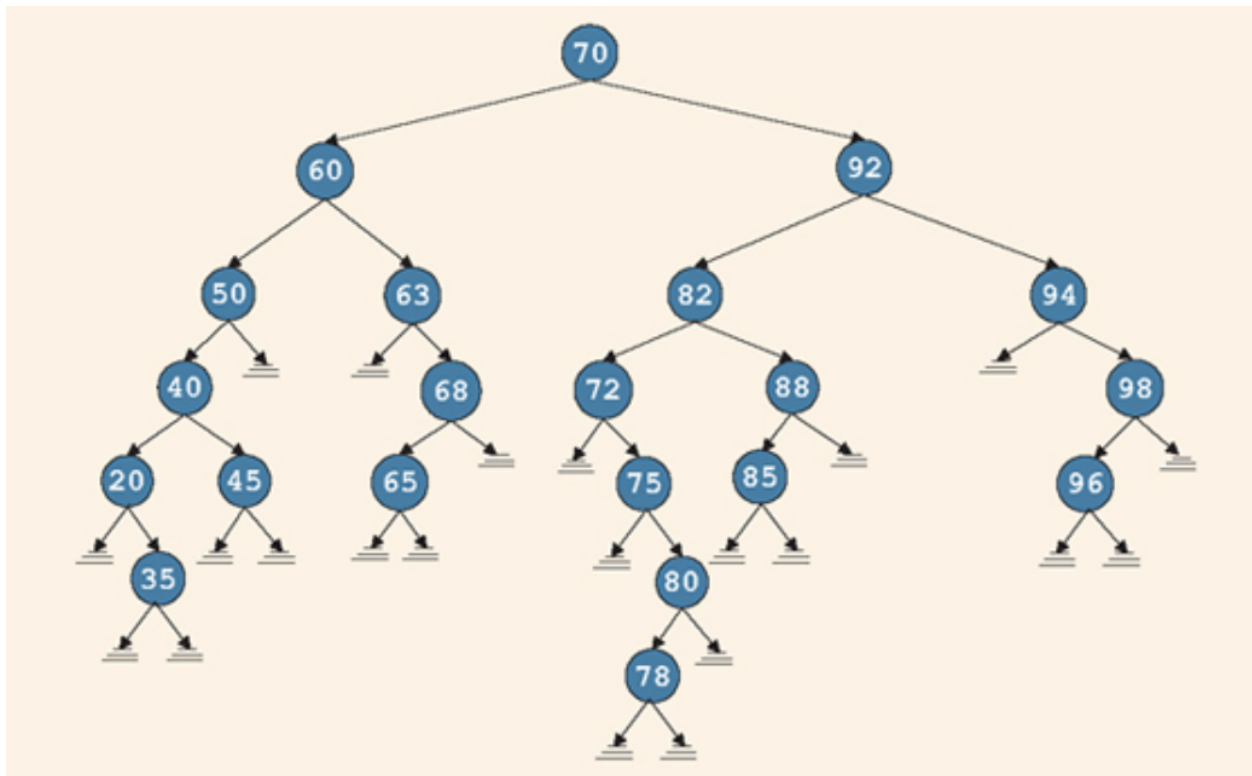


Figure 1: Homework 11 Binary Search Tree

1. List the nodes of this tree in an inorder sequence, then in a preorder sequence, and finally in a postorder sequence.
2. What is the height of the tree with root 60? What is the level of the node with info = 75?
3. List the path from the node with info 92 to the node with info 78.
4. Insert a node with 90 and redraw the tree.
5. Delete the node with 60 and redraw the tree.

## Solution

### Problem 1 Part 1.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lacinia, urna pellentesque gravida hendrerit, libero quam vestibulum lorem, nec sodales nulla nunc sit amet nisl. Morbi euismod volutpat tristique. Ut euismod feugiat vestibulum. Praesent auctor tristique leo elementum vehicula. Donec finibus tristique eleifend. Duis ultricies porta iaculis. Suspendisse potenti. Sed tincidunt neque vitae lorem malesuada dapibus. Suspendisse maximus ultricies iaculis. Maecenas convallis aliquet odio, eget convallis risus scelerisque nec. Vestibulum porta ligula ex, vel imperdiet erat venenatis a. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Cras gravida nisi suscipit felis cursus, vel pellentesque dolor aliquet. Aenean quis metus condimentum, iaculis lacus vel, vulputate nibh. Donec dictum, elit vel placerat porttitor, leo diam malesuada ipsum, vitae ultricies nibh eros eget nisl.

### Problem 1 Part 2.

Suspendisse feugiat augue id libero dictum, in accumsan ipsum feugiat. Sed pharetra diam ac elit condimentum, vitae iaculis elit iaculis. Curabitur auctor metus ac eros scelerisque, a hendrerit turpis porttitor. Curabitur efficitur eleifend mauris, et cursus sapien. Maecenas mi sapien, aliquet non lectus eu, facilisis blandit dui. Aliquam erat volutpat. In consequat, turpis vel malesuada egestas, augue sapien semper urna, pharetra euismod augue mauris sit amet turpis.

### Problem 1 Part 3.

Nullam vitae lacus sem. Maecenas ut lacus tempor, congue leo at, eleifend leo. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Quisque semper quam pretium, dictum ex eget, bibendum lacus. Pellentesque consectetur enim viverra, bibendum lectus vitae, dictum lorem. Nullam hendrerit aliquam condimentum. Aliquam iaculis elit at massa sodales faucibus. Nulla rhoncus massa sit amet viverra venenatis. Quisque vel pharetra diam, at pulvinar quam. Vivamus hendrerit lacus diam, ut commodo erat laoreet id. Cras rhoncus mauris a lectus mollis feugiat. Donec sed sapien sit amet metus hendrerit vehicula quis id arcu. Donec eget urna elementum, luctus ligula ac, placerat ante. Donec mauris sem, vehicula nec magna at, sodales vehicula metus. Proin dui purus, pellentesque ac bibendum sit amet, bibendum sed augue. Pellentesque sed imperdiet magna.

### Problem 1 Part 4.

Aenean accumsan dolor vel tempus vehicula. Pellentesque hendrerit nisi eget justo finibus accumsan. Vestibulum imperdiet magna a placerat commodo. Fusce non vehicula orci. Cras sollicitudin nisl vel lectus suscipit, in pulvinar dolor eleifend. Cras ut tellus mattis, venenatis est nec, pulvinar neque. Curabitur commodo mi ultricies tempor rhoncus. Aenean convallis ultricies hendrerit. Praesent dapibus sit amet augue in eleifend. Nam in pulvinar felis. Suspendisse turpis justo, accumsan id volutpat quis, laoreet vitae felis. Maecenas et eros turpis. Donec consectetur turpis quis diam vestibulum, vel sagittis magna rutrum. Etiam porta elit nec arcu mollis, a sagittis elit pretium. Ut massa tellus, elementum sed interdum a, elementum eu tellus.

### Problem 1 Part 5.

Integer sit amet nisl et lacus lobortis semper. Nullam laoreet orci vestibulum sem malesuada, vel sagittis ante interdum. Nam fringilla odio nec risus euismod finibus ut id purus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec finibus fermentum augue, sed eleifend diam consequat vitae. Fusce pulvinar turpis eu facilisis luctus. In elit felis, lobortis eget sagittis in, convallis non lectus. Ut ornare, nisi ac suscipit semper, erat neque malesuada diam, quis vulputate arcu risus nec mi. Nam eu massa blandit, sagittis nisi ac, pharetra sapien. Curabitur dignissim nisi eleifend ligula congue suscipit at eu libero.

## Problem 2

Consider the following binary tree

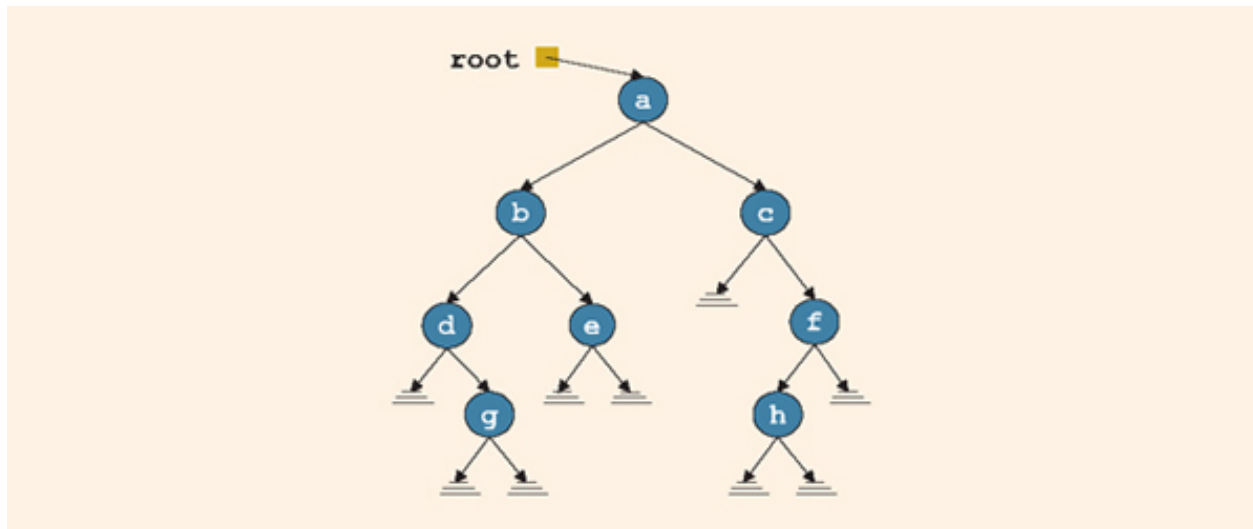


Figure 2: Homework 11 Binary Tree

1. Using the nonrecursive algorithm, perform an inorder traversal of the above tree (by hand) and draw the stack after each push and pop operation.
2. Repeat the above for the nonrecursive preorder traversal drawing the stack after each push and pop operation.

## Solution

### Problem 2 Part 1.

Pseudocode for the nonrecursive inorder traversal is shown:

```
current = root;

while (current is not null and stack is nonempty)
{
    if (current is not null)
    {
        push current onto stack;
        current = current->lLink;    // Move to the left child
    }
    else
    {
        current = stack.top();
        pop stack;
        visit current;              // Visit the node
        current = current->rLink;    // Move to the right child
    }
}
```

### Problem 2 Part 2.

Proin feugiat vestibulum erat, quis ultrices mi vestibulum nec. Ut eu hendrerit lacus. Vivamus et purus ut erat fermentum cursus. Sed vel leo sit amet magna molestie imperdiet. Mauris massa erat, cursus ut tortor at, tempus pharetra erat. Pellentesque vehicula congue tristique. Praesent dapibus turpis at vulputate imperdiet. Praesent fringilla tempor eros id feugiat. Maecenas in vehicula quam.

---

## Problem 3

The `nodeCount()` function in `binaryTreeType` is to count the number of nodes in a binary tree starting from an input root node.

1. Describe a recursive algorithm to for the `nodeCount` function.
2. Complete the `nodeCount` function in the `binaryTreeType` template class.
3. What is the time complexity (in big-O notation) for `nodeCount`?

### Solution

#### Problem 3 Part 1.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas vel placerat quam. Nulla vel eleifend sapien. Curabitur id neque eu elit ultrices varius. Morbi eu ligula tincidunt, feugiat nibh eu, blandit ex. Aliquam aliquet ligula sed augue pharetra pharetra. Curabitur consequat elementum tortor, in iaculis justo tincidunt a. Fusce nec metus lorem. Sed auctor felis ut nisl varius, id porttitor libero interdum.

#### Problem 3 Part 2.

See `binaryTree.h`

#### Problem 3 Part 3.

Morbi vestibulum gravida tempor. Praesent in vulputate nisl. Maecenas efficitur ante ac eros aliquam condimentum. Sed ac ultricies magna. Mauris id tincidunt sem. Cras blandit orci a lorem malesuada, non sollicitudin dolor interdum. Donec volutpat condimentum ligula, eget ultricies erat imperdiet eget. Curabitur feugiat ullamcorper laoreet. Maecenas mollis commodo nulla, a tincidunt libero sollicitudin ac. Phasellus mattis nunc ac tortor condimentum imperdiet nec id neque. Nullam venenatis tincidunt nulla, in cursus augue. Nulla a pretium nulla, sed tincidunt nulla. Nulla facilisis purus eget magna rutrum, non vestibulum lorem condimentum. Proin fermentum at leo vel ultricies. Quisque ut mollis justo.

---

## Problem 4

The `leavesCount()` function in `binaryTreeType` is to count the number of leaves in a binary tree starting from an input root node.

1. Describe a recursive algorithm to for this function.
2. Complete the `leavesCount()` function in the `binaryTreeType` template class.
3. What is the time complexity (in big-O notation) for `leavesCount`?

## Solution

### Problem 4 Part 1.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla consequat dui ut vulputate condimentum. Pellentesque sit amet velit eget diam mattis rhoncus vitae semper ex. Suspendisse pulvinar orci in pharetra sodales. Vivamus libero arcu, pellentesque eu nunc id, bibendum facilisis magna. Aenean sit amet orci feugiat, maximus elit id, gravida lacus. Fusce pulvinar ac diam quis feugiat. Sed sit amet vehicula tellus, a auctor lectus. Nam diam eros, venenatis cursus justo vel, imperdiet lobortis risus. Proin consectetur erat a faucibus tempor. Cras id ipsum et magna blandit porttitor eget at neque.

### Problem 4 Part 2.

See `binaryTree.h`

### Problem 4 Part 3.

Morbi at enim sed libero dictum consectetur id lobortis tellus. Vestibulum leo tortor, imperdiet sit amet tellus vitae, semper sagittis augue. Morbi viverra vulputate urna. Vestibulum maximus blandit ante, at egestas tellus eleifend ultrices. Proin tortor nisl, efficitur a semper eleifend, aliquam ut mi. Ut non finibus nibh. Integer sed dolor scelerisque, dictum magna nec, fermentum mi. Phasellus at nisl nisi. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean molestie imperdiet tortor vitae tincidunt. Nunc volutpat, purus non convallis rhoncus, lectus dui faucibus felis, ut sodales augue magna nec mauris. Nunc ac lorem sit amet ipsum eleifend tempor. Ut molestie sapien ac mi consequat, id vestibulum nunc faucibus. Morbi augue tellus, cursus eget cursus quis, lacinia sit amet libero. Sed vel tincidunt dui. Nulla ex dui, mattis id fringilla id, finibus in quam.

---