

## COSC 2436 Lab 5: Pre-order Binary Search Tree Traversal

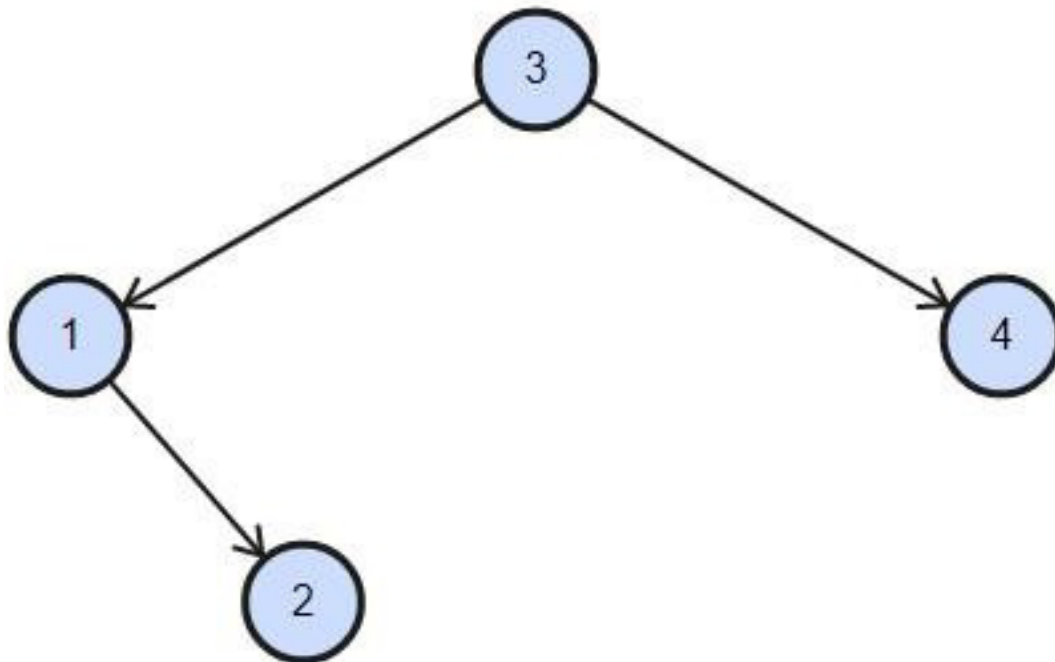
Create a C++ program to read integers from a file and construct a Binary Search Tree(BST). You will then output the Pre-order traversal of the tree. In addition, we will add additional information to show the traversal path through the tree to show the path required to reach each of the nodes.

1. Input files
  - The input files contain a single line with the numbers, separated by spaces. to be added into the tree.
2. Output files
  - Output as single line indicating that a closed tour solution exists or does not exist for the size of board specified.
3. Example

For example, if the input file contains the numbers: 3 1 2 4 Your program will:

  1. Read 3, making it the root node
  2. Read 1, which is less than the root, go to the left child of 3, and since the left child is empty, insert 1 as left child of 3
  3. Read 2, which is less than the root , and go to the left child of 3. Because there is a child node, go to the right child of 1, and since the right child is empty, insert 2 as right child of 1.
  4. Read 4, which is greater than the root, and go to the right child of 3. Since the right child is empty, insert 4 as right child of 3

Then the binary tree stored in your program should be:



The final step of the program is to output the pre-order traversal of this BST to a file.

## COSC 2436 Lab 5: Pre-order Binary Search Tree Traversal

### **Preorder traversal means:**

Access the current node. Traverse the left subtree by recursively calling the pre-order function. Traverse the right subtree by recursively calling the pre-order function.

Then doing a preorder traversal on the previous BST will give:

3 1 2 4

To help visualize the actual structure of the tree, we will add route information as we visit each node where  $x$  means root node. In this case, an  $l$  means a transition to the left child and  $r$  means a transition to the right child.

For the example tree, this decorated pre-order traversal would be:

[x] 3

[xl] 1

[xlr] 2

[xr] 4

The output file can be read as:

3 is at [x], so it is the root

1 is at [xl] so you have to go root->left to access it

2 is at [xlr] so you have to go root->left->right to access it

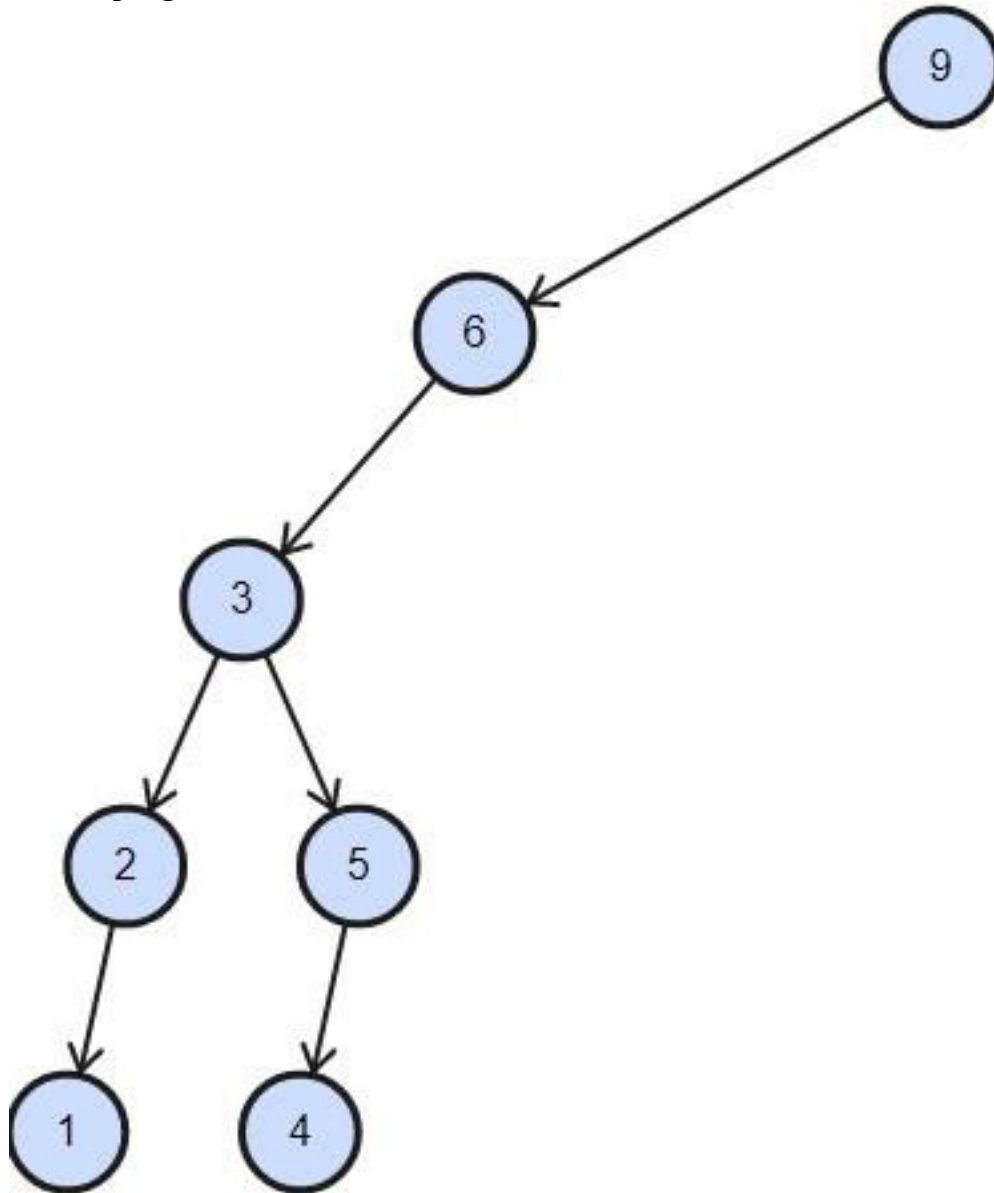
4 is at [xr] so you have to go root->right to access it

COSC 2436 Lab 5: Pre-order Binary Search  
Tree Traversal

**input1.txt:**

9 6 3 2 5 4 1

**BST in program:**



**Expected output1.txt:**

[x] 9

[xl] 6

[xll] 3

[xlly] 2

[xllyl] 1

[xllyr] 5

[xllyrl] 4

## COSC 2436 Lab 5: Pre-order Binary Search Tree Traversal

### 4. Reminder

- Turn in your lab assignment to our Linux server, follow the link [here](#) for more instructions.
- Make sure to only have **one (1)** .cpp file with the main() function in your working directory, otherwise your program will fail the grading script.
  - Create a folder under your root directory, name the folder *lab5* (case sensitive), copy all your .cpp and .h files to the folder (ArgumentManager.h is also needed)
  - Only include the necessary files (.cpp and .h files) in your working directory in your final submission
  - To test your program, copy the input files into the server and run your program. After verifying that they pass, delete the .txt files.

Please reach out to myself or the TAs for any clarifications or typos.