

Q2.1

printTree is inorder traversal

Q2.3

When preorder with no parameter is called in q2.cpp, it refers to BinarySearchTree.h. In that header file, the function of preorder with no parameter first checks if the tree is empty. If not, call the preorder function with two parameters, which visits and prints out every node in preorder order.

Q2.5

For any AVL tree, the maximum difference between max and min depth is one. The difference between max and min depth significantly increases after we randomly insert 100000 nodes

Q3

space complexity:

Since my function is using recursive method, which adds a new stack frame whenever leaving current invocation, the number of stack frame is equal to the number of levels the tree has. Therefore, the space complexity is $S(\log N)$

time complexity:

Since the height function is $O(N)$ and it is called in every iteration of diameter function, which is $O(N)$ without height function, the time complexity as a whole will be the product of two, which is $O(n^2)$

Q5

space complexity:

Since my function is using queue method, and every node is "dequeued" right after check of common node, it does not require additional stack frame to perform the next iteration of the while loop.

Therefore, the space required is a constant and does not increase as the size of the tree increases, so $S(1)$.

time complexity: Since every node is visited and added to the queue one by one, the time complexity is $O(N)$.