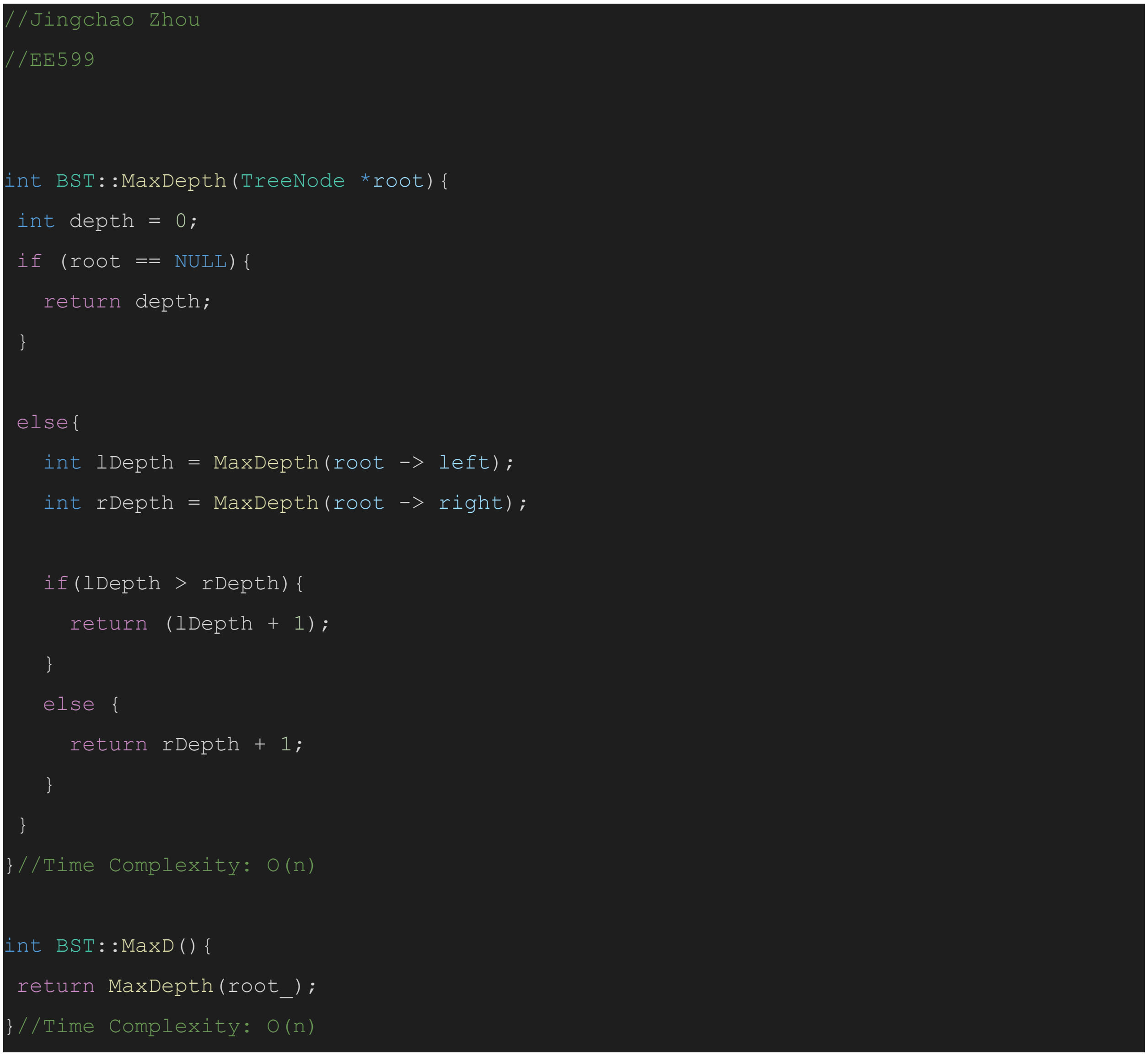
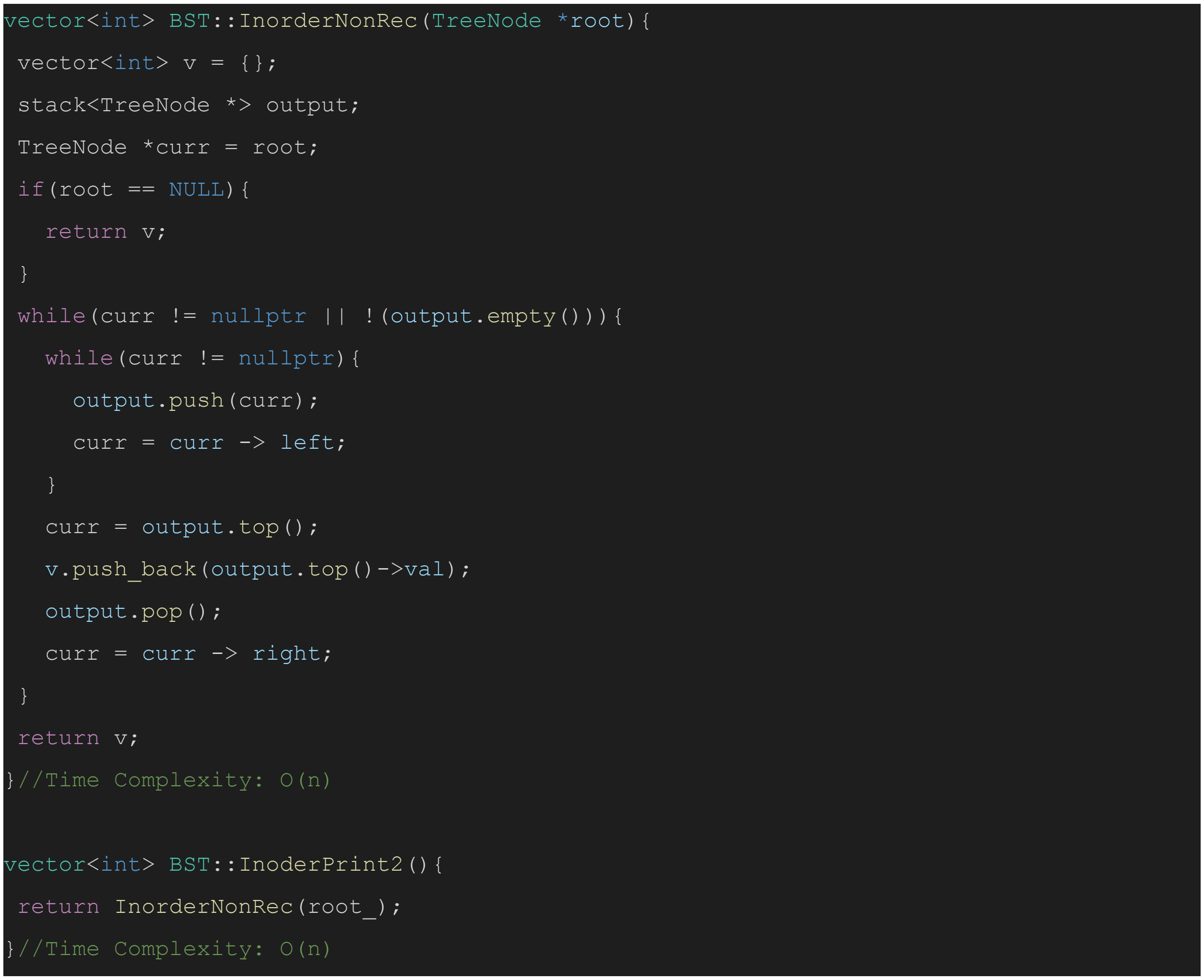
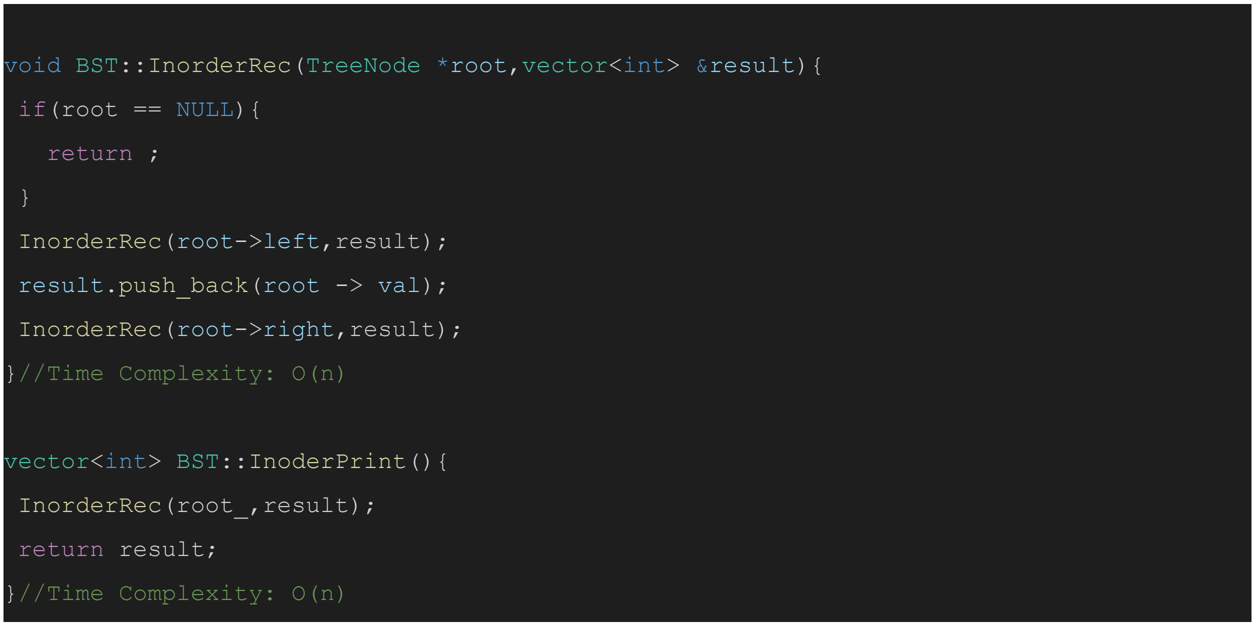


```
//Jingchao Zhou
//EE599


int BST::MaxDepth(TreeNode *root){
 int depth = 0;
 if (root == NULL){
   return depth;
 }

 else{
   int lDepth = MaxDepth(root -> left);
   int rDepth = MaxDepth(root -> right);

   if(lDepth > rDepth){
     return (lDepth + 1);
   }
   else {
     return rDepth + 1;
   }
 }
}//Time Complexity: O(n)

int BST::MaxD(){
 return MaxDepth(root_);
}//Time Complexity: O(n)
```



```
vector<int> BST::InorderNonRec(TreeNode *root){
 vector<int> v = {};
 stack<TreeNode *> output;
 TreeNode *curr = root;
 if(root == NULL){
   return v;
 }
 while(curr != nullptr || !(output.empty())){
   while(curr != nullptr){
     output.push(curr);
     curr = curr -> left;
   }
   curr = output.top();
   v.push_back(output.top()->val);
   output.pop();
   curr = curr -> right;
 }
 return v;
}//Time Complexity: O(n)

vector<int> BST::InoderPrint2(){
 return InorderNonRec(root_);
}//Time Complexity: O(n)
```



```
void BST::InorderRec(TreeNode *root,vector<int> &result){
 if(root == NULL){
   return ;
 }
 InorderRec(root->left,result);
 result.push_back(root -> val);
 InorderRec(root->right,result);
}//Time Complexity: O(n)

vector<int> BST::InoderPrint(){
 InorderRec(root_,result);
 return result;
}//Time Complexity: O(n)
```