# Lab03实验报告

班级：软外1903 姓名：秦政瀚 学号：201906150312

## Task 1

### Theory:

* Binary search tree characteristics:
  + The value of all nodes on the left subtree is less than or equal to the value of its root node.
  + The value of all nodes on the right subtree is greater than or equal to the value of its root nodes.
  + Left and right subtrees are also binary sorting trees.
* Advantages of binary search tree:
  + Binary sorting trees are a more useful trade-off .
  + Array search is more convenient, you can use the subscript directly, but it is more troublesome to delete or insert certain elements.
  + Linked lists, by contrast, delete and insert elements quickly, but find them slowly.
  + Binary sorting trees have both the benefits of a list and the benefits of arrays.
  + : It is more useful to process large amounts of dynamic data.
* Disadvantages of binary search tree:
  + Sequential storage can waste space (in the case of incomplete binary trees), but is more efficient when reading a specified node O(0).
  + Chain storage wastes less space when relatively large than binary trees, but is less efficient when reading a given node O (nlogn).

### Definition：

* struct Node{} that definate the node-struct.
* Node\* left that a pointer to the left child node: left.
* Node\* right that a pointer to the right child node: right.
* int content that an integer variable for the content: content.

## Task 2

### Theory:

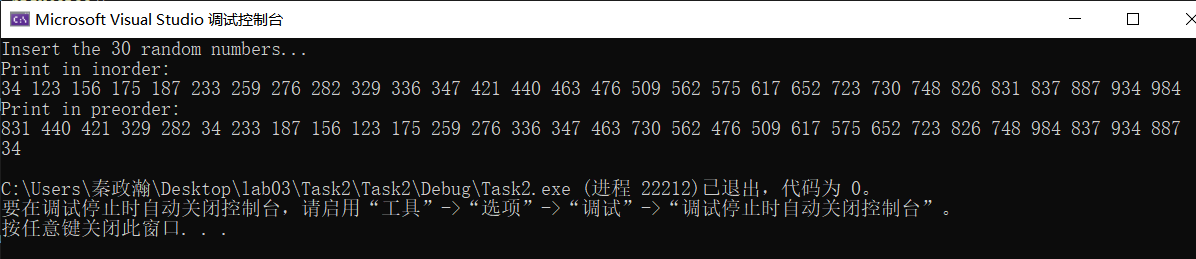
* Using a linked implementation to complete the Binary Search Tree(BST).

### Function:

* bool isEmpty( ) that returns true if the BST is empty.
* void insert(int k) that adds the integer k to the BST.
  + If the content of new node to insert equals to the content in any node, the function to nothing.
  + Using recursion to insert a new node and if the content of new node is less than current node's content, pass the left node as parameter and execute next recursion and conversely otherwise.
* bool contains(int k) that returns true if the integer k is stored in the BST and false otherwise.
  + Using while-loop from the root node to find the node which contains the content which equals to k by judge whether the k is less or more than the content in current node.
* int findMinValue( ) that returns the value of the smallest integer stored in the tree (without removing it from the tree).
  + Using while-loop to find the left node of the current node until the next left node is nullptr.
* void printTree( ) that prints the integers in the BST in inorder.
  + Using recursion to implement and cout the content of current node between the recursion to the left node and the recursion to the right node.
* void printTreePreorder( ) that prints the integers in BST in preorder.
  + Using recursion to implement and cout the content of current node before the recursion to the left node and the recursion to the right node.

## Task 3

### Screenshot of output:



## Task 4

### Answer:

* According to the output in Task 3. The height of the tree is 5.

