



Final Examination
Practical Question Paper

CSD201 # Data Structure and Algorithm

Semester: SU21

Curriculum Code: Software Engineering

Date: 23/07/2020

Duration: 90 Minutes

Total Marks: 30

Do not write on this question paper and return it to the Invigilator after the examination.

1. Reverse traversing

Write a program to build a BST (Binary Search Tree) by inserting N ($1 \leq N \leq 100$) integer values into the BST one by one. Note that the nodes have **no duplicate** values.

Your task is showing the Reverse traversing of the BST.

Here are 3 steps of reverse traversing algorithm:

Step 1: Right-child traversing

Step 2: Process current node

Step 3: Left-child traversing

For example,

- Create a BST tree by inserting into the tree N integer values as follows: 7, 9, 4, 1, 12, 6, 10.
- The reverse traversing of the BST tree is: 12, 10, 9, 7, 6, 4, 1.

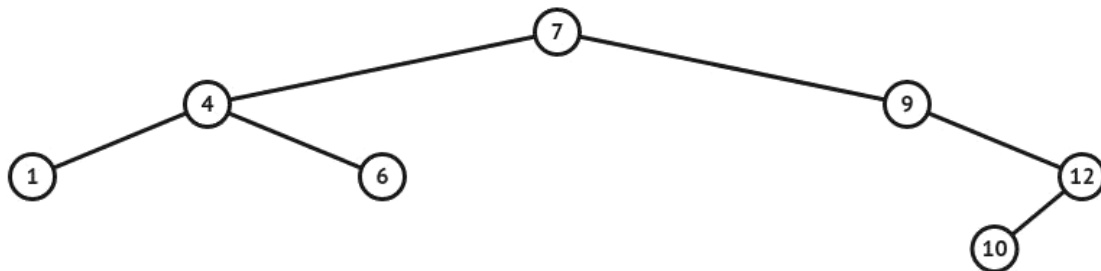


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the *reverseTraversing_input.txt* text file:

The first line contains a positive integer N ($1 \leq N \leq 100$) which is the number of integer values to insert into the BST.

The second line containing N integers that will be inserted into the BST one by one, each number separated by at least one space.

The output: the results need to be saved to the *reverseTraversing_output.txt* text file:

One line contains the list of numbers representing the reverse traversing of the BST. Each number separated by one comma.

Sample Input 1	Sample Output 1
7	12, 10, 9, 7, 6, 4, 1
7 9 4 1 12 6 10	

Sample Input 2	Sample Output 2
10	100, 98, 91, 78, 74, 55, 36, 30, 21, 18
78 21 36 18 30 91 74 55 98 100	

2. BST duplicate

Write a program to build a binary search tree by inserting N ($1 \leq N \leq 100$) integer values into the BST one by one. Note that the nodes **can have duplicate values**.

Your task is to display the BFS traversing of the BST.

For example,

- Create a BST tree by adding into the tree N integer values as follows: 7, 9, 4, 9, 1, 12, 6, 7, 1, 10.
- The BFS traversing of the BST tree is: 7, 7, 4, 9, 9, 1, 1, 6, 12, 10.

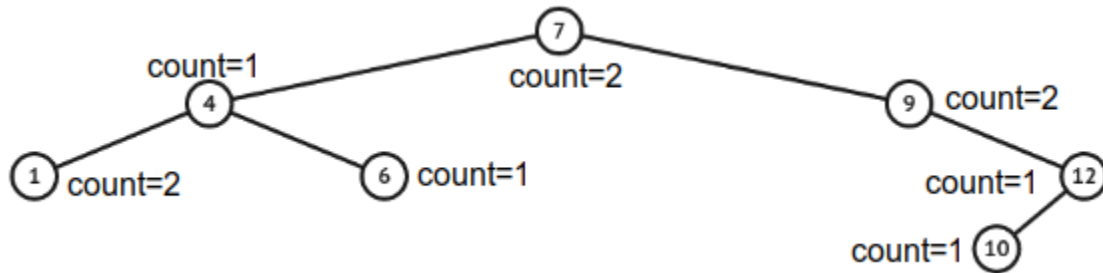


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the *duplicate_input.txt* text file:

The first line contains a positive integer N ($1 \leq N \leq 100$) which is the number of integer values to insert into the BST.

The second line containing N integers that will be inserted into the BST one by one, each number separated by at least one space.

The output: the results need to be saved to the *duplicate_output.txt* text file:

The list of numbers representing the BFS traversing of the BST. Each number separated by one comma.

Sample Input 1	Sample Output 1
10	7, 7, 4, 9, 9, 1, 1, 6, 12, 10
7 9 4 9 1 12 6 7 1 10	

Sample Input 2	Sample Output 2
10	78, 21, 91, 18, 36, 98, 30, 74, 100, 55
78 21 36 18 30 91 74 55 98 100	

3. BST one-child node counting

Write a program to build a binary search tree by inserting N ($1 \leq N \leq 100$) integer values into the BST one by one. Note that the nodes have **no duplicate** values.

Your task is showing the list of one-child node of the BST by using **pre-order traversing**. Note that one-child nodes are node that has only left-child or has only right-child.

For example,

- Create a BST tree by inserting into the tree N integer values as follows: 7, 9, 4, 1, 12, 6, 10.
- The BST has 2 one-child nodes including 9 and 12.

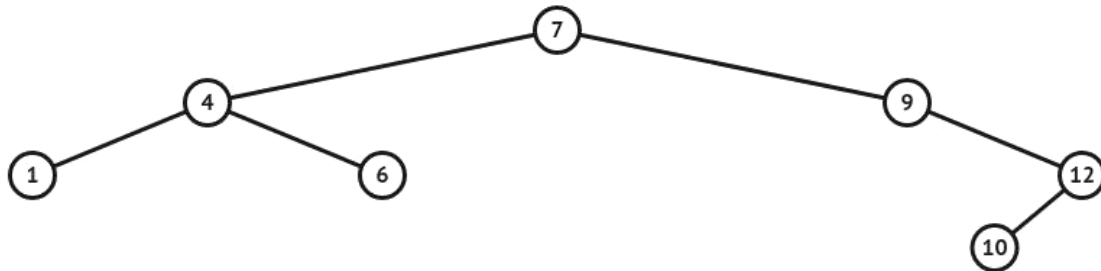


Figure 4. The BST has 3 inside nodes including 4 and 9 and 12

The input: are stored in the *oneChildNode_input.txt* text file:

The first line contains a positive integer N ($1 \leq N \leq 100$) which is the number of integer values to insert into the BST.

The second line containing N integers that will be inserted into the BST one by one, each number separated by at least one space.

The output: the results need to be saved to the *oneChildNode_output.txt* text file:

Only one line contains the list of one-child nodes of the BST by using pre-order traversing, each number separated by one comma.

Sample Input 1	Sample Output 1
7	9, 12
7 9 4 1 12 6 10	

Sample Input 2	Sample Output 2
9	7, 6, 2, 18
10 7 6 2 18 13 2 6 4	