1. BST traversing

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 to display the pre-order and post-order and in-order traversing of the BST.

For example,

- Create a BST tree by successively adding to the tree N integer values as follows: 7, 9, 4, 1, 12, 6, 10.
- The pre-order traversing of the BST tree is: 7, 4, 1, 6, 9, 12, 10.
- The post-order traversing of the BST tree is: 1, 6, 4, 10, 12, 9, 7.

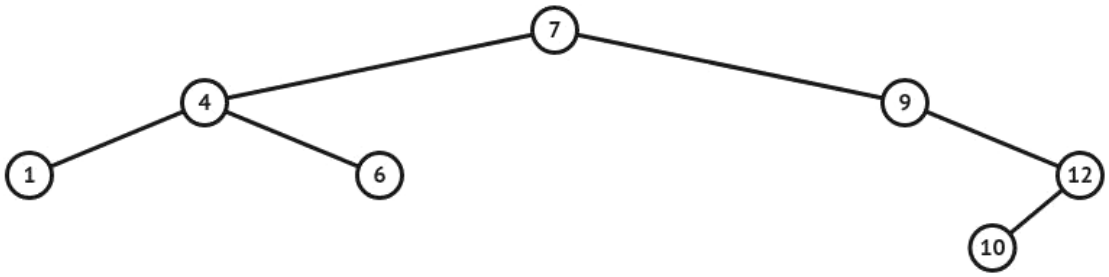


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the ***ex01_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 ***ex01_output.txt*** text file:

The first line contains the sequence of numbers representing the pre-order traversing of the BST.

The second line contains the sequence of numbers representing the post-order traversing of the BST.

The third line contains the sequence of numbers representing the in-order traversing of the BST.

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

Sample Input 2	*Sample Output 2*

9 10 7 6 2 18 13 2 6 4	10,7,6,2,4,18,13 4,2,6,7,13,18,10 2,4,6,7,10,13,18

2. BST deleting

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 delete M ($1 \leq M < N$) nodes of the BST and display the pre-order and post-order traversing of the BST.

For example,

- Create a $BST^{(1)}$ tree by successively adding to the tree N integer values as follows: 7, 9, 4, 1, 12, 6, 10.
- The pre-order traversing of the $BST^{(1)}$ tree is: 7, 4, 1, 6, 9, 12, 10.
- The post-order traversing of the $BST^{(1)}$ tree is: 1, 6, 4, 10, 12, 9, 7.

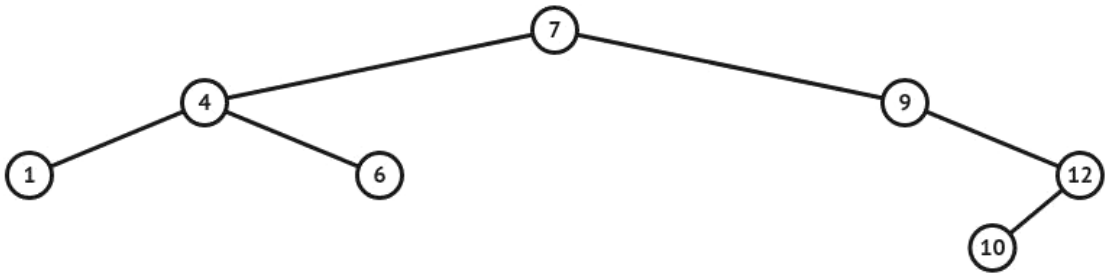


Figure 2. The $BST^{(1)}$ that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

After deleted two nodes 6 and 7 of the $BST^{(1)}$:

- The pre-order traversing of the $BST^{(2)}$ tree is: 4, 1, 9, 12, 10.
- The post-order traversing of the $BST^{(2)}$ tree is: 1, 10, 12, 9, 4.

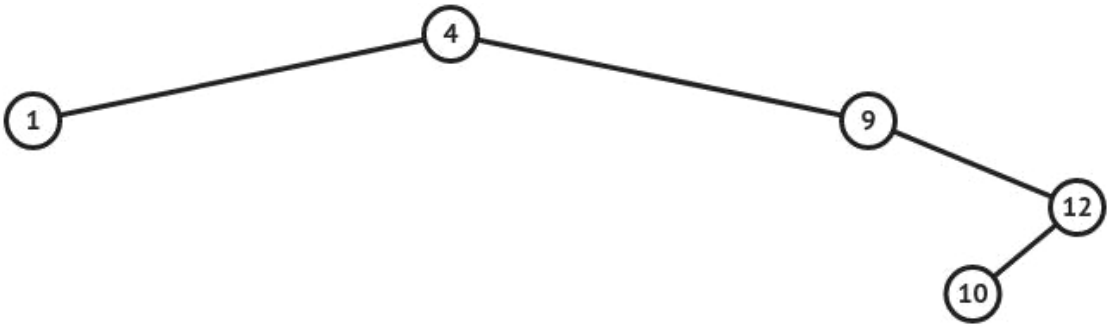


Figure 3. After deleted two nodes 6 and 7 of the $BST^{(1)}$, we get the $BST^{(2)}$

The input: are stored in the ***ex02_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 third line contains a positive integer M ($1 \leq M < N$) which is the number of values to be deleted.

The fourth line containing M integers that will be deleted one by one, each number separated by at least one space.

The output: the results need to be saved to the ***ex02_output.txt*** text file:

The first line contains the sequence of numbers representing the pre-order traversing of the BST(2).

The second line contains the sequence of numbers representing the post-order traversing of the BST(2).

Sample Input 1	*Sample Output 1*
7 7 9 4 1 12 6 10 2 6 7	4,1,9,12,10 1,10,12,9,4

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

3. BST leaf 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 to display the number of leaves of the BST.

For example,

- Create a BST tree by successively adding to the tree N integer values as follows: 7, 9, 4, 1, 12, 6, 10.
- The BST has 3 leaves including 1 and 6 and 10.

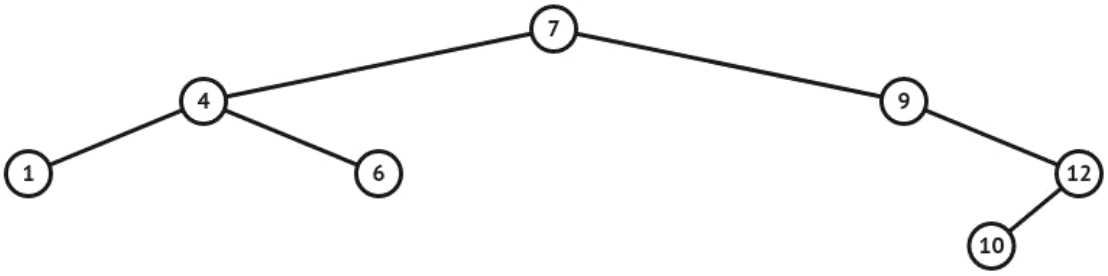


Figure 4. The BST has 3 leaves including 1 and 6 and 10

The input: are stored in the ***ex03_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 ***ex03_output.txt*** text file:

Only one line contains the list of leaves of the BST, each value separated by commas.

Sample Input 1	*Sample Output 1*
7 7 9 4 1 12 6 10	1,6,10 3 1,6,10

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

4. BST inside 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 to display all internal nodes of the BST.

For example,

- Create a BST tree by successively adding to the tree N integer values as follows: 7, 9, 4, 1, 12, 6, 10.
- The BST has 3 internal nodes including 4 and 9 and 12.

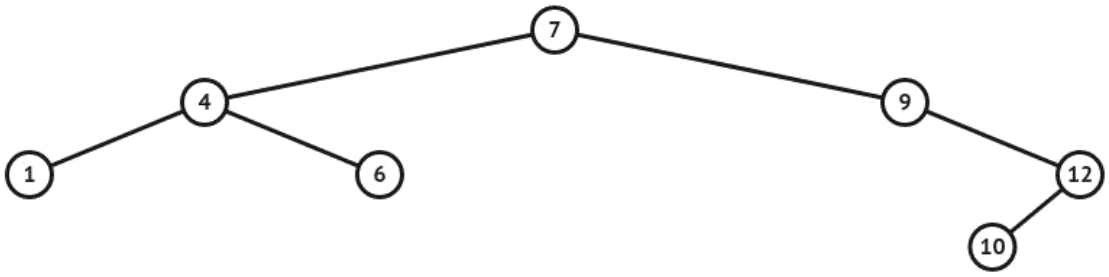


Figure 4. The BST has 3 leaves including 1 and 6 and 10

The input: are stored in the ***ex04_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 ***ex04_output.txt*** text file:

Only one line contains the list of internal nodes of the BST, each value separated by commas.

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

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

5. BST one child 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 to display all the nodes having exactly one child of the BST.

For example,

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

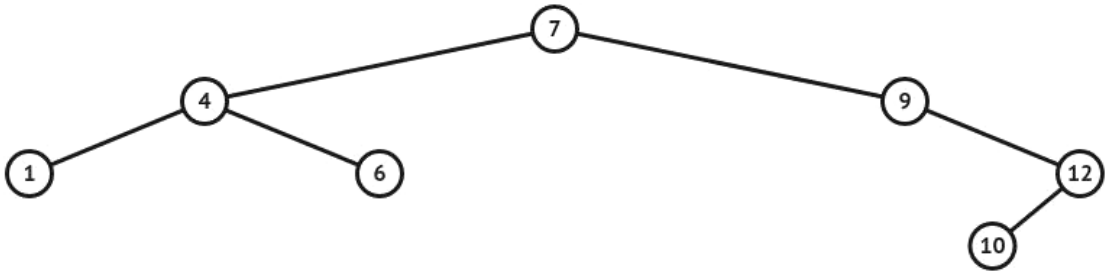


Figure 4. The BST has 3 leaves including 1 and 6 and 10

The input: are stored in the ***ex05_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 ***ex05_output.txt*** text file:

Only one line contains the list of all the nodes having exactly one child of the BST, each value separated by commas.

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

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

6. BST two child 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 to display all the nodes having two child of the BST.

For example,

- Create a BST tree by successively adding to the tree ***N*** integer values as follows: 7, 9, 4, 1, 12, 6, 10.
- The BST has 2 nodes with two child including 7 and 4

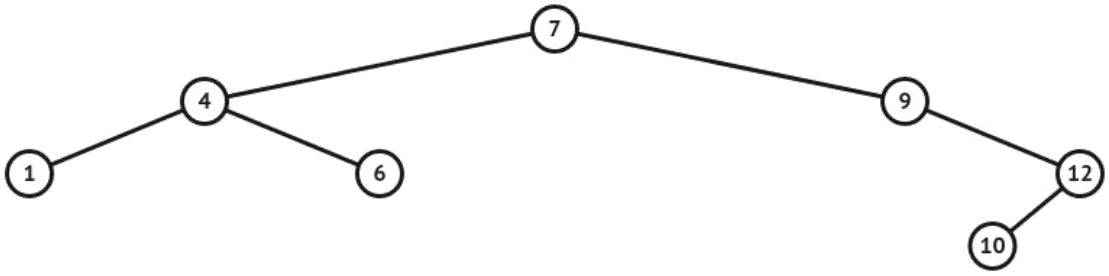


Figure 4. The BST has 3 leaves including 1 and 6 and 10

The input: are stored in the ***ex06_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 ***ex06_output.txt*** text file:

Only one line contains the list of all the nodes having exactly two child of the BST, each value separated by commas.

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

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

7. BST searching

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 to print path from root to a given node in the BST.

For example,

- Create a BST tree by successively adding to the tree ***N*** integer values as follows: 7, 9, 4, 1, 12, 6, 10.
- The BST has 2 nodes with two child including 7 and 4

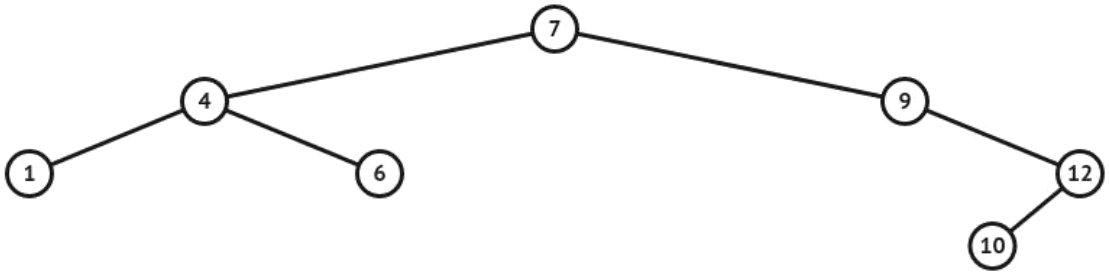


Figure 4. The BST has 3 leaves including 1 and 6 and 10

The input: are stored in the ***ex07_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 third line contains a positive integer ***M*** which is the number of values to be search.

The output: the results need to be saved to the ***ex07_output.txt*** text file:

Only one line contains the list of all the nodes having exactly two child of the BST, each value separated by commas.

Sample Input 1	*Sample Output 1*
7 7 9 4 1 12 6 10 10	7->9->12->10

Sample Input 2	*Sample Output 2*
9 10 7 6 2 18 13 2 6 4	10->7->6->2->4

4	

Sample Input 3	*Sample Output 2*
9 10 7 6 2 18 13 2 6 4 15	No

8. BST DFS and BFS traversing

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

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

For example,

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

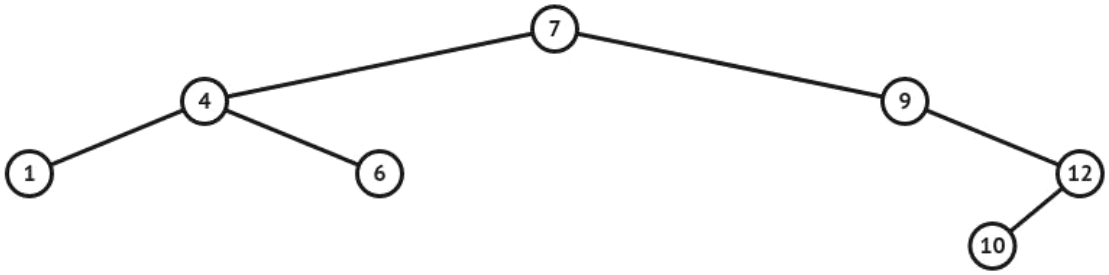


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the ***ex08_input.txt*** text file:

The first line contains a positive integer ***N*** ($1 \le N \le 100$) 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 ***ex08_output.txt*** text file:

The first line contains the sequence of numbers representing the DFS traversing of the BST.

The second line contains the sequence of numbers representing the BFS traversing of the BST.

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

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

9. BST Level

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 to Print Levels of all nodes in the BST.

For example,

- Create a BST tree by successively adding to the tree ***N*** integer values as follows: 7, 9, 4, 1, 12, 6, 10.

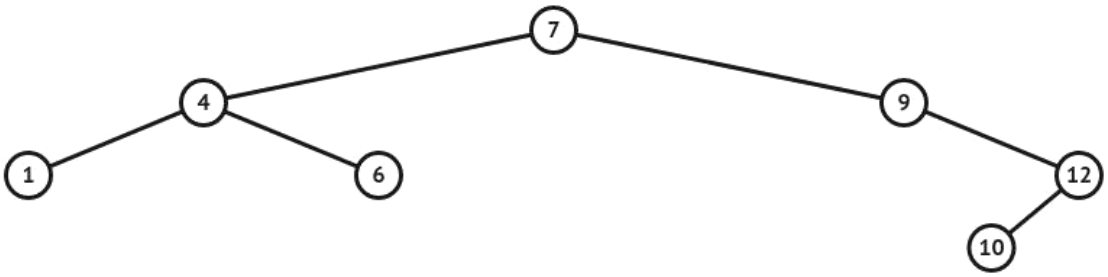


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the ***ex09_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 ***ex09_output.txt*** text file:

Only one line contains levels of all nodes in the BST, each value with its level separated by commas.

Sample Input 1	*Sample Output 1*
7 7 9 4 1 12 6 10	7[0],4[1],1[2],6[2],9[1],12[2],10[3]

Sample Input 2	*Sample Output 2*
9 10 7 6 2 18 13 2 6 4	10[0],7[1],6[2],2[3],4[4],18[1],13[2]

10. BST Parent

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 to Print parent of all nodes in the BST.

For example,

- Create a BST tree by successively adding to the tree ***N*** integer values as follows: 7, 9, 4, 1, 12, 6, 10.

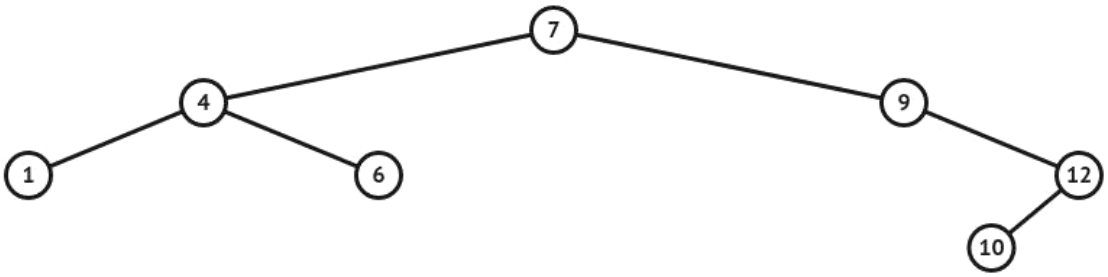


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the ***ex10_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 ***ex10_output.txt*** text file:

Only one line contains parent of all nodes in the BST, each value with its parent separated by commas.

Sample Input 1	*Sample Output 1*
7 7 9 4 1 12 6 10	7[null],4[7],1[4],6[4],9[7],12[9],10[12]

Sample Input 2	*Sample Output 2*
9 10 7 6 2 18 13 2 6 4	10[null],7[10],6[7],2[6],4[2],18[10],13[18]

11. BST height

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 to Print height of all nodes in the BST.

For example,

- Create a BST tree by successively adding to the tree ***N*** integer values as follows: 7, 9, 4, 1, 12, 6, 10.

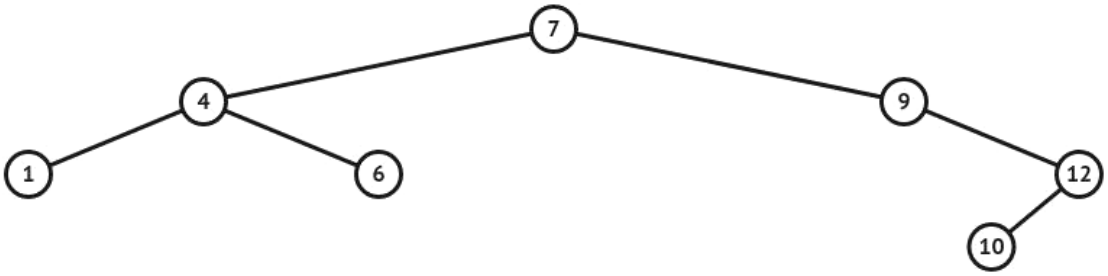


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the ***ex10_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 ***ex10_output.txt*** text file:

Only one line contains height of all nodes in the BST, each value with its height separated by commas.

Sample Input 1	*Sample Output 1*
7 7 9 4 1 12 6 10	7[3],4[1],1[0],6[0],9[2],12[1],10[0]

Sample Input 2	*Sample Output 2*
9 10 7 6 2 18 13 2 6 4	10[4],7[3],6[2],2[1],4[0],18[1],13[0]

12. BST Balancing

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.

Given an unbalanced BST, Your task is to convert it into a balanced BST

For example,

- Create a BST$^{(1)}$ tree by successively adding to the tree N integer values as follows: 10 7 6 2 18 13 4
- The pre-order traversing of the BST$^{(1)}$ tree is: 10,7,6,2,4,18,13.
- The post-order traversing of the BST$^{(1)}$ tree is: 4,2,6,7,13,18,10

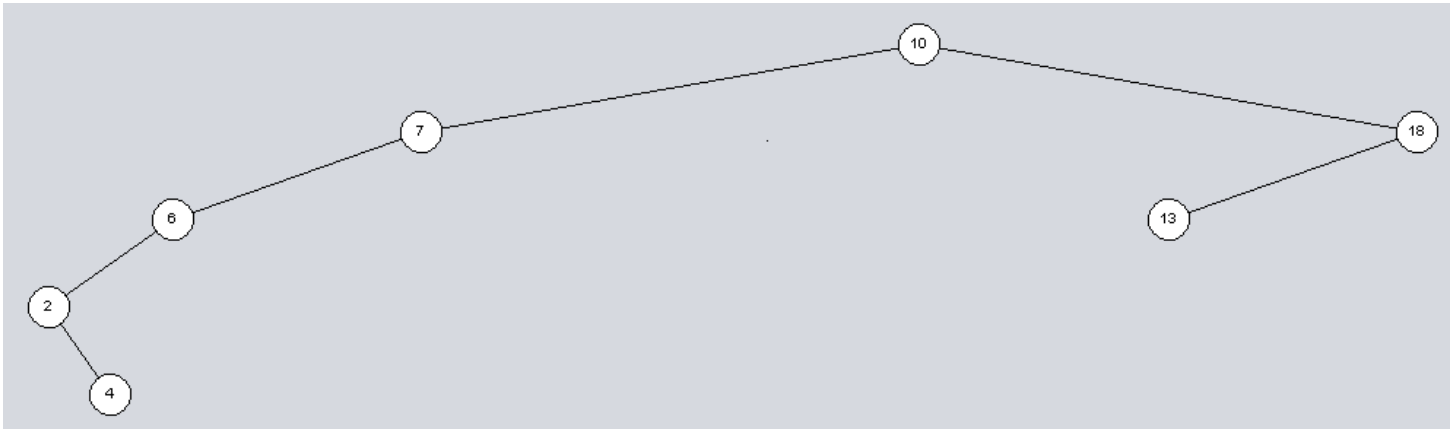


Figure 2. The BST$^{(1)}$ that created by inserting 10 7 6 2 18 13 4 one by one

After balanced of the BST$^{(1)}$:

- The pre-order traversing of the BST$^{(2)}$ tree is: 7, 4, 2, 6, 13, 10, 18
- The post-order traversing of the BST$^{(2)}$ tree is: 2, 6, 4, 10, 18, 13, 7

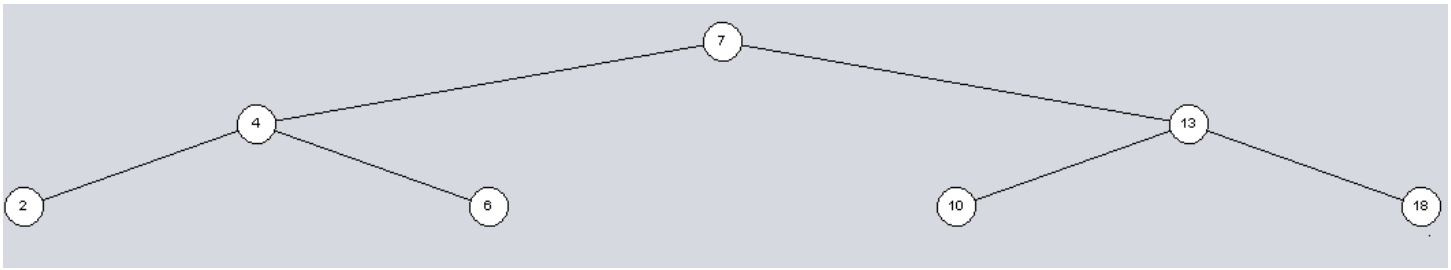


Figure 3. After balanced of the BST$^{(1)}$, we get the BST$^{(2)}$

The input: are stored in the ***ex12_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 ***ex12_output.txt*** text file:

The first line contains the sequence of numbers representing the pre-order traversing

of the BST$^{(2)}$.

The second line contains the sequence of numbers representing the post-order traversing of the BST$^{(2)}$.

The third line contains the sequence of numbers representing the in-order traversing of the BST$^{(2)}$.

Sample Input 1	*Sample Output 2*
9 10 7 6 2 18 13 2 6 4	7,4,2,6,13,10,18 2,6,4,10,18,13,7 2,4,6,7,10,13,18
Sample Input 2	*Sample Output 1*
7 7 9 4 1 12 6 10	7,4,1,6,10,9,12 1,6,4,9,12,10,7 1,4,6,7,9,10,12

13. BST Max Min

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 to Print max and min in the BST.

For example,

- Create a BST tree by successively adding to the tree ***N*** integer values as follows: 7, 9, 4, 1, 12, 6, 10.

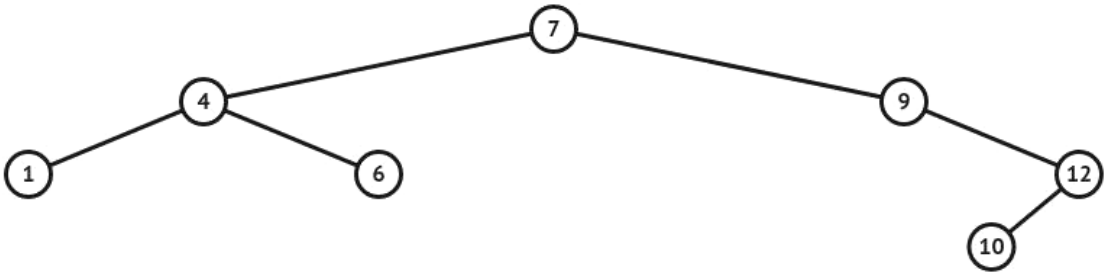


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the ***ex13_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 ***ex13_output.txt*** text file:

The first line contains max value in BST

The second line contains min value in BST

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

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

14. BST Sibling

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 to find Sibling node of a given node value in in the BST.

For example,

- Create a BST tree by successively adding to the tree ***N*** integer values as follows: 7, 9, 4, 1, 12, 6, 10.
- The sibling of 4 is 9, the sibling 6 is 1, the sibling of 12 is null

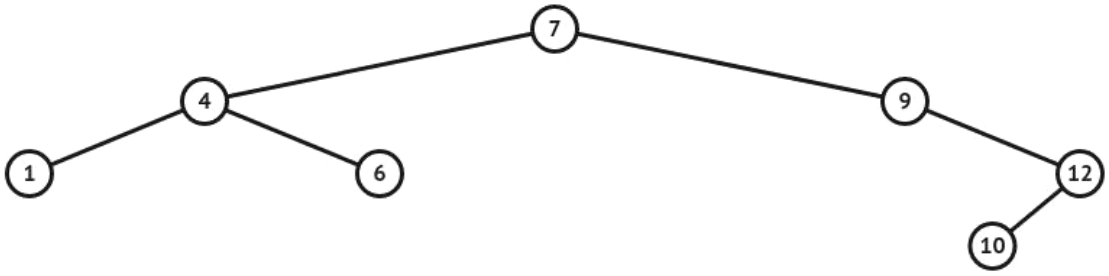


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the ***ex14_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 third line contains a positive integer ***M*** which is the number of values to be found sibling.

The fourth line containing ***M*** integers that will be found sibling one by one, each number separated by at least one space.

The output: the results need to be saved to the ***ex14_output.txt*** text file:

Only one line contains siblings of nodes to look for in the BST, each value separated by a comma.

Sample Input 1	*Sample Output 1*
7 7 9 4 1 12 6 10 3 4 6 12	9, 1, null

Sample Input 2	*Sample Output 2*

10 10 7 6 2 18 13 2 6 4 1 3 7 13 4	18, null, 1

15. BST: a full binary tree or not

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 to check whether a binary tree is a full binary tree or not. A full binary tree is defined as a binary tree in which all nodes have either zero or two child nodes

For example,

- Create a BST tree by successively adding to the tree ***N*** integer values as follows: 7, 9, 4, 1, 12, 6, 10. And This BST is not full binary tree.

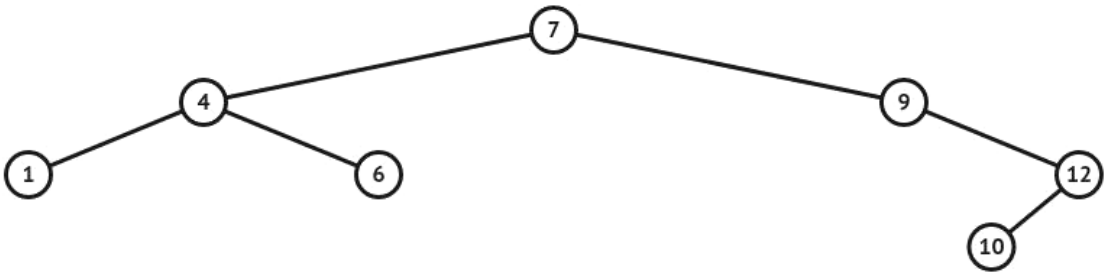


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

- Create a BST tree by successively adding to the tree ***N*** integer values as follows: 10, 7, 15, 5, 9, 5, 6. And This BST is full binary tree.

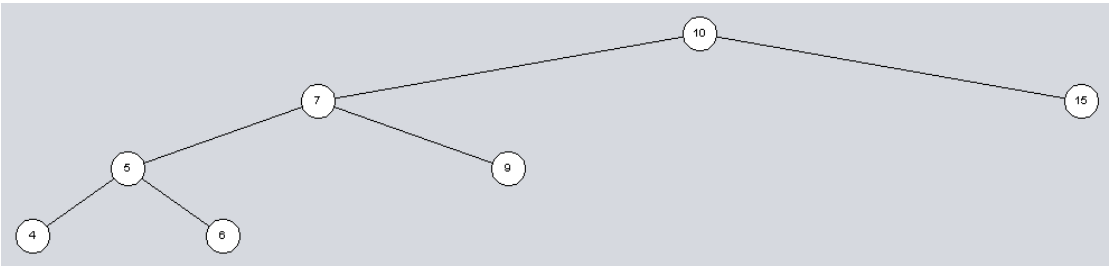


Figure 2. The BST that created by inserting 10, 7, 15, 5, 9, 5, 6 one by one

The input: are stored in the ***ex15_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 ***ex15_output.txt*** text file:

Only one line contains **Yes** if it is a full binary tree, otherwise **No**

Sample Input 1	*Sample Output 1*
7 7 9 4 1 12 6 10	No

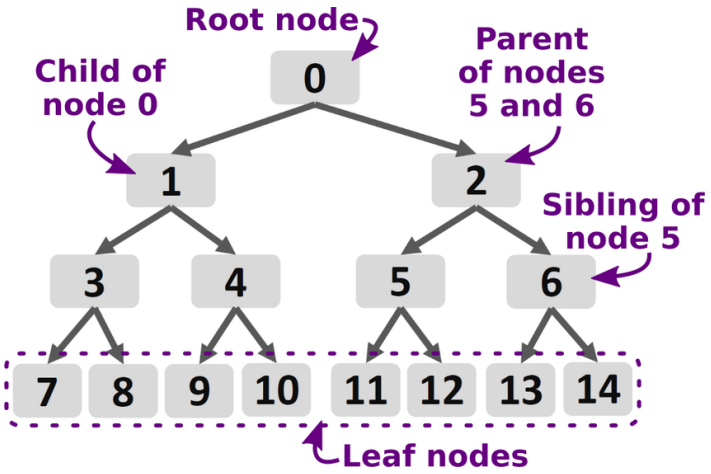
Sample Input 2	*Sample Output 2*
7 10 7 15 5 9 4 6	Yes

16. BST order

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 to print the order of all nodes in BST.

For example,



- Create a BST tree by successively adding to the tree N integer values as follows: 7, 9, 4, 1, 12, 6, 10. And This BST is not full binary tree.

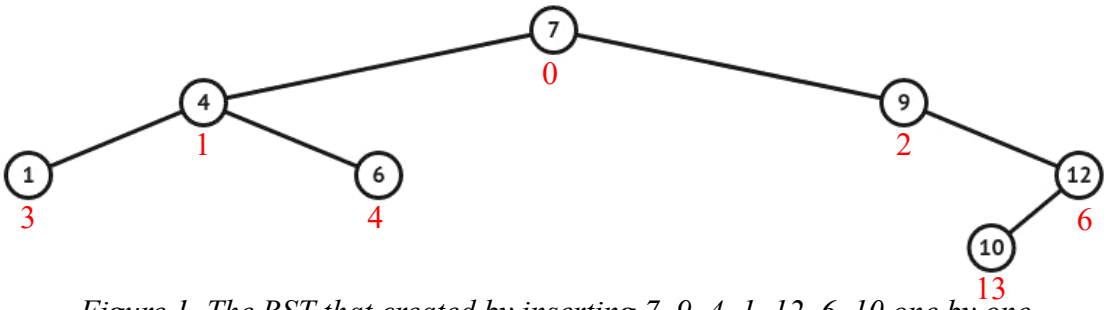


Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one

The input: are stored in the ***ex16_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 ***ex16_output.txt*** text file:

Only one line contains order of all nodes in the BST, each value with its order separated by commas. (pre-order)

Sample Input 1	*Sample Output 1*
7	7[0],4[1],1[3],6[4],9[2],12[6],10[13]

7 9 4 1 12 6 10	

Sample Input 2	*Sample Output 2*
9 10 7 6 2 18 13 2 6 4	10[0],7[1],6[3],2[7],4[16],18[2],13[5]