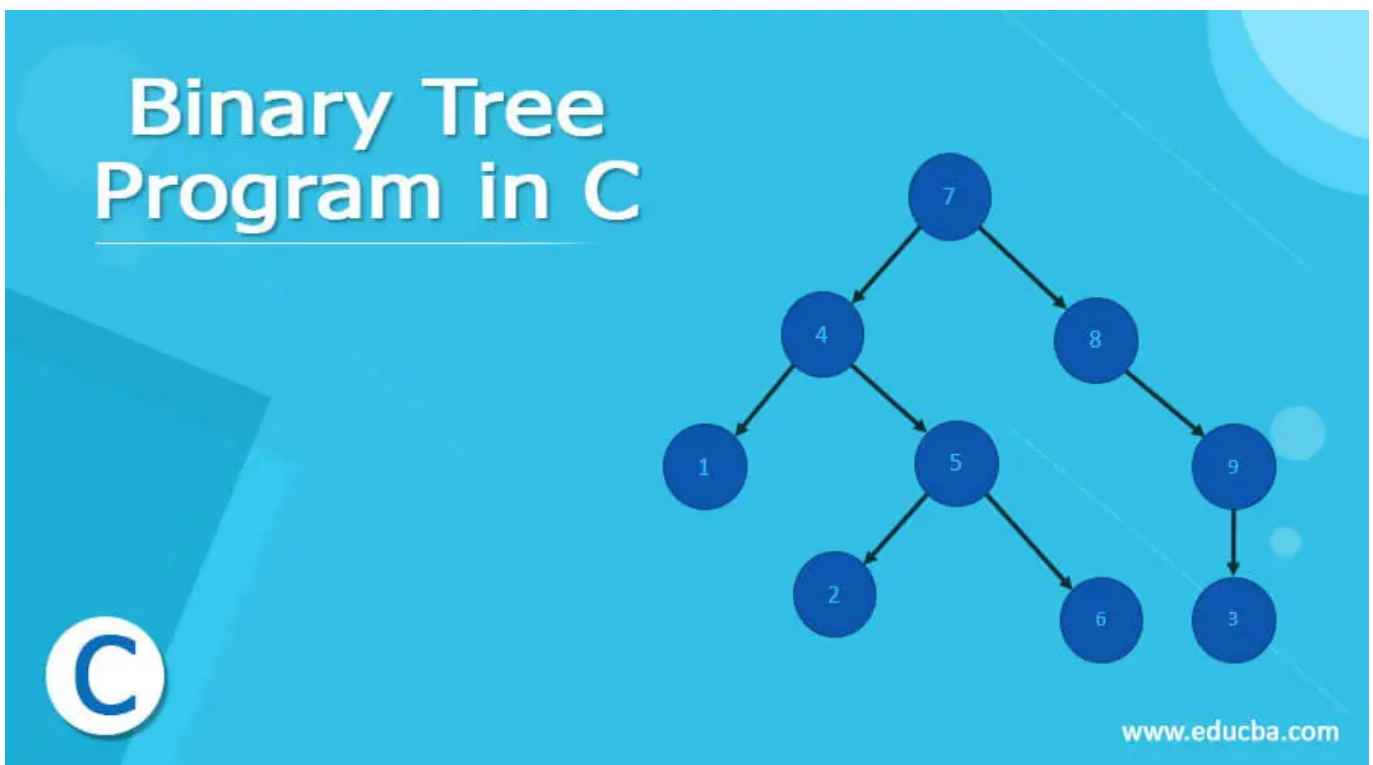




(<https://www.educba.com/software-development/>)

← (<https://www.educba.com/atm-program-in-c/>)

→ (<https://www.educba.com/string-concatenation-in-c/>)



Introduction to Binary Tree Program in C

Binary tree program in C is a nonlinear data structure used for data search and organization. Binary tree is comprised of nodes, and these nodes each being a data component, have left and right child nodes. Unlike other data structures, such as, Arrays, Stack and queue, Linked Lists, which are Linear type data structures whereas Trees are Hierarchical type of data structures. Binary search tree or BST in short, whose nodes each store keys greater than their left child nodes and less than all the right child nodes. As the data in a binary tree is organized, it allows





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

Binary tree does not have any particular Syntax but has an Algorithm to follow in implementing Binary Tree.

```
struct BT {  
    int data;  
    struct BT *rightNode, *leftNode;  
};
```

- Left sub tree of node contains nodes with keys less than node's key
- Right sub tree of node contains nodes with keys greater than node's key
- Both left and right sub tree must also be a Binary tree, and no duplicates are allowed.

Illustration of Binary Tree:





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Above properties of Binary tree provide ordering among keys such that operations like search, min, and max can be done faster. If in case there is no order, then user has to compare every key for searching a given key.

Algorithm for Binary Tree:

1. A new binary tree is created and values are assigned
2. Write a function insert() in such a way that node and key will be two parameters and check for below conditions,
 - a. If rootNode == NULL, then return new node to calling function.
 - b. If rootNode => data < keyValue, then call insert() with rootNode => rightNode and assign return value in rootNode => rightNode.
 - c. If rootNode => data > keyValue, then call insert() with rootNode => leftNode and assign return value in rootNode => leftNode
3. Then finally, we can return original rootNode pointer to calling function.

Example #1: C program for insertion in a Binary tree

```
#include<stdio.h>
#include<stdlib.h>
struct BTreeNode
{
    int keyVal;
    struct BTreeNode *leftNode;
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
struct BNode newNode = malloc(sizeof(struct BNode));
newNode->keyVal = value;
newNode->leftNode = NULL;
newNode->rightNode = NULL;
return newNode;
}

struct BNode *insert(struct BNode *rootNode, int value)
{
    if(rootNode == NULL)
        return getNode(value);
    if(rootNode->keyVal < value)
        rootNode->rightNode = insert(rootNode->rightNode,value);
    else if(rootNode->keyVal > value)
        rootNode->leftNode = insert(rootNode->leftNode,value);
    return rootNode;
}

void insertorder(struct BNode *rootNode)
{
    if(rootNode == NULL)
        return;
    insertorder(rootNode->leftNode);
    printf("%d ",rootNode->keyVal);
    insertorder(rootNode->rightNode);
}

int main()
{
```





(<https://www.educba.com/software-development/>)

```
rootNode = insert(rootNode, 1, ,
rootNode = insert(rootNode, 5);
rootNode = insert(rootNode, 2);
rootNode = insert(rootNode, 9);
rootNode = insert(rootNode, 3);
insertorder(rootNode);
return 0;
}
```

🔗 Popular Course in this category



C Programming Training (3 Courses, 5 Project)

3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access

★★★★★ 4.5 (8,635 ratings)

Course Price

\$79 ~~\$399~~

[View Course](#)

(<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner1>)

Related Courses

C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

So here we are creating a Binary tree and then inserting the node values.

Example #2: Binary tree, inorder, preorder, and postorder traversal in C programming.

```
#include <stdio.h>
#include <stdlib.h>
struct BTreeNode {
    int data;
    struct BTreeNode* leftNode;
    struct BTreeNode* rightNode;
};
void inorder(struct BTreeNode* rootNode) {
    if (rootNode == NULL) return;
    inorder(rootNode->leftNode);
    printf("%d ->", rootNode->data);
    inorder(rootNode->rightNode);
}
void preorder(struct BTreeNode* rootNode) {
    if (rootNode == NULL) return;
    printf("%d ->", rootNode->data);
    preorder(rootNode->leftNode);
    preorder(rootNode->rightNode);
}
void postorder(struct BTreeNode* rootNode) {
```





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

```
,
struct BTreeNode* createNode(value) {
    struct BTreeNode* newNode = malloc(sizeof(struct BTreeNode));
    newNode->data = value;
    newNode->leftNode = NULL;
    newNode->rightNode = NULL;
    return newNode;
}

struct BTreeNode* insertLeftNode(struct BTreeNode* rootNode, int value)
{
    rootNode->leftNode = createNode(value);
    return rootNode->leftNode;
}

struct BTreeNode* insertRightNode(struct BTreeNode* rootNode, int value)
{
    rootNode->rightNode = createNode(value);
    return rootNode->rightNode;
}

int main() {
    struct BTreeNode* rootNode = createNode(7);
    insertLeftNode(rootNode, 4);
    insertRightNode(rootNode, 8);
    insertLeftNode(rootNode->leftNode, 1);
    insertRightNode(rootNode->rightNode, 5);
    insertLeftNode(rootNode->leftNode, 6);
    insertRightNode(rootNode->rightNode, 3);
}
```





(<https://www.educba.com/software-development/>)

```
printf("\n Postorder : \n ");
postorder(rootNode);
}
```

Output:

So here we have performed inorder, preorder, and postorder traversal for a Binary tree by inserting nodes.

While searching for a value in Binary Tree, node is traversed from left to right.

Types of Binary Tree

Full Binary Tree:

Special type of Binary Tree where every parent node or an internal node has either 2 or no child nodes.

Perfect Binary Tree:

A Binary tree in which each internal node has exactly two children and all leaf nodes at same level.

Complete Binary Tree:

It is same as Full Binary Tree, but all leaf nodes must be at left and every level must have both left and right child nodes. And the last leaf node should not have the right child.





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Skewed Binary Tree:

It is similar to a pathological tree in which the binary tree is either dominated by left or right nodes. And it has two types: Left Skewed Binary tree and Right Skewed Binary Tree.

Balanced Binary Tree:

Type of Binary Tree in which difference between the height of left and right subtree for each child node is 0 or 1

Conclusion

With this, we shall conclude our topic “Binary Tree program in C”. We have seen what Binary tree is and its Algorithm. Seen few examples on How Binary tree is created, how insertion is done, and how we can search the Binary Tree nodes and display. There are also types of Binary Tree which we have explained above. There are other operations too, which can help us to understand the concept of Binary Tree completely.

Recommended Articles

This is a guide to Binary Tree Program in C. Here we discuss the Introduction, syntax, Types of Binary Tree with Examples with code implementation. You may also have a look at the following articles to learn more –

1. [Splay Tree in Data Structure \(https://www.educba.com/splay-tree-in-data-structure/\)](https://www.educba.com/splay-tree-in-data-structure/)
2. [Spanning Tree Algorithm \(https://www.educba.com/spanning-tree-algorithm/\)](https://www.educba.com/spanning-tree-algorithm/)
3. [Tree Traversal Techniques \(https://www.educba.com/tree-traversal-techniques/\)](https://www.educba.com/tree-traversal-techniques/)
4. [Binary Tree Interview Questions \(https://www.educba.com/binary-tree-interview-](https://www.educba.com/binary-tree-interview-)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

- ☒ 600+ Online Courses
- ☒ 50+ projects
- ☒ 3000+ Hours
- ☒ Verifiable Certificates
- ☒ Lifetime Access

Learn More

<https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3>

About Us

Blog (<https://www.educba.com/blog/?source=footer>)

Who is EDUCBA? (<https://www.educba.com/about-us/?source=footer>)

Sign Up (<https://www.educba.com/software-development/signup/?source=footer>)

Corporate Training (<https://www.educba.com/corporate/?source=footer>)

Certificate from Top Institutions (<https://www.educba.com/educbalive/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

Terms and Conditions (<https://www.educba.com/terms-and-conditions/?source=footer>)

Privacy Policy (<https://www.educba.com/privacy-policy/?source=footer>)

Apps

iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

Android (<https://play.google.com/store/apps/details?id=com.educba.www>)

Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

Python Tutorials (<https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer>)

All Tutorials (<https://www.educba.com/software-development/software-development-tutorials/?source=footer>)

Certification Courses

All Courses (<https://www.educba.com/software-development/courses/?source=footer>)

Software Development Course - All in One Bundle
(<https://www.educba.com/software-development/courses/software-development-course/?source=footer>)

Become a Python Developer (<https://www.educba.com/software-development/courses/python-certification-course/?source=footer>)





[\(https://www.educba.com/software-development/\)](https://www.educba.com/software-development/)

ASP.NET Course (<https://www.educba.com/software-development/courses/asp-net-course/?source=footer>)

VB.NET Course (<https://www.educba.com/software-development/courses/vb-net-course/?source=footer>)

PHP Course (<https://www.educba.com/software-development/courses/php-course/?source=footer>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

