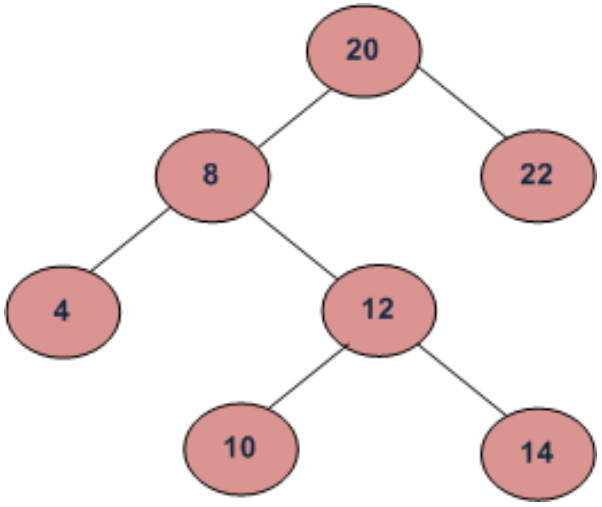
Find the node with minimum value in a Binary Search Tree

This is quite simple. Just traverse the node from root to left recursively until left is NULL. The node whose left is NULL is the node with minimum value.



For the above tree, we start with 20, then we move left 8, we keep on moving to left until we see NULL. Since left of 4 is NULL, 4 is the node with minimum value.

```
#include <stdio.h>
#include<stdlib.h>

/* A binary tree node has data, pointer to left child
   and a pointer to right child */
struct node
{
    int data;
```


Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

```
    struct node* left;
    struct node* right;
};

/* Helper function that allocates a new node
with the given data and NULL left and right
pointers. */
struct node* newNode(int data)
{
  struct node* node = (struct node*)
                      malloc(sizeof(struct node));
  node->data  = data;
  node->left  = NULL;
  node->right = NULL;

  return(node);
}

/* Give a binary search tree and a number,
inserts a new node with the given number in
the correct place in the tree. Returns the new
root pointer which the caller should then use
(the standard trick to avoid using reference
parameters). */
struct node* insert(struct node* node, int data)
{
  /* 1. If the tree is empty, return a new,
      single node */
  if (node == NULL)
    return(newNode(data));
  else
  {
    /* 2. Otherwise, recur down the tree */
    if (data <= node->data)
        node->left  = insert(node->left, data);
    else
        node->right = insert(node->right, data);

    /* return the (unchanged) node pointer */
    return node;
  }
}

/* Given a non-empty binary search tree,
return the minimum data value found in that
tree. Note that the entire tree does not need
to be searched. */
```


Popular Posts

All permutations of a given string

Memory Layout of C Programs

Understanding “extern” keyword in C

Median of two sorted arrays

Tree traversal without recursion and without stack!

Structure Member Alignment, Padding and Data Packing

Intersection point of two Linked Lists

Lowest Common Ancestor in a BST.

Check if a binary tree is BST or not

Sorted Linked List to Balanced BST

```
int minValue(struct node* node) {
  struct node* current = node;

  /* loop down to find the leftmost leaf */
  while (current->left != NULL) {
    current = current->left;
  }
  return(current->data);
}

/* Driver program to test sameTree function*/
int main()
{
  struct node* root = NULL;
  root = insert(root, 4);
  insert(root, 2);
  insert(root, 1);
  insert(root, 3);
  insert(root, 6);
  insert(root, 5);

  printf("\n Minimum value in BST is %d", minValue(root));
  getchar();
  return 0;
}
```

Time Complexity: O(n) Worst case happens for left skewed trees.

Similarly we can get the maximum value by recursively traversing the right node of a binary search tree.

References:
http://cslibrary.stanford.edu/110/BinaryTrees.html

Related Tpoics:

- Print a Binary Tree in Vertical Order | Set 2 (Hashmap based Method)
- Print Right View of a Binary Tree
- Red-Black Tree | Set 3 (Delete)
- Construct a tree from Inorder and Level order traversals
- Print all nodes at distance k from a given node
- Print a Binary Tree in Vertical Order | Set 1
- Interval Tree
- Check if a given Binary Tree is height balanced like a Red-Black Tree

Writing code in comment? Please use **ideone.com** and share the link here.

16 Comments GeeksforGeeks

Sort by Newest

Recent Comments

karthik it should have been max_wrap= max_wrap -...

Maximum circular subarray sum · 2 minutes ago

affiszerv Your example has two 4s on row 3, that's why it...

Backtracking | Set 7 (Sudoku) · 46 minutes ago

RVM Can someone please elaborate this Qs from above...

Flipkart Interview | Set 6 · 1 hour ago

Vishal Gupta I talked about as an Interviewer in general,...

Software Engineering Lab, Samsung Interview | Set 2 · 1 hour ago

@meya Working solution for question 2 of 4f2f round....

Amazon Interview | Set 53 (For SDE-1) · 1 hour ago

sandeep void rearrange(struct node *head) {...

Given a linked list, reverse alternate nodes and append at the end · 3 hours ago

DarkProtocol • 7 months ago

is it O(n) or O(h) where h is height of the tree.. Pls explain,,,

GuruSimhe ➦ DarkProtocol • 5 months ago

It is actually O(h), but in case of Left skewed tree it will be O(n) and tha

wannaC • 10 months ago

```
 int minVal(struct node *root){
if(!root)
   return -1;
if(root->left==NULL)
{
   return root->data;
}
else
   minVal (root->left);
return -1;
}
/* Paste your code here (You may delete these lines if not writing co
```


Himanshu • 10 months ago

/* here is recursive method*/
int minValue(struct node* node) {

if(node==NULL)
{

```
printf("emptry tree");
return -1;
}

if(node->left==NULL)
{
return node->data;
}
else
return minValue(node->left);

}
```

 • Reply • Share ›

amit • 11 months ago

won't work for empty binary tree..no NULL check

```
/* Paste your code here (You may delete these lines if not writing c
```

 • Reply • Share ›

abhishek08aug • a year ago

C++ code: extended from my post on: http://www.geeksforgeeks.org/l...

```
#include <iostream>
#include <stdlib.h>
using namespace std;

class tree_node {
  private:
    int data;
    tree_node * left;
    tree_node * right;
```

```
public:
  tree_node() {
    left=NULL;
    right=NULL;
  }
  void set_data(int data) {
    this->data=data;
```

see more

1 ▲ ▼ • Reply • Share ›

shailraj • a year ago

What will happen if tree is empty ?

▲ ▼ • Reply • Share ›

Ankit Sablok • a year ago

The problem is quite simple for BST's in general but can anyone tell what is th problem for binary trees in general.

```
/* Paste your code here (You may delete these lines if not writing c
```

▲ ▼ • Reply • Share ›

Pranav ➦ Ankit Sablok • 4 months ago

I think for binary tree you need to traverse the tree(in any of your choice minimum found so far.

int minValue(struct node* node) {
return minValueUtil(node, INT_MAX);
}

int minValueUtil(struct node* node, int min)

```
{
if(!node) return;

// Check left subtree first
return minValueUtil(node->left, node->data);

//Check current node
if(node->data < min)
return node->data;

// Check right subtree
return minValueUtil(node->right, node->data);
}
```


vindhya • 2 years ago

won't it work if we dont initialize current as node??
I think it will.

```
/* Paste your code here (You may delete these lines if not writing co
```


vindhya ➦ vindhya • 2 years ago

```
int minVal(struct node *root){
while(root->left!=NULL)
{
root=root->left;
}
return (root->data);
}
```

```
/* Paste your code here (You may delete these lines if not writ
```

Reply • Share ›

kartik ➚ vindhya • 2 years ago

It will crash for an empty tree. If root is NULL, then this code wil cause the crash.

Reply • Share ›

rohit ➚ kartik • a year ago

but now current also points where root point if the bst is current is also null so null->left again leads to crash .ho crash..plss explain..

Reply • Share ›

Pranay Singhania • 2 years ago

Can you tell me if the minimum of the tree is not in the left sub tree of root ther you entered like this 1 2 3 4 5 then also your program will not work.

Reply • Share ›

kartik ➚ Pranay Singhania • 2 years ago

If the left subtree is empty, then the minimum value will be the root valu current as node. It will work for all cases in a non empty BST.

Reply • Share ›

Prashanth • 2 years ago

Hi,

If the Binary Search Tree is balanced. The we can find the node with maximur O(H) time, where H is the height of the tree.

1 Reply • Share ›