GeeksforGeeks
A computer science portal for geeks

GeeksQuiz

Login

Home Algorithms DS GATE Interview Corner Q&A C C++ Java Books Contribute Ask a Q About

Array Bit Magic C/C++ Articles GFacts Linked List MCQ Misc Output String Tree Graph

Floor and Ceil from a BST

There are numerous applications we need to find floor (ceil) value of a key in a binary search tree or sorted array. For example, consider designing memory management system in which free nodes are arranged in BST. Find best fit for the input request.

Ceil Value Node: Node with smallest data larger than or equal to key value.

Imagine we are moving down the tree, and assume we are root node. The comparison yields three possibilities,

A) Root data is equal to key. We are done, root data is ceil value.

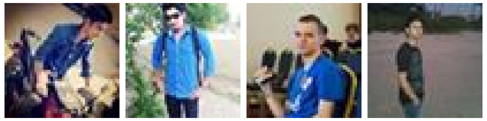
B) Root data < key value, certainly the ceil value can’t be in left subtree. Proceed to search on right subtree as reduced problem instance.

C) Root data > key value, the ceil value *may be* in left subtree. We may find a node with is larger data than key value in left subtree, if not the root itself will be ceil node.

Here is code in C for ceil value.

```
// Program to find ceil of a given value in BST
#include <stdio.h>
#include <stdlib.h>

/* A binary tree node has key, left child and right child */
struct node
{
    int key;
    struct node* left;
    struct node* right;
};
```



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

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

// Function to find ceil of a given input in BST. If input is more
// than the max key in BST, return -1
int Ceil(node *root, int input)
{
    // Base case
    if( root == NULL )
        return -1;

    // We found equal key
    if( root->key == input )
        return root->key;

    // If root's key is smaller, ceil must be in right subtree
    if( root->key < input )
        return Ceil(root->right, input);

    // Else, either left subtree or root has the ceil value
    int ceil = Ceil(root->left, input);
    return (ceil >= input) ? ceil : root->key;
}

// Driver program to test above function
int main()
{
    node *root = newNode(8);

    root->left = newNode(4);
    root->right = newNode(12);

    root->left->left = newNode(2);
    root->left->right = newNode(6);

    root->right->left = newNode(10);
    root->right->right = newNode(14);
```

Geometric Algorithms

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

```
    for(int i = 0; i < 16; i++)
        printf("%d  %d\n", i, Ceil(root, i));

    return 0;
}
```

Output:

```
0  2
1  2
2  2
3  4
4  4
5  6
6  6
7  8
8  8
9  10
10  10
11  12
12  12
13  14
14  14
15  -1
```

Exercise:

1. Modify above code to find floor value of input key in a binary search tree.

2. Write neat algorithm to find floor and ceil values in a sorted array. Ensure to handle all possible boundary conditions.

— **Venki**. Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Better Than Hadoop.
HPCC Systems is Big Data Processing and Analytics
Open Source. Proven. Trusted.
LexisNexis
Learn More

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

0 Tweet 0 1

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

16 Comments **GeeksforGeeks**

Sort by Newest ▾

695 Subscribe

Recent Comments

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

Backtracking | Set 7 (Sudoku) · 35 minutes ago

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

Flipkart Interview | Set 6 · 55 minutes ago

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

Software Engineering Lab, Samsung Interview | Set 2 · 55 minutes 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 · 2 hours ago

Neha I think that is what it should return as, in...

Find depth of the deepest odd level leaf node · 3 hours ago

Join the discussion…

devil • 4 months ago

I think this is much better approach.
Do inorder traverasl and update Ceil & Floor as we iterate.
http://ideone.com/oswEpn

Reply • Share ›

Piyush Kapoor • 7 months ago

Your code doesn't even compile !!

Reply • Share ›

srb • 10 months ago

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

int floor(node *root, int input)
{
    // Base case
    if( root == NULL )
        return -1;

    // We found equal key
    if( root->key == input )
        return root->key;

    // If root's key is bigger, floor must be in left subtree
    if( root->key> input )
        return floor(root->left, input);

    // Else, either right subtree or root has the floor value
    int floor = floor(root->right, input);
    return (floor <= input && floor!=-1) ? floor : root->key;
  }
  */
```

1 ^ | ˅ • Reply • Share ›

abhishek08aug • 11 months ago

Intelligent :D

^ | ˅ • Reply • Share ›

Amit • a year ago

```
 /* Paste your code here (You may delete these lines if not writing c
#include<stdio.h>
#include <limits.h>
#include<stdlib.h>
int floor = 0;
int ceil = 0;
struct node{
       int data;
       struct node *left;
       struct node *right;
};
struct node * newNode(int data ){
       struct node * node = (struct node *)malloc(sizeof(struct node)
       node->left = NULL;
       node->data = data;
       node->right = NULL;
       return node;
}
```

see more

Reply · Share ›

Nomind · a year ago

```
#include
#include
#include

/* A binary tree node has key, left child and right child */
struct node
{
int key;
struct node* left;
struct node* right;
};

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

see more

Reply · Share ›

Paparao Veeragandham · a year ago

```
    int ceil = INT_MIN;
    if(root->data > input) //with-out this condition it will go left-mos
   ceil = Ceil(root->left, input);
```

Reply · Share ›

kiran • 2 years ago

Good post !!! Just with minimal change in BST algo :-)

^ | v • Reply • Share ›

Venki ➦ kiran • 2 years ago

@Kiran, it is simple algorithm. We are just using the binary search tree However, this approach leads to linear search when the tree is skewed binary search tree.

Try yourself to write bug free code for the same algorithms on sorted a conditions.

^ | v • Reply • Share ›

a2 • 2 years ago

Could you please explain me why this has been added in the return statement Won't ceil always be less than root-key since it is in the left subtree ?

^ | v • Reply • Share ›

a2 ➦ a2 • 2 years ago

> ```
> ceil < root->key
> ```

^ | v • Reply • Share ›

Kartik ➦ a2 • 2 years ago

Thanks for sharing your thoughts. Your point is valid, we have r

^ | v • Reply • Share ›

Meghanath Macha ➦ Kartik • a year ago

Hey Ceil / Floor is similar to Successor / Predecessor r

^ | v • Reply • Share ›

spandan ➦ Meghanath Macha • a year ago

Yes

^ | ⌄ • Reply • Share ›

Kamal • 2 years ago

Cool! the floor program will be similar, only left right conditions will be reversec

^ | ⌄ • Reply • Share ›

vishal ➦ Kamal • 2 years ago

can you explain how conditions are changed and what is output

^ | ⌄ • Reply • Share ›

✉ Subscribe   Add Disqus to your site