Algorithms @tutorialhorizon

Home
IDE
Site Map
Contribute
Arrays
Linked List
Recursion
DP
Backtracking
Binary Tree
Trees
Difficulty Level
Beginner
Intermediate
Expert
Interviews
Amazon Questions
Goldman Sachs
Google Interview
Microsoft Interview
Apache
NetApp
Adobe
Facebook
NVIDIA
Oracle

MISC
Strings
Sorting
Hash Table
Search

MORE PROBLEMS

Prim's – Minimum Spanning Tree (MST) |using Adjacency List and Priority Queue with…

Least Recently Used (LRU) Cache – Using HashMap and Doubly Linked List | Set 1

Djkstra's – Shortest Path Algorithm (SPT)

Graph – Depth First Search using Recursion

Merge K sorted Linked List - Using Priority Queue

Stack Data Structure – Introduction and Implementation

Find the number of distinct Islands OR connected components.

Dijkstra Algorithm Implementation – TreeSet and Pair Class

Graph – Find Cycle in Undirected Graph using Disjoint Set (Union-Find)

Number of Islands using BFS

Binary Search Tree (BST) Complete Implementation.

Binary Tree : A data structure in which we have nodes containing data and two references to other nodes, one on the left and one on the right.

Binary Tree consist of Nodes

- Nodes are nothing but objects of a class and each node has data and a link to the left node and right node.
- Usually we call the starting node of a tree as *root*.

Prim's – Minimum Spanning Tree (MST) |using Adjacency List and Min Heap

Number of Islands

Dijkstra's – Shortest Path Algorithm (SPT) – Adjacency List and Priority Queue –…

Graph Implementation – Adjacency List - Better| Set 2

Max Flow Problem - Ford-Fulkerson Algorithm

K-Means Algorithm

Graph – Print all paths between source and destination

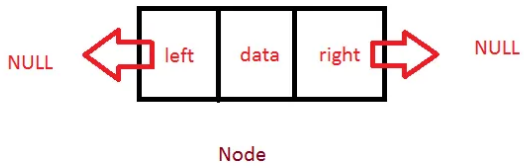
K-Means Algorithm

Find the increasing OR decreasing point in an array

Prim's Algorithm - Minimum Spanning Tree (MST)

```
class Node{
        int data;
        Node left;
        Node right;
        public Node(int
data){
                this.data =
data;
                left =
null;
                right =
null;
        }
}
```

- Left and right node of a Leaf node points to NULL so you will know that you have reached to the end of the tree.



TOP POSTS & PAGES

Binary Search Tree (BST) Complete Implementation.

RECENT POSTS

Find all unique combinations of numbers (from 1 to 9) with sum to N

Number of Intervals in which given value lies

Unique Integers in array that sum up to zero.

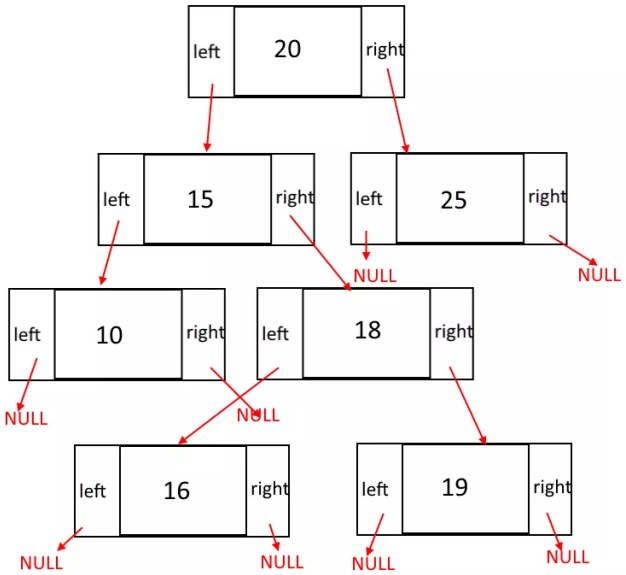
Activity Selection Problem

Find the sum of overlapping elements in two sets

Binary Search Tree:

Often we call it as BST, is a type of Binary tree which has a special property.

Nodes smaller than root goes to the left of the root and Nodes greater than root goes to the right of the root.



Operations:

Insert(int n) : Add a node the tree with value n. Its O(lgn)

AVL Tree - Insertion

Inorder Predecessor and Successor in Binary Search Tree

Dynamic Programming - Longest Common Substring

Dynamic Programming - Coin Change Problem

Convert Infix to Postfix Expression

Graph Implementation – Adjacency List - Better| Set 2

Print all subarrays of a given array

Home - All Articles

Dynamic Programming - Subset Sum Problem

DIFFICULTY LEVEL

Beginner

Intermediate

Expert

Given two coordinates, Print the line equation

Check if Graph is Bipartite – Adjacency List using Breadth-First Search(BFS)

Articulation Points OR Cut Vertices in a Graph

Print all middle elements of the given matrix/2D array.

Check If Given Undirected Graph is a tree

TOP COMPANIES

Google

Microsoft

Amazon

Facebook

More…

Find(int n) : Find a node the tree with value n. Its O(lgn)

Delete (int n) : Delete a node the tree with value n. Its O(lgn)

Display(): Prints the entire tree in increasing order. O(n).

Detail Explanations for the Operations:

Find(int n):

- Its very simple operation to perform.
- start from the root and compare root.data with n
- if root.data is greater than n that means we need to go to the left of the root.
- if root.data is smaller than n that means we need to go to the right of the root.
- if any point of time root.data is equal to the n then we have found the node, return true.
- if we reach to the leaves (end of the tree) return false, we didn't find the element

BONUS PROBLEMS

Separate 0's and 1's in a given array

Longest substring with at most K unique characters

Rearrange Positive and Negative Numbers of Array On Each Side in O(nlogn)

Find the sum of overlapping elements in two sets

Monotone Increasing Digits

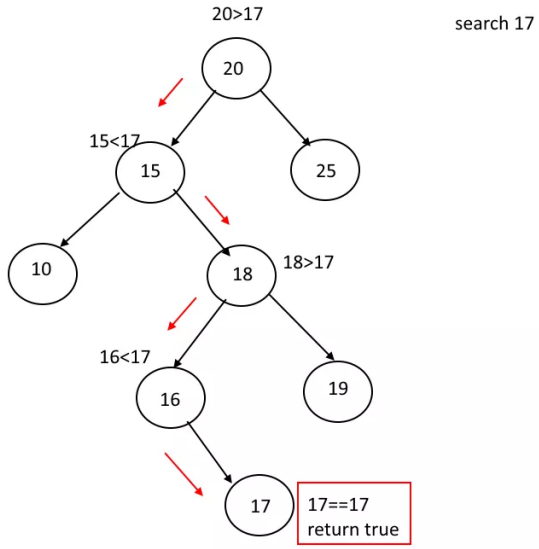
Check if Array is Consecutive Integers

Reverse a Stack using recursion – In Place (Without using extra memory)

Double Threaded Binary Tree Co

Inc an Bi

AV



SUBSCRIBE (NO SPAMS!!)

Enter your email address to subscribe to this blog and receive notifications of new posts by email.

Join 1,357 other subscribers

Email Address

Subscribe

Insert(int n):

- Very much similar to find() operation.
- To insert a node our first task is to find the place to insert the node.
- Take current = root .
- start from the current and compare root.data with n
- if current.data is greater than n that means we need to go to the left of the root.
- if current.data is smaller than n that means we need to go to the right of the root.

Track the Maximum Element in a Stack.

Dynamic Programming – Egg Dropping Problem

Dynamic Programming – Count all paths in 2D Matrix with Obstructions in it

CATEGORIES

Activity Selection (1)

Adobe (59)

Airbnb (13)

Amazon Questions (201)

Apache (35)

Apple (12)

Arrays (147)

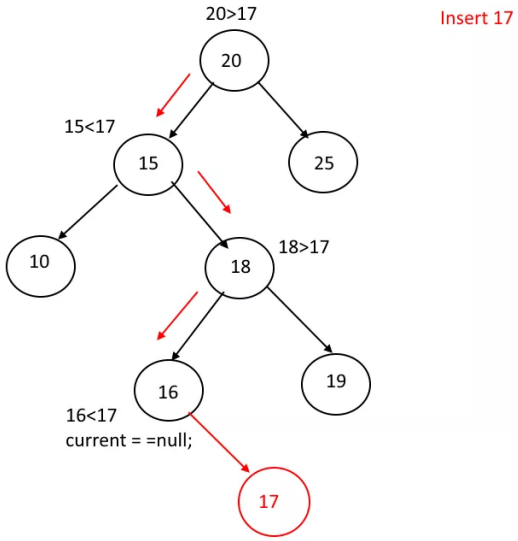
Backtracking (9)

Beginner (184)

Binary Tree (76)

Bit Manipulation (22)

- if any point of time current is null that means we have reached to the leaf node, insert your node here with the help of parent node. (See code)



Delete(int n):

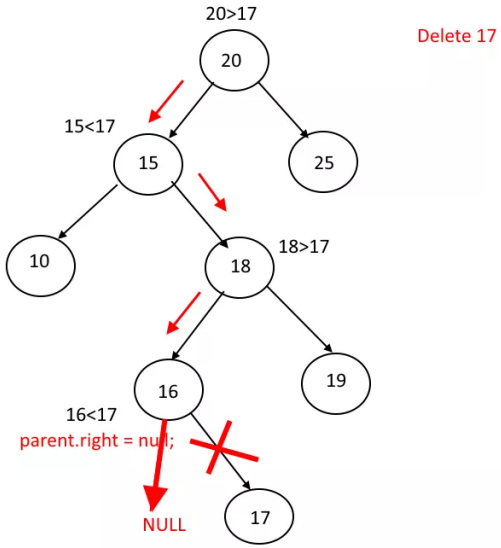
Complicated than Find() and Insert() operations. Here we have to deal with 3 cases.

- Node to be deleted is a leaf node (No Children).
- Node to be deleted has only one child.
- Node to be deleted has two childrens.

Bloomberg (24)
Deque (3)
Difficulty Level (45)
Divide and Conquer (12)
Dynamic Programming (31)
Epic Systems (33)
Expedia (13)
Expert (96)
Expressions (12)
Facebook (59)
Flipkart (26)
Goldman Sachs (16)
Google Interview (79)
Graphs (53)
Greedy Algorithm (6)
Groupon (18)
Hash Table (9)

Node to be deleted is a leaf node (No Children).

its a very simple case, if a node to be deleted has no children then just traverse to that node, keep track of parent node and the side in which the node exist(left or right) and set *parent.left = null or parent.right = null;*

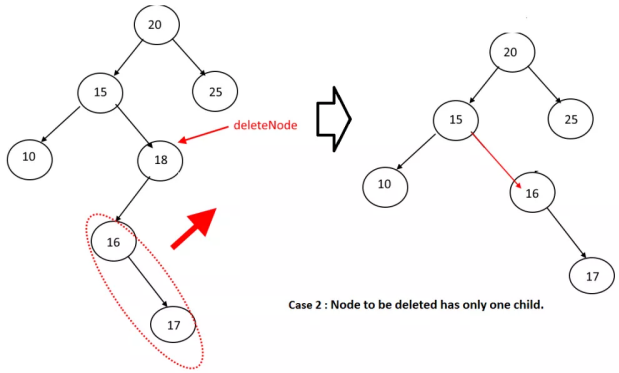


Case 1 : Node to be deleted is a leaf node (No Children).

Node to be deleted has only one child.

HCL (10)
Heaps (4)
Hulu (13)
Informatica (17)
Intermediate (174)
Java Concepts (9)
Linked List (29)
LinkedIn (18)
Machine Learning (2)
Maths (32)
Microsoft Interview (128)
MISC (32)
NetApp (26)
Numbers (43)
NVIDIA (16)
Oracle (26)
Pair (2)

1. its a slightly complex case. if a node to be deleted(deleteNode) has only one child then just traverse to that node, keep track of parent node and the side in which the node exist(left or right).
2. check which side child is null (since it has only one child).
3. Say node to be deleted has child on its left side . Then take the entire sub tree from the left side and add it to the parent and the side on which deleteNode exist, see step 1 and example.



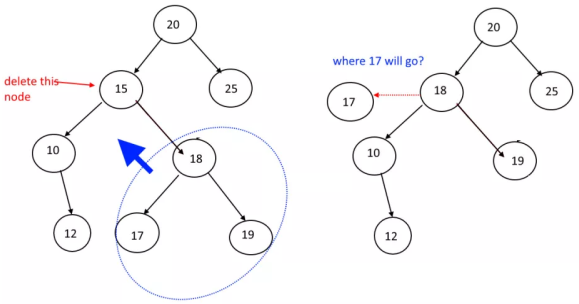
Node to be deleted has two children.

Now this is quite exciting ☺

You just cannot replace the deleteNode with any of its child, Why? Lets try out a

Position (1)
Priority Queue (9)
Qualcomm (16)
Queue (7)
Recursion (133)
Samsung (4)
Scheduling/Intervals (4)
Search Algorithms (10)
Software Development Engineer (SDE) (250)
Software Development Engineer in Test (111)
Software Engineer (251)
Software Engineer In Test (121)
Sorting Techniques (18)
Stack (21)
Strings (49)
Test Enginner (73)

example.



What to do now?????

Dont worry we have solution for this ☺

Find The Successor:

Successor is the node which will replace the deleted node. Now the question is to how to find it and where to find it.

Successor is the smaller node in the right sub tree of the node to be deleted.

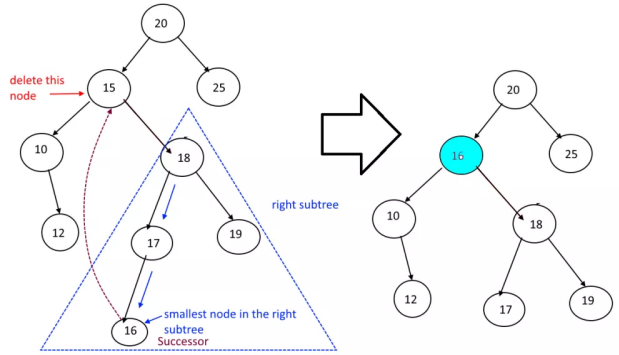
Top Companies (82)

Trees (62)

Twitter (23)

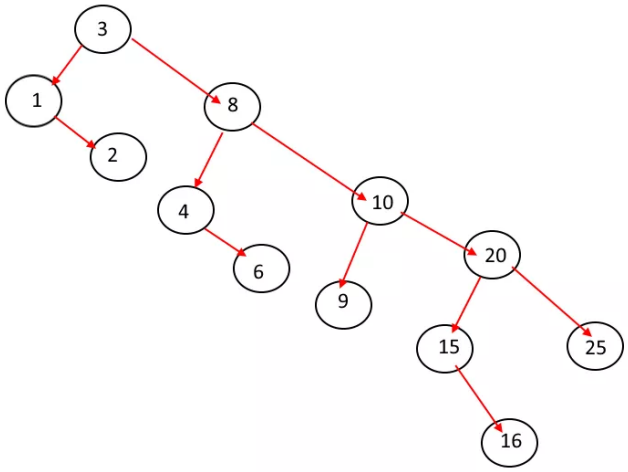
Uncategorized (8)

Yahoo (51)



Display() : To know about how we are displaying nodes in increasing order, Click Here

Complete Example :



Complete Example Code:

Run This Code

```
1  public class BinarySearc
2          public static  N
3          public BinarySea
4                  this.roo
5          }
6
7          public boolean f
8                  Node cur
9                  while(cu
10
11
12
13
14
15
16
17                 }
18                 return f
19         }
20         public boolean d
21                 Node par
22                 Node cur
23                 boolean
24                 while(cu
25
26
27
28
29
30
31
32
33
34
```

```
35
36                  }
37                  //if i a
38                  //Case 1
39                  if(curre
40
41
42
43
44
45
46
47
48                  }
49                  //Case 2
50                  else if(
51
52
53
54
55
56
57
58                  }
59                  else if(
60
61
62
63
64
65
66
67                  }else if
68
```

```
69
70
71
72
73
74
75
76
77
78
79                  }
80                  return t
81          }
82
83          public Node getS
84                  Node suc
85                  Node suc
86                  Node cur
87                  while(cu
88
89
90
91                  }
92                  //check
93                  // if it
94  //              successs
95                  if(succe
96
97
98                  }
99                  return s
100         }
101         public void inse
102                 Node new
```

```
103                 if(root=
104
105
106                 }
107                 Node cur
108                 Node par
109                 while(tr
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124                 }
125         }
126         public void disp
127                 if(root!
128
129
130
131                 }
132         }
133         public static vo
134                 BinarySe
135                 b.insert
136                 b.insert
```

```
137                 b.insert
138                 System.o
139                 b.displa
140                 System.o
141                 System.o
142                 System.o
143                 b.displa
144                 System.o
145                 b.displa
146                 System.o
147                 b.displa
148         }
149 }
150
151 class Node{
152         int data;
153         Node left;
154         Node right;
155         public Node(int
156                 this.dat
157                 left = n
158                 right =
159         }
160 }
```

BinarySearchTree.java hosted with ♡ by **GitHub** **view raw**

Output:

```
Original Tree :
1 2 3 4 6 8 9 10 15 16 20
25
```

```
Check whether Node with
value 4 exists : true
Delete Node with no
children (2) : true
1 3 4 6 8 9 10 15 16 20 25
Delete Node with one child
(4) : true
1 3 6 8 9 10 15 16 20 25
Delete Node with Two
children (10) : true
1 3 6 8 9 15 16 20 25
```

Top Companies Interview Questions..-

Google Microsoft Amazon Facebook mo

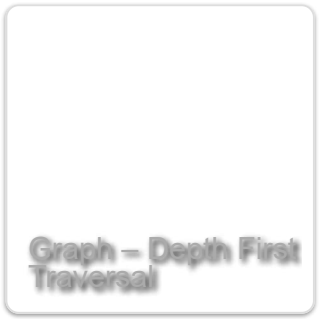
If you find anything incorrect or you feel that there is any better approach to solve the above problem, please write comment.

Related Posts:

Check the completeness of given binary tree | Set 1 - Using Node Count

Check if the given binary tree is Full or not.


Disjoint Set Data Structure - Union Find Algorithm


Graph – Depth First Traversal


Heap Sort – Java Implementation


Check the completeness of given binary tree | Set 2 - Using Level Order Traversal


Count the number of nodes in a given binary tree

Share this:

LinkedIn Print Tumblr Facebook

Twitter Email Reddit

Tags: Beginner

YOU MAY ALSO LIKE...

Convert BST to Greater Sum Tree	Binary Tree-Postorder Traversal – Non Recursive Approach	Longest contiguous character in a given String – O(N) Solution

delete O(N) Θ(logN),
find O(N) Θ(logN),
display O(N) Θ(N)

AVL trees have the complexity you said above...

also we usually using logN instead of lgN to express the time complexity.

4 ^ | v • Reply • Share ›

Akash Kumar • a year ago

better solution here
https://www.latestudy.in

^ | v • Reply • Share ›

Ibrahim Babangida • 3 years ago

QUESTION:

A company wishes to keep
record of its products updated, sorted and capable of being search rapidly.
Each product is described by Prod Id, Prod description quality, unit price. If each product details are to be kept in a Binary search tree, write:

1) A
binary search tree declaration that can be used to store product information.

2) A
method called insert() that can be used to insert a product record in a tree

3) A
method called delete() that can be used to delete a product

4) A
method called update () to update a
product detail.

5) Test
all your method in a program.
please i need help on this.
thanks

Reply • Share ›

Nkululeko Rwaxa • 3 years ago

About the 3rd case on Delete(), is the
Successor not 12 instead of 16 ?
I think it is 12 though

Reply • Share ›

Ta Anh Tu • 3 years ago

Thanks. This post is very useful for me!

Reply • Share ›

Suchith S Pillai • 3 years ago

For Delete, No children
if(current.left==null && current.right==null)
{ current = null;} will do the deletion right ?
Or we need the if conditions given ?

Reply • Share ›

Eldo Joseph ➚ Suchith S Pillai
• 3 years ago

Nope. current's parent should set
either parent.left = null or
parent.right = null.
setting current = null would only
nullify the value in your current
pointer, this does nothing in the
tree.

Reply • Share ›

Suchith S Pillai ➚ Eldo Joseph • 3 years ago

Please correct my understanding: Current Indicates a child node. So it will be either parent.left==current or parent.right==current, Since it is an existing BST. SO what is the difference between parent.left = null and current = null, since both referes to the same object.

^ | v • Reply • Share ›

Eldo Joseph ➚ Suchith S Pillai • 3 years ago

when u set current = null, u r just removing the reference of the current variable to the child node. Notice that, after doing this the node is actually in memory and only the current pointer's reference to it has been removed. But the parent pointer is still pointing to that node.

(In garbage

collected languages, like java and python, since the parent is still pointing to the node, it won't be removed. I don't know if c works differently though)

Reply • Sha

Prashant Bisht • 3 years ago

I dont understand what is the significance of parent=current in insert .I think it will work without this line also.

Reply • Share ›

Rotem Uzan • 3 years ago

Hi,
I have a question, in the Delete() method, why we can't just use the insert method and run it on the children of the deleted node ?

Home
IDE
Site Map
Contribute
Arrays
Linked List
Recursion
DP

- Backtracking
- Binary Tree
- Trees
- Difficulty Level
- Beginner
- Intermediate
- Expert
- Interviews
- Amazon Questions
- Goldman Sachs
- Google Interview
- Microsoft Interview
- Apache
- NetApp
- Adobe
- Facebook
- NVIDIA
- Oracle
- MISC
- Strings
- Sorting
- Hash Table
- Search