



Coursary



Data Analysis Courses

[Home](#)[IDE](#)[Difficulty Level](#) ▾[Interviews](#) ▾

Sorted Array to Binary Search Tree of Minimal Height

September 27, 2014 by Sumit Jain

Objective: Given a sorted array with unique elements, Create a binary search tree with minimal height.

Why minimal height is important :

We can do the linear scan to the array and make the first element as root and insert all other elements into the tree but in that case tree will be **skewed**, which means all the nodes of the tree will be on the one side of the root so the height of the tree will be equal to the number of elements in the array. So here our objective is to keep the tree balanced as much as possible.

What is balanced Tree: A balanced tree is a tree in which difference between heights of sub-trees of any node in the tree is not greater than one. To read more about balanced tree, click

More Problems

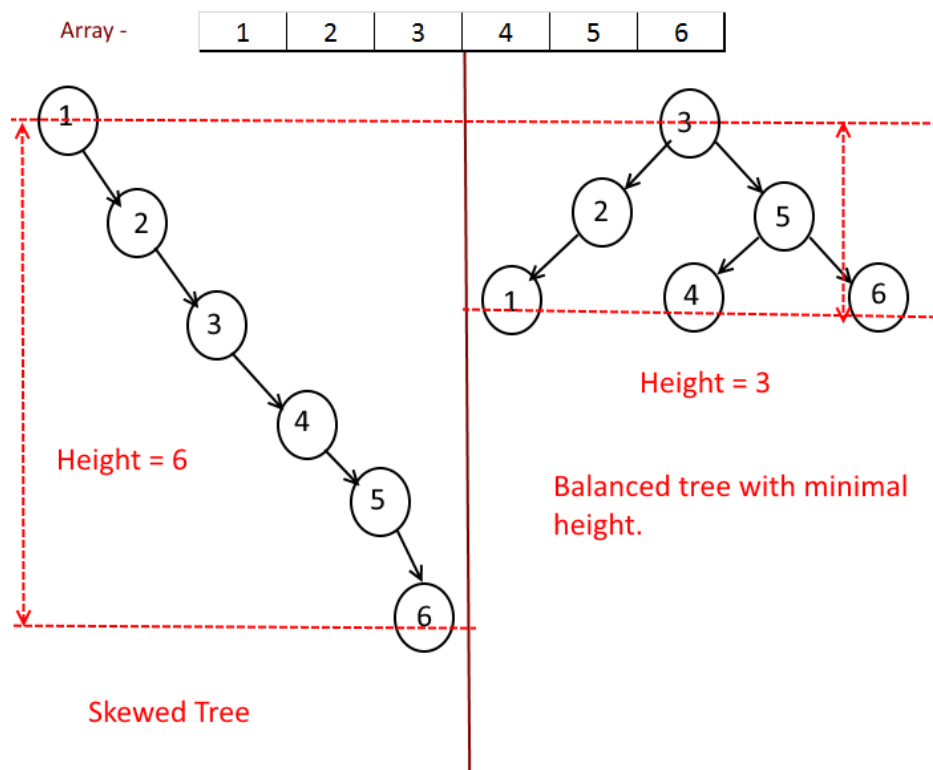
Replace array elements with maximum element on the right.

here

Input: A one dimensional array

Output: Binary Search Tree of Minimal Height

Example:



Sorted Array To BST Example

Approach:

Recursion:

Check If Given Undirected Graph is a tree

Minimum Increments to make all array elements unique

Convert to Non-decreasing Array with one change

Minimum Deletions to make the occurrence of each character unique.

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

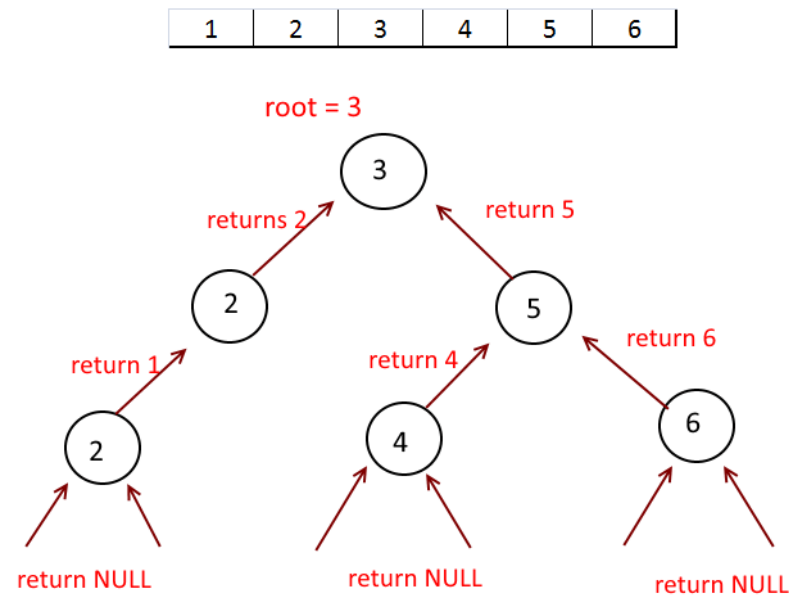
Given an array, find all unique subsets with a given sum with allowed repeated digits.

Sort a given stack - Using Recursion

Maximum number edges to make Acyclic Undirected/Directed Graph

- Get the middle of the array
- make it as root. *(By doing this we will ensure that half of the elements of array will be on the left side of the root and half on the right side.)*
- Take the left half of the array, call recursively and add it to root.left.
- Take the right half of the array, call recursively and add it to root.right.
- return root.

Given an array, print all unique subsets with a given sum.



Array To BST Recursion

Complete Code:

Run This Code

```
1 public class SortedArrayToBST {
2     public BSTNode convert(int [] arrA, int start, int end){
3         if(start>end){
4             return null;
5         }
6         int mid = (start + end)/2;
7         BSTNode root = new BSTNode(arrA[mid]);
8         root.left = convert(arrA, start, mid-1);
9         root.right =convert(arrA, mid+1, end);
10        return root;
11    }
12    public void displayTree(BSTNode root){
13        if(root!=null){
14            displayTree(root.left);
15            System.out.print(" " + root.data);
16            displayTree(root.right);
17        }
18    }
19    public static void main(String args[]){
20        int [] arrA = {2,3,6,7,8,9,12,15,16,18,20};
21        SortedArrayToBST s = new SortedArrayToBST();
22        BSTNode x = s.convert(arrA, 0, arrA.length-1);
23        System.out.println("Tree Display : ");
24        s.displayTree(x);
25    }
26 }
27 class BSTNode{
28     int data;
29     BSTNode left;
30     BSTNode right;
31     public BSTNode(int data){
32         this.data = data;
33         left = null;
```



Subscribe (No Spam!!)

```
34  right = null;  
35  }  
36  }
```

[view raw](#)

SortedArrayToBST.java

hosted with ❤ by GitHub

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

Output:

Tree Display :

2 3 6 7 8 9 12 15 16 18 20

Related Posts:

- [Print sorted unique elements of a given array](#)
- [Count the number of nodes in a given binary tree](#)
- [Check the completeness of given binary tree | Set 1 - Using Node Count](#)
- [Check the completeness of given binary tree | Set 2 - Using Level Order Traversal](#)
- [Check if the given binary tree is Full or not.](#)
- [Linear Search vs Binary Search](#)
- [Check if array contains all unique or distinct numbers.](#)
- [Sort the indexes of the array as per the elements of the array](#)
- [Sum of all Unique elements in a given array](#)
- [Sort the two dimensional \(2D\) array - In-place](#)
- [Replace array elements with maximum element on the right.](#)
- [Check If Given Undirected Graph is a tree](#)

Recent Posts

Find an extra element in two almost similar arrays

find the Arithmetic Progression sequence

Find the Nth-term in a given arithmetic progression

Find The Minimum time difference

Departure and Destination Cities in a given itinerary

- [Minimum Increments to make all array elements unique](#)
- [Convert to Non-decreasing Array with one change](#)
- [Minimum Deletions to make the occurrence of each character unique.](#)

📁 Amazon Questions, Arrays, Binary Tree, Divide and Conquer, Intermediate, Recursion, Software Development Engineer (SDE), Software Development Engineer in Test, Software Engineer, Software Engineer In Test

🔑 Intermediate

- < In a Binary Tree, Create Linked Lists of all the nodes at each depth.
- > Inorder Successor in Binary Tree

Given an array, rank its elements

Find Three Consecutive Odd Numbers in an array

Convert to Non-decreasing Array with one change

In an array, Duplicate the zeroes without expanding it

Maximum Depth of Valid Nested Parentheses in an arithmetic expression

