### Array to BST

**Easy**

Given a sorted array. Convert it into a Height balanced Binary Search Tree (BST). Find the preorder traversal of height balanced BST. If there exist many such balanced BST consider the tree whose preorder is lexicographically smallest.  
Height balanced BST means a binary tree in which the depth of the left subtree and the right subtree of every node never differ by more than 1.

**Example 1:**

**Input:** nums = {1, 2, 3, 4}

**Output:** {2, 1, 3, 4}

**Explanation:**

The preorder traversal of the following

BST formed is {2, 1, 3, 4}:

  2

  / \

1 3

  \

  4

**Example 2:**

**Input:** nums = {1,2,3,4,5,6,7}

**Ouput:** {4,2,1,3,6,5,7}

**Explanation:**

The preorder traversal of the following

BST formed is {4,2,1,3,6,5,7} :

4

/ \

2 6

/ \ / \

1 3 5 7

**Expected Time Complexity:**O(n)  
**Expected Space Complexity:**O(n)

**Constraints:**  
1 ≤ |nums| ≤ 104  
-104 ≤ nums[i] ≤ 104

**Company Tags**

[**VMWare**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=VMWare) [**Amazon**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Amazon) [**Microsoft**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Microsoft) [**Snapdeal**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Snapdeal) [**Adobe**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Adobe) [**Cisco**](https://practice.geeksforgeeks.org/explore/?company%5b%5d=Cisco)

//{ Driver Code Starts

//Initial Template for Java

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class CodingMaxima

{

public static void main(String[] args) throws IOException

{

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int T = Integer.parseInt(br.readLine().trim());

while(T-->0)

{

int n = Integer.parseInt(br.readLine().trim());

int[] nums = new int[n];

String[] S = br.readLine().trim().split(" ");

for(int i = 0; i < n; i++){

nums[i] = Integer.parseInt(S[i]);

}

Solution obj = new Solution();

int[] ans = obj.sortedArrayToBST(nums);

for(int i = 0; i < ans.length; i++)

System.out.print(ans[i] + " ");

System.out.println();

}

}

}

// } Driver Code Ends

public class Node{

int data;

Node left;

Node right;

public Node(int data){

this.data=data;

left=null;

right=null;

}

}

class Solution

{

public int[] sortedArrayToBST(int[] nums)

{

if(nums==null)

return nums;

Node root=making\_tree(nums, 0, nums.length-1);

ArrayList<Integer> ar=new ArrayList<>();

preorder(root, ar);

int[] temp=new int[nums.length];

for(int i=0;i<ar.size();i++){

temp[i]=ar.get(i);

}

return temp;

}

public Node making\_tree(int[] nums, int start, int end){

if(start>end)

return null;

int mid=start+(end-start)/2;

Node root=new Node(nums[mid]);

root.left=making\_tree(nums, start,mid-1);

root.right=making\_tree(nums,mid+1, end);

return root;

}

private void preorder(Node root, ArrayList<Integer> ar){

if(root==null)

return ;

ar.add(root.data);

preorder(root.left, ar);

preorder(root.right,ar);

}

}