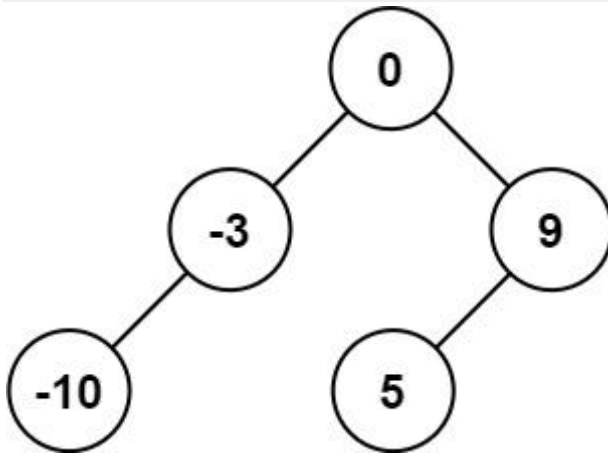


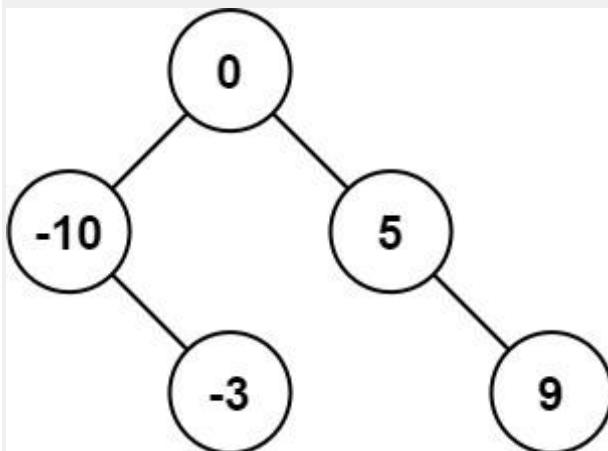
Given an integer array `nums` where the elements are sorted in ascending order, convert *it to a Height-balanced binary search tree*. Example 1:



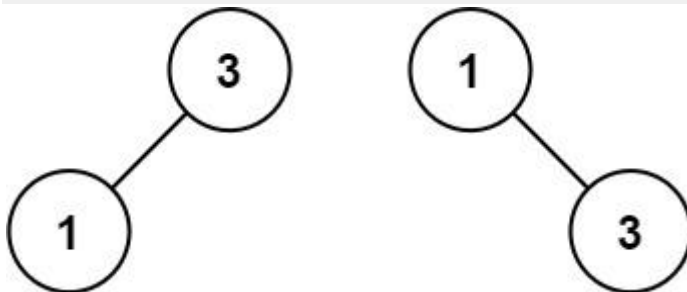
Input: `nums = [-10,-3,0,5,9]`

Output: `[0,-3,9,-10,null,5]`

Explanation: `[0,-10,5,null,-3,null,9]` is also accepted:



Example 2:



Input: nums = [1,3]

Output: [3,1]

Explanation: [1,null,3] and [3,1] are both height-balanced BSTs.

Constraints:

- $1 \leq \text{nums.length} \leq 10^4$
- $-10^4 \leq \text{nums}[i] \leq 10^4$
- nums is sorted in a strictly increasing order.

## Solution:

```
class Solution {  
  
    public TreeNode arrtobst(int []nums,int st,int ed){  
  
        if(st>ed){  
  
            return null;  
  
        }  
  
        int mid=(st+ed)/2;  
  
        TreeNode root=new TreeNode(nums[mid]);  
  
        root.left=arrtobst(nums,st,mid-1);  
  
        root.right=arrtobst(nums,mid+1,ed);  
  
        return root;  
  
    }  
  
    public TreeNode sortedArrayToBST(int[] nums) {  
  
        return arrtobst(nums,0,nums.length-1);  
  
    }  
}
```