

# Problem 108: Convert Sorted Array to Binary Search Tree

## Problem Information

**Difficulty:** Easy

**Acceptance Rate:** 74.83%

**Paid Only:** No

**Tags:** Array, Divide and Conquer, Tree, Binary Search Tree, Binary Tree

## Problem Description

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 <= nums.length <= 104` \* `-104 <= nums[i] <= 104` \* `nums` is sorted in a **strictly increasing** order.

## Code Snippets

### C++:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
 *     right(right) {}
 * };
 */
class Solution {
public:
    TreeNode* sortedArrayToBST(vector<int>& nums) {

    }
};
```

### Java:

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
 *
 class Solution {
    public TreeNode sortedArrayToBST(int[] nums) {
```

```
    }  
}
```

### Python3:

```
# Definition for a binary tree node.  
# class TreeNode:  
#     def __init__(self, val=0, left=None, right=None):  
#         self.val = val  
#         self.left = left  
#         self.right = right  
class Solution:  
  
    def sortedArrayToBST(self, nums: List[int]) -> Optional[TreeNode]:
```