

108. Convert Sorted Array to Binary Search Tree

108. Convert Sorted Array to Binary Search Tree Solved

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

Example 1:

```
graph TD
    0((0)) --- -3((-3))
    0 --- 9((9))
    -3 --- -10((-10))
    -3 --- 5((5))
    9 --- 9((9))
```

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

Accepted 31 / 31 testcases passed
Rahul kumar submitted at Feb 11, 2025 10:27

Runtime: 2 ms | Beats 70.18%
Memory: 22.76 MB | Beats 95.77%

Testcase: Case 1 Case 2

Code:

```
/**
 * 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* buildBST(vector<int>& nums, int left, int right) {
        if (left > right)
            return nullptr;

        int mid = left + (right - left) / 2;
        TreeNode* root = new TreeNode(nums[mid]);
```

```
root->left = buildBST(nums, left, mid - 1);  
root->right = buildBST(nums, mid + 1, right);
```

```
return root;  
}
```

```
TreeNode* sortedArrayToBST(vector<int>& nums) {  
    return buildBST(nums, 0, nums.size() - 1);  
}  
};
```