

783. Minimum Distance Between BST Nodes

[Description](#)[Hints](#)[Submissions](#)[Discuss](#)[Solution](#)

Given a Binary Search Tree (BST) with the root node `root`, return the minimum difference between the values of any two different nodes in the tree.

Example :

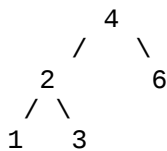
Input: `root = [4,2,6,1,3,null,null]`

Output: 1

Explanation:

Note that `root` is a `TreeNode` object, not an array.

The given tree `[4,2,6,1,3,null,null]` is represented by the following diagram:



while the minimum difference in this tree is 1, it occurs between node 1 and node 2, also between node 3 and node 2.

Note:

1. The size of the BST will be between 2 and 100.
 2. The BST is always valid, each node's value is an integer, and each node's value is different.
-

Seen this question in a real interview before?

- Difficulty: Easy
- Total Accepted: 3.8K
- Total Submissions: 8.2K
- Contributor: [awice](#)
-
- [Subscribe](#) to see which companies asked this question.

```
/**
```

```
 * Definition for a binary tree node.
```

```
 * struct TreeNode {
```

```

*      int val;
*      TreeNode *left;
*      TreeNode *right;
*      TreeNode(int x) : val(x), left(NULL), right(NULL) {}
* };
*/
class Solution {
public:
    int prev = INT_MIN;
    int ans = INT_MAX;
    int minDiffInBST(TreeNode* root) {
        check(root);
        return ans;
    }

    void check(TreeNode* node)
    {
        if(node==NULL) return;
        check(node->left);
        if(prev!=INT_MIN)
        {
            ans = min(ans,node->val-prev);
        }
        prev = node->val;
        check(node->right);
    }
};

```