

ABHISHEK KUMAR SINGH

Articles

Book Notes

About Me

Find Closest Value In BST



Write a function that takes in a Binary Search Tree (BST) and a target integer value and returns the closest value to that target value contained in the BST.

You can assume that there will only be one closest value.

Each BST node has an integer value, a left child node, and a right child node. A node is said to be a valid BST node if and only if it satisfies the BST property: its value is strictly greater than the values of every node to its left; its value is less than or equal to the values of every node to its right; and its children nodes are either valid BST nodes themselves or None / null.

Sample Input

```
JS
1  tree = 10
2      /   \
3     5     15
4    /  \  /  \
5   2   5 13 22
6  /       \
7 1         14
8  target = 12
```

Copy

Sample Output

JS

1 13

Copy

Hints

Hint 1

Try traversing the BST node by node, all the while keeping track of the node with the value closest to the target value. Calculating the absolute value of the difference between a node's value and the target value should allow you to check if that node is closer than the current closest one.

Hint 2

Make use of the BST property to determine what side of any given node has values close to the target value and is therefore worth exploring.

Hint 3

What are the advantages and disadvantages of solving this problem iteratively as opposed to recursively?

Optimal Space & Time Complexity

Average: $O(\log(n))$ time | $O(1)$ space - where n is the number of nodes in the BST
Worst: $O(n)$ time | $O(1)$ space - where n is the number of nodes in the BST

JS

```
1  function findClosestValueInBst(tree, target) {  
2      return findClosestValueHelper(tree, target, tree.value)  
3  }  
4  
5  function findClosestValueHelper(tree, target, closest) {  
6      if (tree === null) return closest;  
7  
8      if(Math.abs(target - closest) > Math.abs(target - tree.value)) {  
9          closest = tree.value  
10     }  
11  
12     if (target < tree.value) {  
13         return findClosestValueHelper(tree.left, target, closest)  
14     } else if (target > tree.value) {  
15         return findClosestValueHelper(tree.right, target, closest)  
16     } else {  
17         return closest;  
18     }
```

Copy

```
19 }  
20  
21 // This is the class of the input tree. Do not edit.  
22 class BST {  
23     constructor(value) {  
24         this.value = value;  
25         this.left = null;  
26         this.right = null;  
27     }  
28 }
```

[SHARE ARTICLE](#)[Twitter](#)[Facebook](#)[LinkedIn](#)[Mail to Friend](#)[← Depth-First Search](#)[Sorted Squared Array →](#)

Do you have any questions, or simply wish to contact me privately? Don't hesitate to shoot me a DM on [Twitter](#).

Have a wonderful day.

Abhishek 🙏

Subscribe To My Newsletter

Get email from me about my ideas, full-stack development resources, tricks and tips as well as exclusive previews of upcoming articles.

Sign me up!

No spam. Just the highest quality ideas you'll find on the web.

ABHISHEK KUMAR SINGH

Thanks for reading. It makes a difference. I'll try to help developers with tutorials & blogs. Blogs on design, development and happy-productive life. And it will also support and motivate me to share my knowledge with the community.

© 2022, ABHISHEK KUMAR SINGH 🙌

PAGES

[Articles](#)[Book Notes](#)[About](#)[SiteMap](#)[RSS](#)[Uses](#)[Now](#)[Favourite Links](#)[Algorithms](#)

SOCIAL

[Twitter](#)[Instagram](#)[LinkedIn](#)