

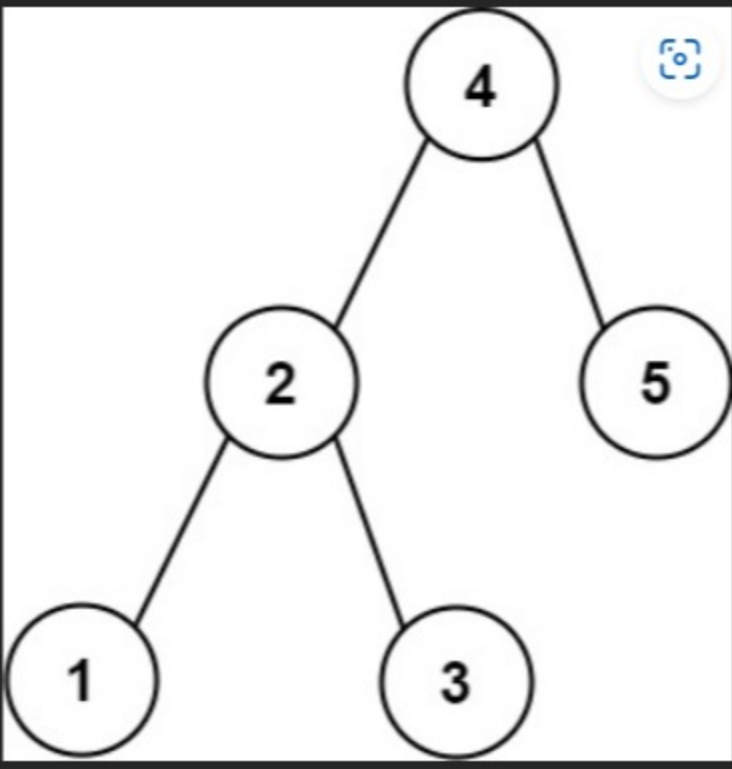
272. Closest Binary Search Tree Value II Premium

Hard Topics Companies Hint

Given the `root` of a binary search tree, a `target` value, and an integer `k`, return *the `k` values in the BST that are closest to the `target`*. You may return the answer in **any order**.

You are **guaranteed** to have only one unique set of `k` values in the BST that are closest to the `target`.

Example 1:



Input: `root = [4,2,5,1,3]`, `target = 3.714286`, `k = 2`
Output: `[4,3]`

Example 2:

Input: `root = [1]`, `target = 0.000000`, `k = 1`
Output: `[1]`

Constraints:

- The number of nodes in the tree is `n`.
- $1 \leq k \leq n \leq 10^4$.
- $0 \leq \text{Node.val} \leq 10^9$
- $-10^9 \leq \text{target} \leq 10^9$

Follow up: Assume that the BST is balanced. Could you solve it in less than $O(n)$ runtime (where `n = total nodes`)?

Seen this question in a real interview before? 1/5

Yes No

Accepted 123K | Submissions 206.3K | Acceptance Rate 59.6%

Topics

Two PointersStackTreeDepth-First SearchBinary Search TreeHeap (Priority Queue)Binary Tree

Companies

0 - 3 months
LinkedIn 2

0 - 6 months
Google 2

6 months ago
Amazon 2

Hint 1

Consider implement these two helper functions:
1. `getPredecessor(N)`, which returns the next smaller node to N.
2. `getSuccessor(N)`, which returns the next larger node to N.

Hint 2

Try to assume that each node has a parent pointer, it makes the problem much easier.

Hint 3

Without parent pointer we just need to keep track of the path from the root to the current node using a stack.

Hint 4

You would need two stacks to track the path in finding predecessor and successor node separately.

Similar Questions

Binary Tree Inorder TraversalEasy

Closest Binary Search Tree Value 🔒Easy

Closest Nodes Queries in a Binary Search TreeMedium

Discussion (8)