Median of BST

Find it in O(1) space & O(n) time complexity.

If I use a recursive function or auxilary array extra space will be involved. So we use morris inorder traversal method.

Approach:

1) Count the number of nodes(n) in BST using morris traversal
2) Perform morris traversal again to find the median
   - If n is odd, return middle element $(\frac{n+1}{2})$
   - If n is even, return average of $\frac{n}{2}$ & $\frac{n}{2}+1^{th}$ elements.

1) Count the total nodes

counting is done using morris traversal.

Steps:

1) Start from root node
2) If the left child does not exist, increment the count & move to the right child.
3) If the left child exists, find its inorder predecessor (rightmost child node in left subtree)
   - If the predecesor does not have a link (right pointer to root), create a temporary link to the current node & move left.
   - If the predecessor does have a link, remove it, increment count, move right.
4) Continue until entire tree is traversed.

2) Find median

If n is odd → median $= \left(\frac{n+1}{2}\right)^{th}$ element.

If n is even → Median $= \frac{\left(\frac{n}{2}\right)^{th} + \left(\frac{n}{2}+1\right)^{th}}{2}$

Morris traversal again

i) Start from root

ii) If there is no left child, increase currCount.

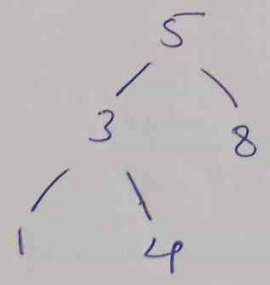
iii) If there is a left child, find inorder predecessor.

- If no link exists, create one & move left
- If link exists, remove it, increase currCount, & move right.

iv) Check for median condition

- If currCount == $\frac{n+1}{2}$ (for odd n) return value
- If currCount == $\frac{n}{2}$ or currCount == $\frac{n}{2}+1$ (for even) take average of the 2 values

e.g.

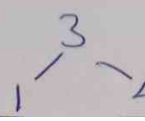


Morris inorder traversal

1) Start at root (5) (i.e, current = root)

- Left subtree exists → Find the inorder predecessor (right most node in left subtree)

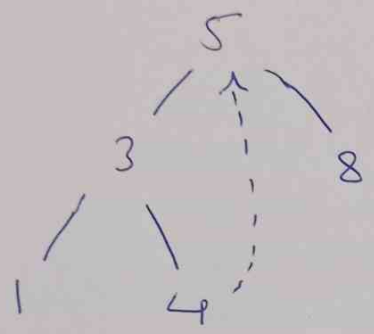
Left subtree is



2) The inorder predecessor of 5 is 4 (rightmost node in 3's subtree)

- Create temporary link by pointing 4's right to 5.

  prev.right = current.



3) Move to left child (3) (current = current.left)

At node (3)

- Left subtree exists → Find inorder predecessor (rightmost node in left subtree).

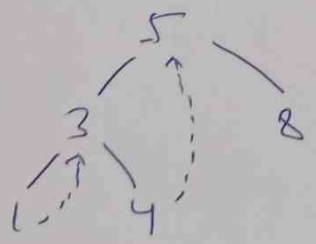
  The left subtree of 3:

  1

  Inorder predecessor of 3 is 1. (prev = 1)

  Create temporary link by pointing 1's right to 3

  prev.right = current.



4) Move to left child (1) (current = current.left) again

At node (1)

- No left subtree → Count = 1

- move to the right (link exists because we created it in previous step) → Go back to 3 (remove thread/link)

- Back to node(3)

  → Count=2

  move to right child (4)

- At Node(4)

  → No left subtree → Count=3

  Move to the right (link is there pointing to 5) → Go back to 5 (remove link).

- Back to Node(5)

  → Count = 4

  Move to the right child 8

- At Node(8)

  No left subtree → count=5.

  Traversal ends

  All nodes counted

This is how morris traversal works.

Do this traversal twice, one for counting nodes, another for finding median.

Then I will get median in BST.