Notebook

August 2, 2024

[ ]:
```python
# https://leetcode.com/problems/convert-sorted-list-to-binary-search-tree/

class Solution:
    # Simple nums sorted array to BST

    def sortedArrayToBST(self, nums: List[int]) -> Optional[TreeNode]:
        if not nums: return None
        mid = len(nums) // 2
        root = TreeNode(nums[mid])
        root.left = self.sortedArrayToBST(nums[:mid])
        root.right = self.sortedArrayToBST(nums[mid+1:])
        return root

    # LinkedList Sorted List to BST
    def sortedListToBST(self, head: Optional[ListNode]) -> Optional[TreeNode]:

        def findMid(head):
            if head is None or head.next is None: return head

            slow, fast, prev = head, head, None
            while fast is not None and fast.next is not None:
                prev = slow
                slow = slow.next
                fast = fast.next.next

            if prev is not None: prev.next = None

            return slow

        if head is None: return None

        mid = findMid(head)
        if mid is None: return None

        root = TreeNode(mid.val)
        if head == mid: return root
```

```
        root.left = self.sortedListToBST(head)
        root.right = self.sortedListToBST(mid.next)
        return root
```