Sorted DLL to BST/BT

```python
import sys


class Node:
    def __init__(self, val):
        self.data = val
        self.prev = None
        self.next = None




def convertDLLToTree(head):
    if head is None or head.next is None:
        return head
    mid = findMidOfDLL(head)
    last = mid.prev
    forward = mid.next
    mid.prev = None
    mid.next = None
    forward.prev = None
    if last!=None:
        last.next = None
    leftNode = head if last is not None else None
    rightNode = forward
    root = mid
    root.prev = convertDLLToTree(leftNode)
    root.next = convertDLLToTree(rightNode)
    return root


def findMidOfDLL(head):
    if head is None or head.next is None:
        return head
    slow = head
    fast = head
    while fast.next is not None and fast.next.next is not None:
        slow = slow.next
        fast = fast.next.next
    return slow



def addFirst(node, head):
    node.next = head
    head.prev = node
    head = node
    return head


head = Node(7)
head = addFirst(Node(6), head)
head = addFirst(Node(5), head)
head = addFirst(Node(4), head)
head = addFirst(Node(3), head)
head = addFirst(Node(2), head)
head = addFirst(Node(1), head)


root = convertDLLToTree(head)

def inorder(root):
    if root is None:
        return
    inorder(root.prev)
    print(root.data, end=' ')
    inorder(root.next)

inorder(root)
```