AlgoExpert Quad Layout Python 12px Sublime Monok

Prompt Scratchpad Our Solution(s) Video Explanation Run Code

Solution 1

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n) time | O(1) space - where n is the number of nodes in the Linked List
def rearrangeLinkedList(head, k):
    smallerListHead = None
    smallerListTail = None
    equalListHead = None
    equalListTail = None
    greaterListHead = None
    greaterListTail = None

    node = head
    while node is not None:
        if node.value < k:
            smallerListHead, smallerListTail = growLinkedList(smallerListHead, smallerListTail, node)
        elif node.value > k:
            greaterListHead, greaterListTail = growLinkedList(greaterListHead, greaterListTail, node)
        else:
            equalListHead, equalListTail = growLinkedList(equalListHead, equalListTail, node)

        prevNode = node
        node = node.next
        prevNode.next = None

    firstHead, firstTail = connectLinkedLists(smallerListHead, smallerListTail, equalListHead, equalListTail)
    finalHead, _ = connectLinkedLists(firstHead, firstTail, greaterListHead, greaterListTail)
    return finalHead


def growLinkedList(head, tail, node):
    newHead = head
    newTail = node

    if newHead is None:
        newHead = node
    if tail is not None:
        tail.next = node

    return (newHead, newTail)


def connectLinkedLists(headOne, tailOne, headTwo, tailTwo):
    newHead = headTwo if headOne is None else headOne
    newTail = tailOne if tailTwo is None else tailTwo

    if tailOne is not None:
        tailOne.next = headTwo

    return (newHead, newTail)


# This is the class of the input linked list.
class LinkedList:
    def __init__(self, value):
        self.value = value
        self.next = None
```