42. Merge k Sorted Lists

You are given an array of k linked-lists lists, each linked-list is sorted in ascending order.

Merge all the linked-lists into one sorted linked-list and return it.

Code:

```
from heapq import heappush, heappop
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next
def mergeKLists(lists):
    min_heap = []
    dummy = ListNode()
    current = dummy
    for i, l in enumerate(lists):
        if l:
            heappush(min_heap, (l.val, i, l))
    while min_heap:
        val, i, node = heappop(min_heap)
        current.next = ListNode(val)
        current = current.next
        node = node.next
        if node:
            heappush(min_heap, (node.val, i, node))
    return dummy.next
def printList(node):
    while node:
        print(node.val, end=" -> ")
        node = node.next
    print("None")
def createList(arr):
    if not arr:
        return None
    head = ListNode(arr[0])
    current = head
    for value in arr[1:]:
```

Output:

```
Merged Linked List:
1 -> 1 -> 2 -> 3 -> 4 -> 4 -> 5 -> 6 -> None
```

Time Complexity:

- T(n)= O(n)