

k개의 정렬된 리스트를 1개의 정렬된 리스트로 병합하라.

Input: lists = [[1,4,5],[1,3,4],[2,6]]

Output: [1,1,2,3,4,4,5,6]

Explanation: The linked-lists are:

```
[
  1->4->5,
  1->3->4,
  2->6
]
```

merging them into one sorted list:

1->1->2->3->4->4->5->6

# Definition for singly-linked list.

# class ListNode:

# def \_\_init\_\_(self, val=0, next=None):

# self.val = val

# self.next = next

## 1. 브루트 포스

class Solution:

def mergeKLists(self, lists: List[ListNode]) -> ListNode:

\_list = []

for i in lists:

while i:

\_list.append(i.val)

i = i.next

\_list.sort()

\_list.reverse()

result = None

for i in \_list:

result = ListNode(i, result)

return result

## 2. heapq 활용

class Solution:

def mergeKLists(self, lists: List[ListNode]) -> ListNode:

root = result = ListNode(None)

heap = []

for i in range(len(lists)):

if lists[i]:

heapq.heappush(heap, (lists[i].val, i, lists[i]))

while heap:

node = heapq.heappop(heap)

idx = node[1]

result.next = node[2]

result = result.next

if result.next:

heapq.heappush(heap, (result.next.val, idx, result.next))

return root.next

