# US # MEDIUM + # META # AMAZON

## 23. Merge k Sorted Lists

Hard 18.6K 670

Companies

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.*

**Example 1:**

```
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
```

**Example 2:**

```
Input: lists = []
Output: []
```

**Example 3:**

```
Input: lists = [[]]
Output: []
```

```java
public class MergeKSortedList {
    // O(n log n) O(n)
    public ListNode mergeKLists(ListNode[] lists) {
        PriorityQueue<ListNode> heap = new PriorityQueue<>((a, b) -> a.val - b.val);
        for(ListNode node : lists){
            if(node != null)
                heap.add(node);
        }
        ListNode head = new ListNode(0);
        ListNode current = head;
        while(!heap.isEmpty()){
            ListNode node = heap.poll();
            current.next = node;
            current = current.next;
            if (node.next != null)
                heap.add(node.next);
        }
        return head.next;
    }
    public static void printList(ListNode node) {
        while (node != null) {
            System.out.print(node.val + " ");
            node = node.next;
        }
        System.out.println();
    }

    public static void main(String[] args) {
        MergeKSortedList merger = new MergeKSortedList();
        ListNode[] lists = new ListNode[3]; // Example initialization
        lists[0] = new ListNode(1, new ListNode(4, new ListNode(5)));
        lists[1] = new ListNode(1, new ListNode(3, new ListNode(4)));
        lists[2] = new ListNode(2, new ListNode(6));
        ListNode mergedList = merger.mergeKLists(lists);
        printList(mergedList);
    }
}
```