**[Merge k Sorted Lists](https://leetcode.com/problems/merge-k-sorted-lists/)**

**class Solution {**

**public ListNode mergeKLists(ListNode[] lists) {**

**if(lists == null || lists.length == 0) {**

**return null;**

**}**

**PriorityQueue<ListNode> pq = new PriorityQueue<>(new Comparator<ListNode>(){**

**public int compare(ListNode a , ListNode b) {**

**return a.val - b.val;**

**}**

**});**

**ListNode dummy = new ListNode(0);**

**ListNode current = dummy;**

**for(ListNode list : lists) {**

**if(list != null) {**

**pq.offer(list);**

**}**

**}**

**while(!pq.isEmpty()) {**

**ListNode temp = pq.poll();**

**current.next = temp;**

**current = current.next;**

**if(temp.next == null) {**

**continue;**

**}**

**pq.offer(temp.next);**

**}**

**return dummy.next;**

**}**

**}**

Time Complexity : O(n \* (logk)) where n is total no of elements and k is given no of lists

Space Complexity : O(k) where k is given no of lists