Merge k Sorted Lists

Question: Merge k sorted linked lists and return it as one sorted list. Analyse and describe its complexity.

Solutions:

```
class ListNode:
    def __init__(self, x):
        self.val = x
        self.next = None


class Solution:
    # @param a list of ListNode
    # @return a ListNode
    def mergeKLists(self, lists):
        if len(lists)==0:
            return None
        while len(lists)>1:
            nextLists = []
            for i in range(0,len(lists)-1,2):
                nextLists.append(self.mergeLists(lists[i],lists[i+1]))
            if len(lists)%2==1:
                nextLists.append(lists[len(lists)-1])
            lists = nextLists
        return lists[0]


    def mergeLists(self, list1, list2):
        dummy = ListNode(0)
        list = dummy
```

```
        while list1 != None and list2 != None:
            if list1.val < list2.val:
                list.next = list1
                list1 = list1.next
            else:
                list.next = list2
                list2 = list2.next
            list = list.next
        if list1 == None:
            list.next = list2
        else:
            list.next = list1
        return dummy.next

    def printll(self, node):
        while node:
            print ( node.val )
            node = node.next

if __name__ == '__main__':
    ll1, ll1.next, ll1.next.next = ListNode(2), ListNode(3), ListNode(5)
    ll2, ll2.next, ll2.next.next = ListNode(4), ListNode(7), ListNode(15)
    ll3, ll3.next, ll3.next.next = ListNode(6), ListNode(9), ListNode(10)
    Solution().printll( Solution().mergeKLists([ll1,ll2,ll3]) )
```