Medium Codewriting 3000

Note: Your solution should have `O(l1.length + l2.length)` time complexity, since this is what you will be asked to accomplish in an interview.

Given two singly linked lists sorted in non-decreasing order, your task is to merge them. In other words, return a singly linked list, also sorted in non-decreasing order, that contains the elements from both original lists.

Example

- For `l1 = [1, 2, 3]` and `l2 = [4, 5, 6]`, the output should be `mergeTwoLinkedLists(l1, l2) = [1, 2, 3, 4, 5, 6]`;
- For `l1 = [1, 1, 2, 4]` and `l2 = [0, 3, 5]`, the output should be `mergeTwoLinkedLists(l1, l2) = [0, 1, 1, 2, 3, 4, 5]`.

Input/Output

- **[execution time limit] 20 seconds (scala)**
- **[input] linkedlist.integer l1**

  A singly linked list of integers.

  Guaranteed constraints:

  $0 \le$ list size $\le 10^4$,

  $-10^9 \le$ element value $\le 10^9$.

- **[input] linkedlist.integer l2**

  A singly linked list of integers.

  Guaranteed constraints:

  $0 \le$ list size $\le 10^4$,

  $-10^9 \le$ element value $\le 10^9$.

- **[output] linkedlist.integer**

  A list that contains elements from both `l1` and `l2`, sorted in non-decreasing order.

[Scala] Syntax Tips

```
def helloWorld(name: String): String = {
    println("This prints to the console when you Run Tests")
    "Hello, " + name
}
```