Merge Two Sorted Lists

Solving the merge two sorted lists problem — a fundamental linked list question that can be tackled either iteratively or recursively.

Recommended Strategy:
Iterative with Dummy Node:

1. Use a dummy node to simplify handling the head of the new list.
2. Use a tail pointer to keep track of the last node in the merged list.
3. Loop as long as both lists are non-empty:
   - Append the node with the smaller value to tail → next.
   - Advance the list pointer (list1 or list2) and tail.
4. After the loop, append the remaining part of the non-empty list (if any).
5. Return dummy.next.

Time Complexity: $O(m+n)$ where m and n are the lengths of the two lists.

Edge Cases:
   - One or both lists are empty.
   - Lists have duplicate values.
   - All elements of one list are smaller than the other.

Recursive Alternative (optional idea):

```
if (!list1) return list2;
if (!list2) return list1;
if (list1→val < list2→val) {
    list1→next = mergeTwoLists (list1→next, list2);
    return list1;
} else {
    list2→next = mergeTwoLists (list1, list2→next);
    return list2;
}
```