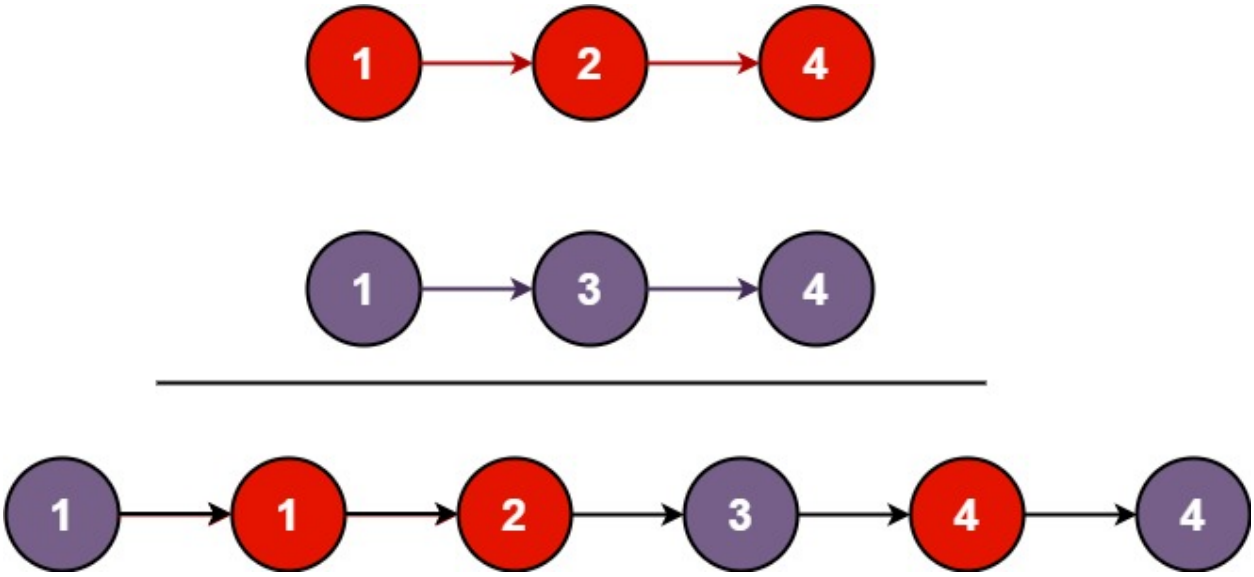
Merge two sorted lists

You are given the heads of two sorted linked lists `list1` and `list2`.

Merge the two lists into one **sorted** list. The list should be made by splicing together the nodes of the first two lists.

Return *the head of the merged linked list*.

Example 1:




```
Input: list1 = [1,2,4], list2 = [1,3,4]
Output: [1,1,2,3,4,4]
```

Example 2:

```
Input: list1 = [], list2 = []
Output: []
```

Example 3:

```
Input: list1 = [], list2 = [0]
Output: [0]
```

The best solution i found uses the approach is to find the smallest elements from both lists and add them together so lets initalise a dummy node that represents the start of the list i.e. `dummyNode = ListNode(-1)` we also need to make two variables to represent the current process in the list

```
dummyNode = ListNode(-1)
        temp = dummyNode

        curr1, curr2 = list1, list2
```

Next we need to initialize a loop to say whilst curr1 and curr2 is not empty and our first condition we will check if the value of curr1 < than curr2 temp is then moved to the next node `temp.next = curr1` after adding curr1 to the list the pointer is moved to the next node in the first list.

```
while curr1 and curr2:
            if curr1.val < curr2.val:
                temp.next = curr1
                curr1 = curr1.next
```

Now we will just do the same check for curr2 this time the else statement will just represent if curr2 is less than or equal to curr1 in this case it is stored in the temp

variable the curr value is moved to the next at the end of both conditional blocks the temp variable is moved to next node.

```
else:
            temp.next = curr2
            curr2 = curr2.next
        temp = temp.next
```

Next we need to make sure that all nodes are being added to the dummylist we have

```
# Attach the remaining nodes
    temp.next = curr1 if curr1 else curr2
```

Complete Solution

```
class ListNode:
  def __init__(self, val=0, next=None):
    self.val = val
    self.next = next

class Solution:
  def mergeTwoLists(self, list1: ListNode, list2: ListNode) -> ListNode:
    dummyNode = ListNode(-1)
    temp = dummyNode

    curr1, curr2 = list1, list2
```

```
        while curr1 and curr2:
            if curr1.val < curr2.val:
                temp.next = curr1
                curr1 = curr1.next
            else:
                temp.next = curr2
                curr2 = curr2.next
            temp = temp.next

        # Attach the remaining nodes
        temp.next = curr1 if curr1 else curr2

        return dummyNode.next

if __name__ == '__main__':
    s = Solution()

    # Create first linked list: 1 → 2 → 4
    list1 = ListNode(1)
    list1.next = ListNode(2)
    list1.next.next = ListNode(4)

    # Create second linked list: 1 → 3 → 4
    list2 = ListNode(1)
    list2.next = ListNode(3)
    list2.next.next = ListNode(4)

    merged_list = s.mergeTwoLists(list1, list2)

    # Print merged linked list
    while merged_list:
        print(merged_list.val, end=" → ")
        merged_list = merged_list.next
```