

# <재귀구조>

```
def mergeTwoLists(self, l1: ListNode, l2: ListNode) -> ListNode:
    if (not l1) or (l2 and l1.val > l2.val):
        l1, l2 = l2, l1
    if l1:
        l1.next = self.mergeTwoLists(l1.next, l2)
    return l1
```

list1 => [1] -> [2] -> [4] -> [ ]  
list2 => [3] -> [4] -> [ ]  
풀이가 명확하고 코드도 길지 않다. 크거나 이 짧은 코드에 너무 많은 내용이 함축되어 있어서 이해하기가 쉽지 않을 뿐더러, 재귀가 포함되어 있어 더욱 어렵다.

하나씩 풀어서 살펴보자. 먼저 l1과 l2의 값을 비교해 작은 값이 왼쪽에 오게 하고, next (코드부분)은 그대로 값이 뒤이도록 재귀 호출하는 게 이 전체 코드의 전부다. 여기서 첫 번째 if 문인 괄호로 묶인 부분과 변수를 스왑(Swap)하는 부분부터 살펴보자.

1) next = M.T (2, 4 / 3, 4)  
1 2 3 4 4  
switch

2) next = M.T (3, 4 / 2, 4)  
1 2 3 4 4  
switch

3) next = M.T (4 / 3, 4)  
2 3 4 4  
switch

4) next = M.T (4 / 4)  
3 4 4  
switch

$$\textcircled{8}.next = m.T(\textcircled{None} / \textcircled{4})$$

$$\begin{matrix} 4 & 4 \end{matrix}$$

Diagram showing a switch operation between  $\textcircled{None}$  and  $\textcircled{4}$ . A bracket labeled "switch" connects the two nodes. Red numbers 1 and 2 are above the nodes, and a red arrow points from the first node to the second.

$$\textcircled{8}.next = m.T(\textcircled{None} / \textcircled{None})$$

$$\begin{matrix} 4 \end{matrix}$$

A red arrow points from the  $\textcircled{None}$  node in the previous equation to the  $\textcircled{None}$  node in this equation.

$$8) = None \text{ 리턴시작}$$

$$None \text{ 이 되는 순간 반환시작}$$