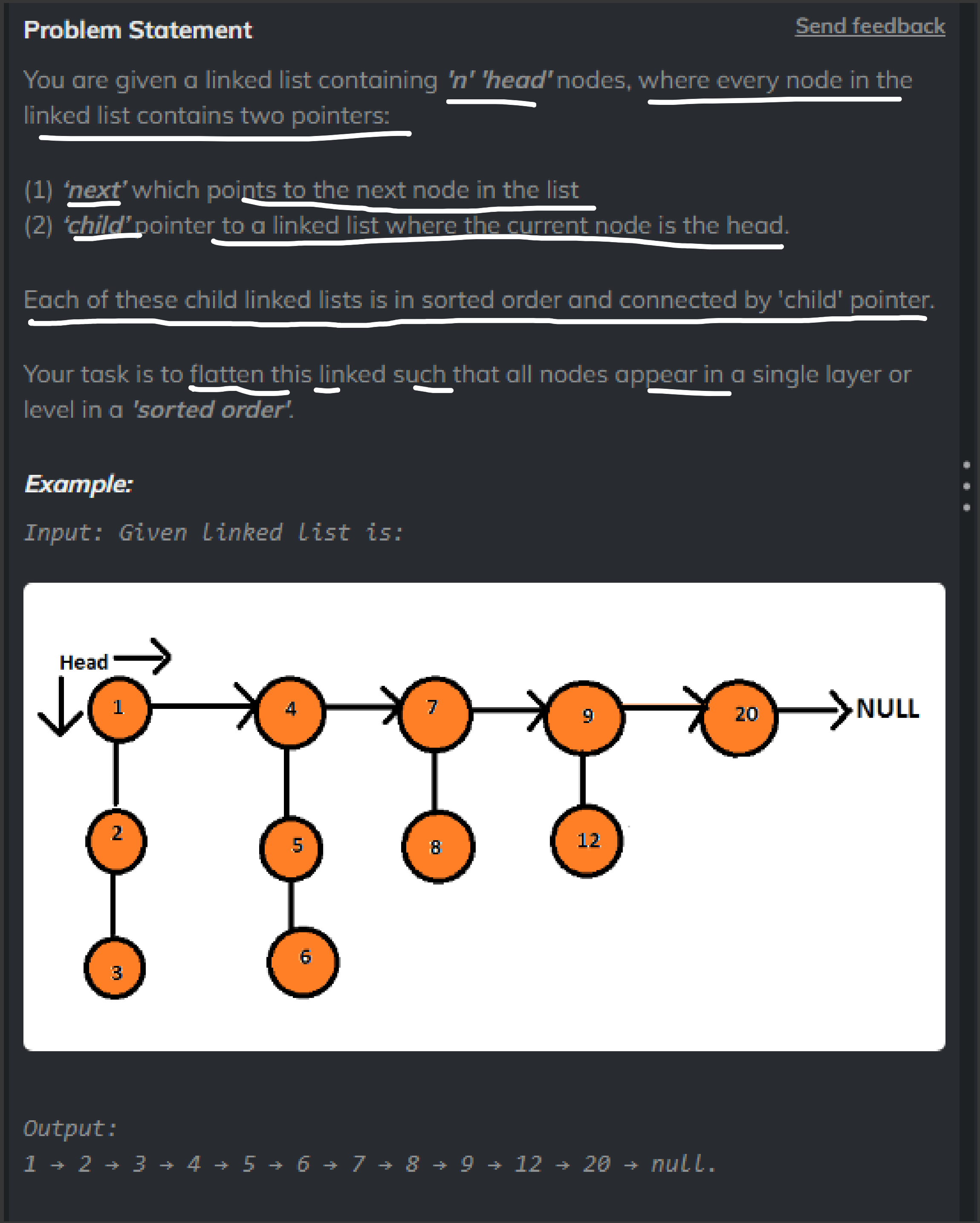
Flatten the Linked List



Problem Statement Send feedback

You are given a linked list containing *'n' 'head'* nodes, where every node in the linked list contains two pointers:

(1) *'next'* which points to the next node in the list
(2) *'child'* pointer to a linked list where the current node is the head.

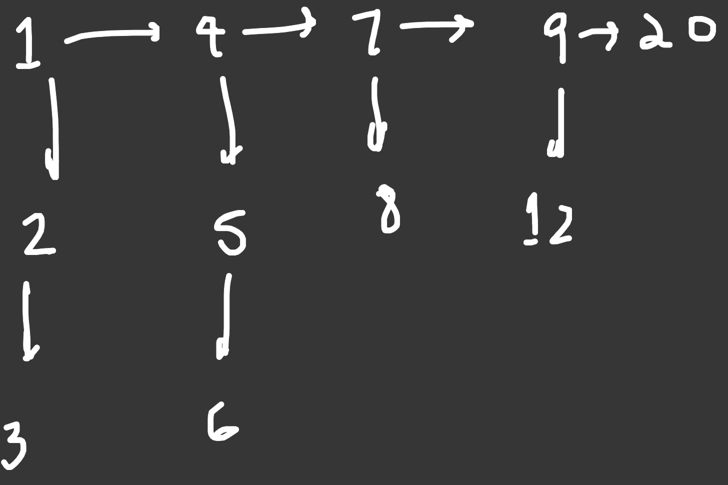
Each of these child linked lists is in sorted order and connected by 'child' pointer.

Your task is to flatten this linked such that all nodes appear in a single layer or level in a *'sorted order'*.

Example:

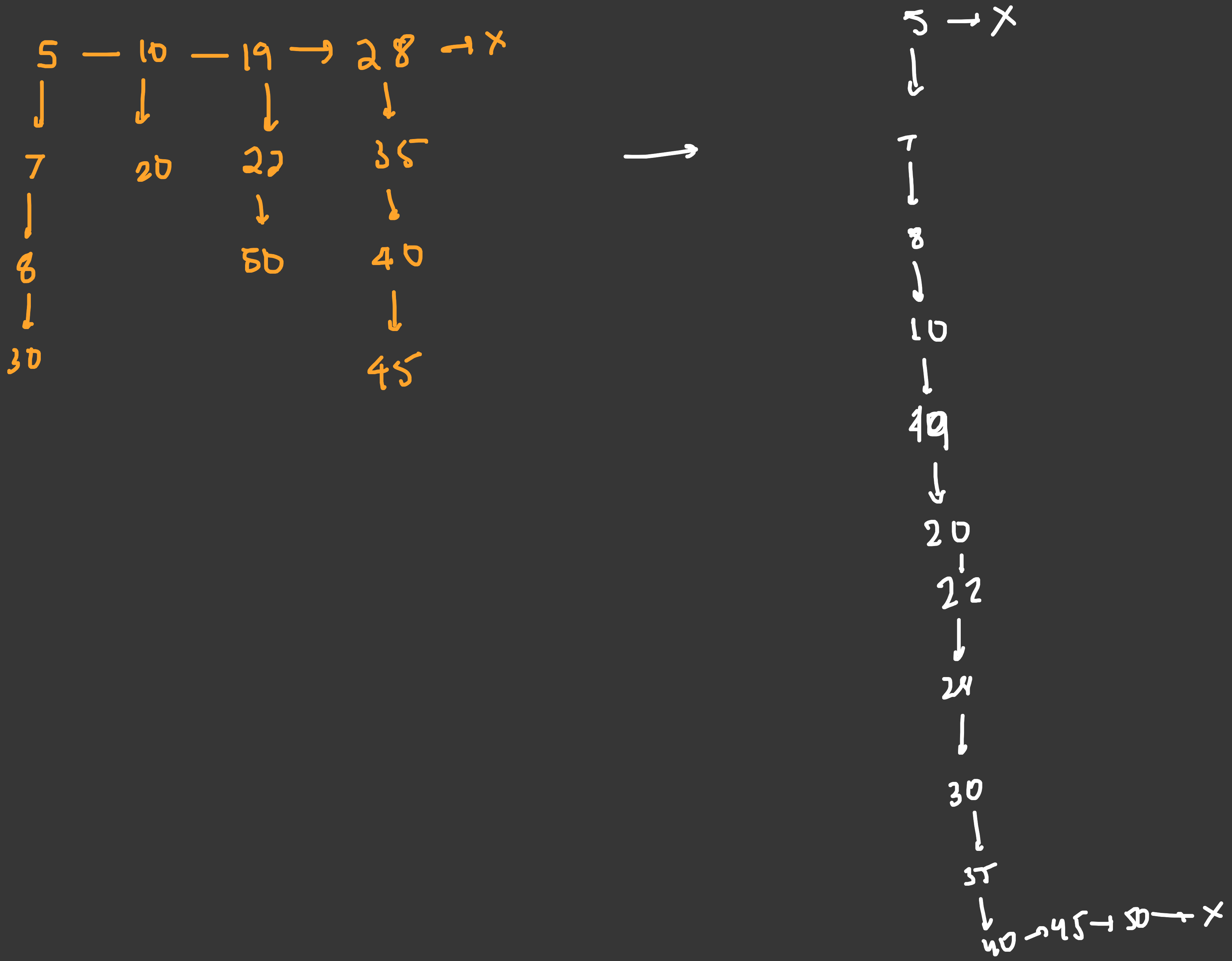
```
Input: Given linked list is:
```

```
Output:
1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 12 → 20 → null.
```



Output

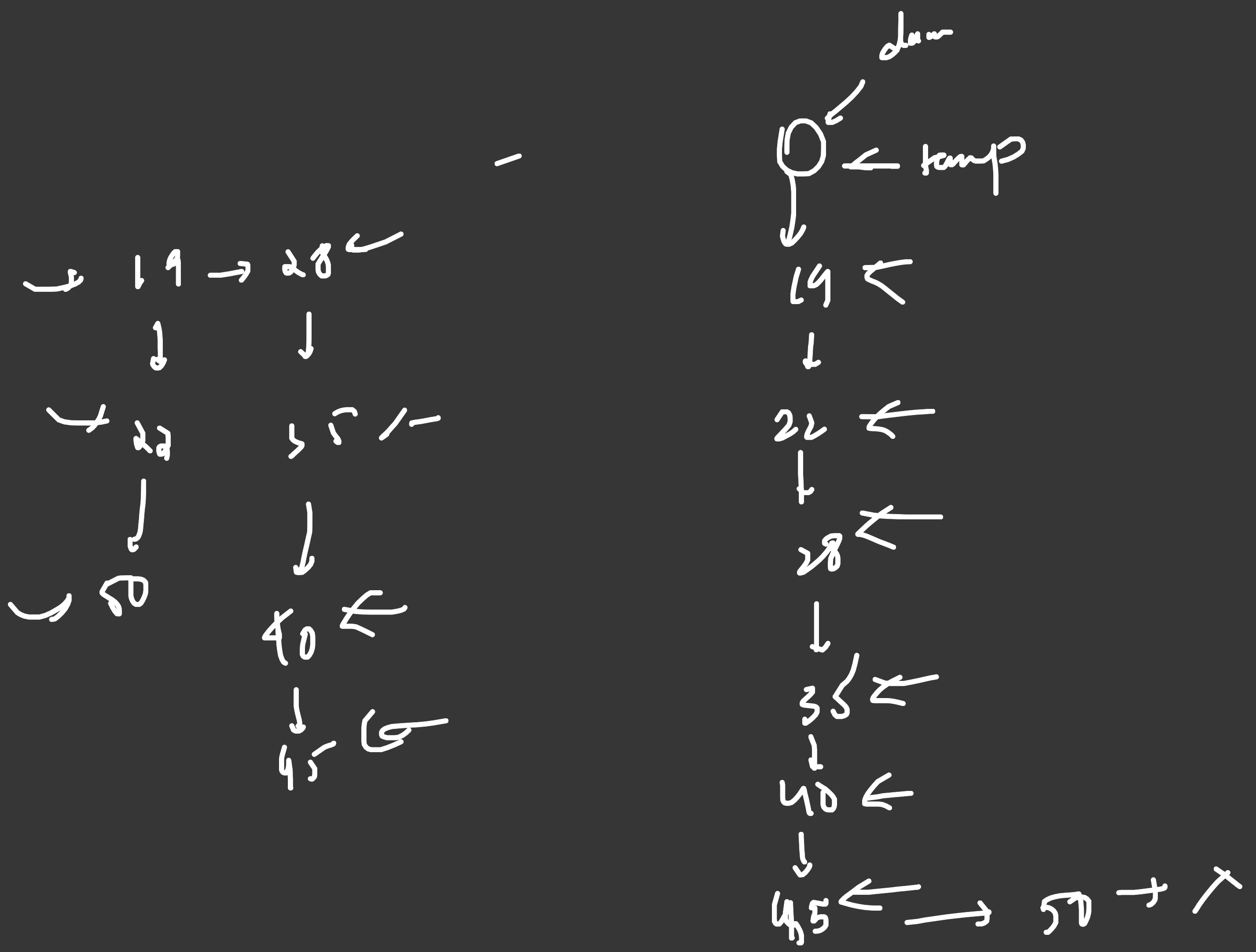
1 → 2 → 3 → 4 → 5 → 6 → 7 → 8
↓
9
NULL ← 20 ← 12 ← 9


5
10
19
28
X
7
20
22
35
8
50
40
30
45
5
X
7
8
10
19
20
22
28
30
35
40
45
50
X

Approach:

Idea is to use merge Sort Algorithm as linked list bottom is in sorted.

So, we merge the linked list from last linked list. For that we use recursion



```
if(head== NULL || head→ next == NULL
        return head;

head → next = flatten( heat → next);

head = merge( head, head→ next);

return head.
```

Time Complexity

$O(N)$

$N \rightarrow$ summation of all Nodes in LinkedList

$S.C. = O(1)$