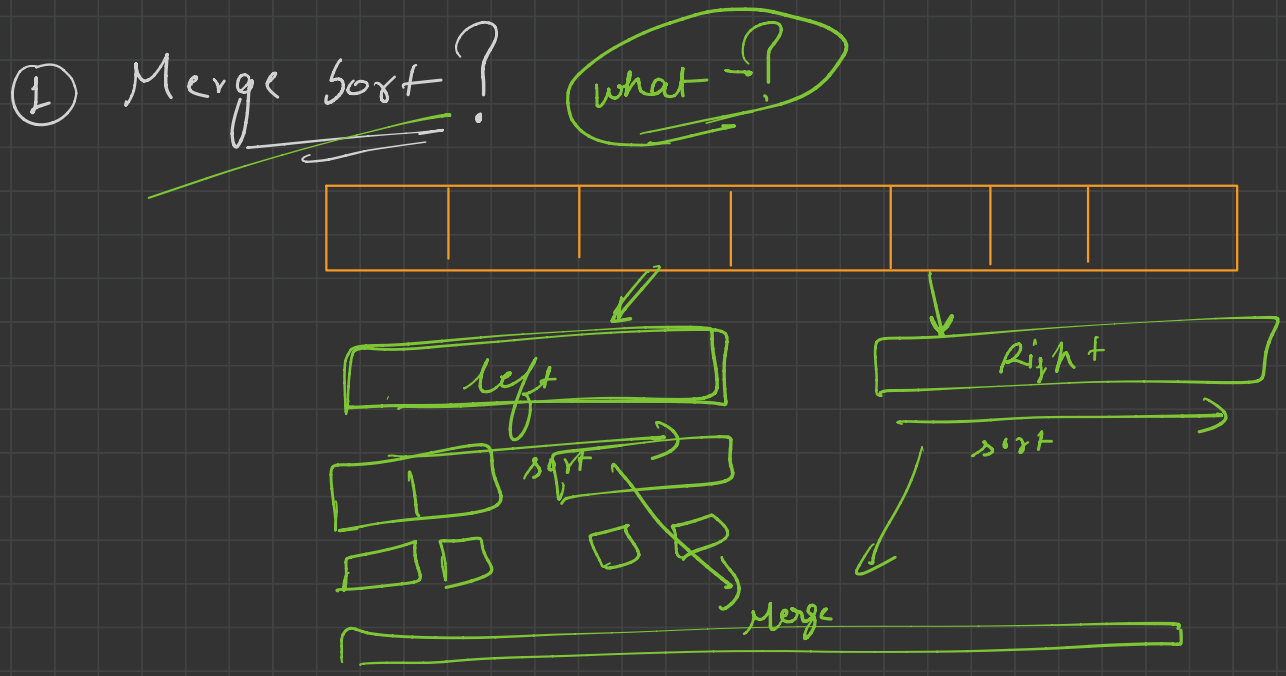
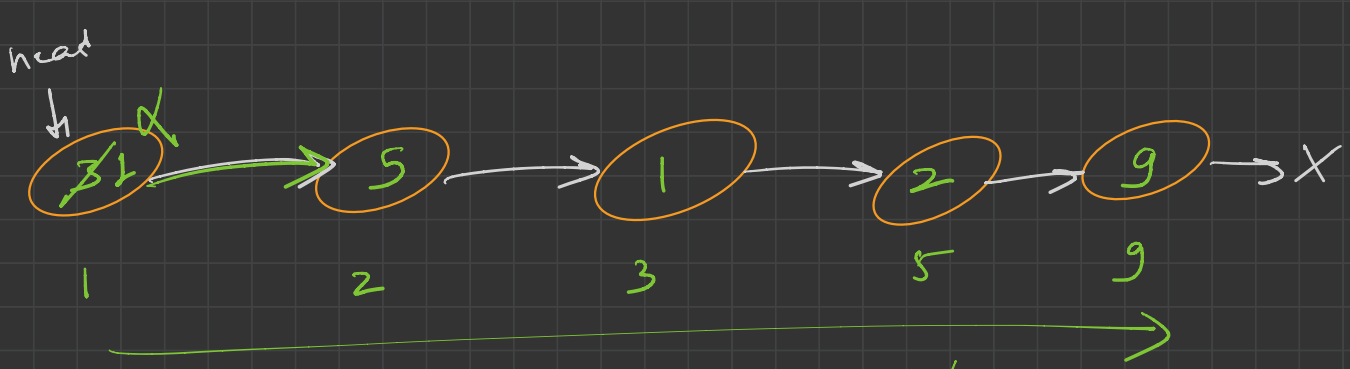
Merge Sort

Linked List

① Merge sort ?



i/p



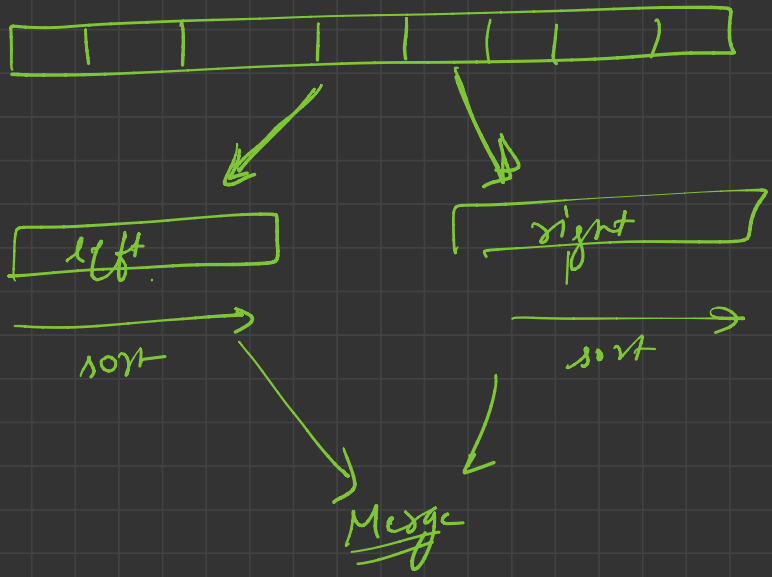
o/p → sorted

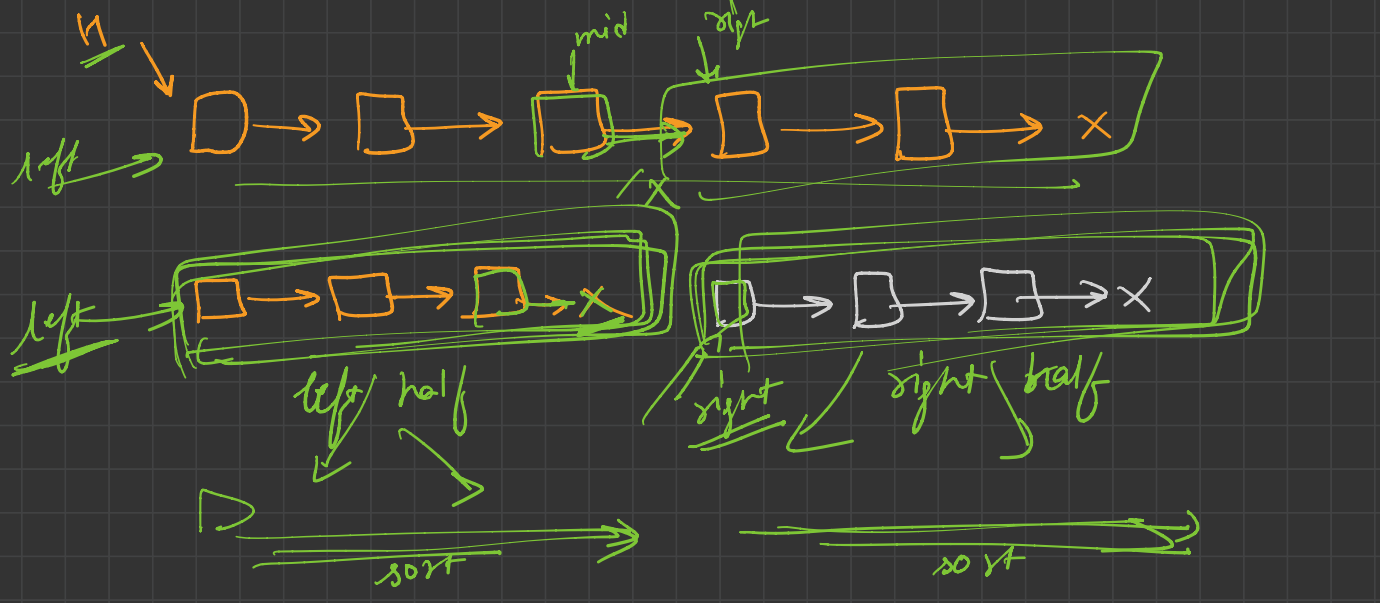
data replacement → ✗

change links

How - ?

Approach:-



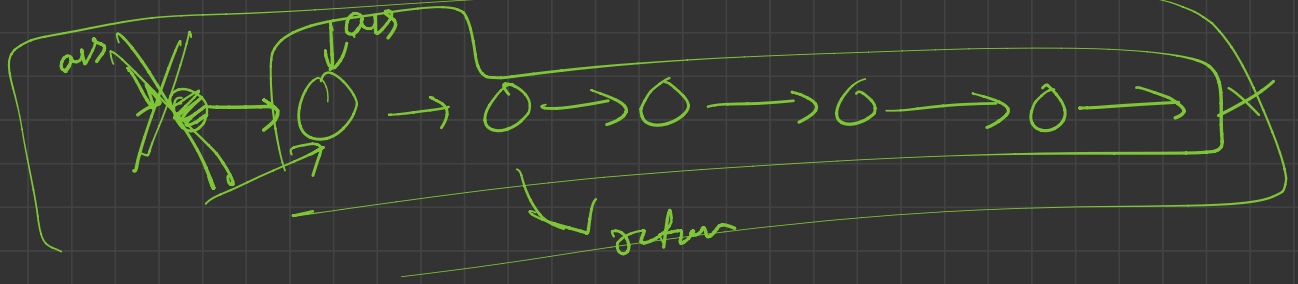


T.C → O(n log n)

Merge

(Merge 2 sorted LL);

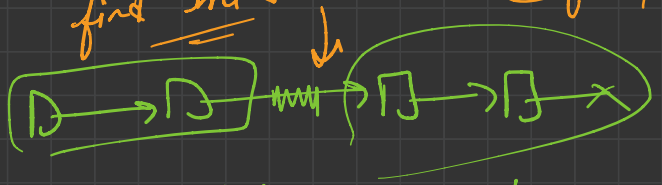
M.S (LL)



M·S → (L2)

I → find mid ——— (fast/slow approach)

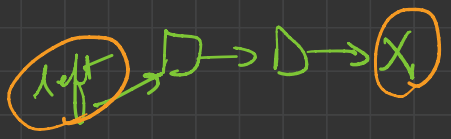
II →

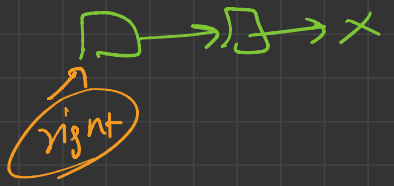


break into 2 halves

T·C → O(n log n)

S·C → O(log n)





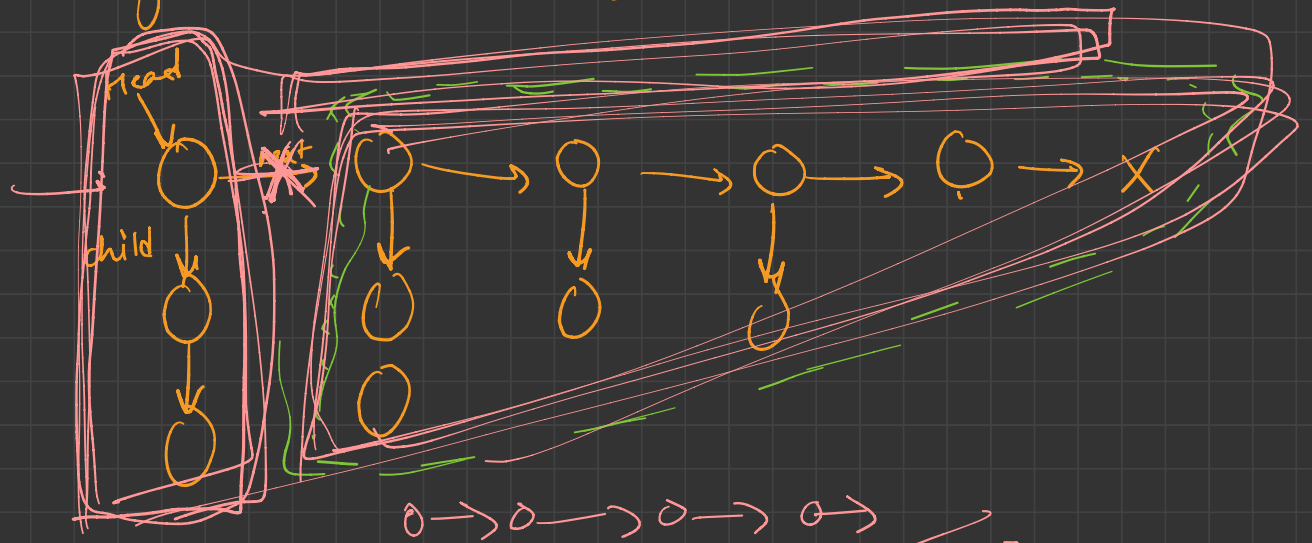
(iii) Recursively sort left/right halves

(iv) merge left/right sorted halves → (Merge 2 sorted LL)

(v) return merged List

(iii)/(v)

(11) flatten a Linked List



head

sorted

Flatten (head)

{

Node * down = head ;

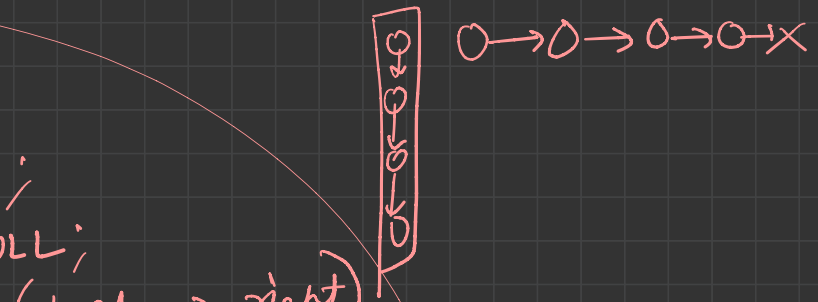
down → next = NULL ;

Node * right = flatten (head → right)

Node * ans = merge (down, right)

return ans ;

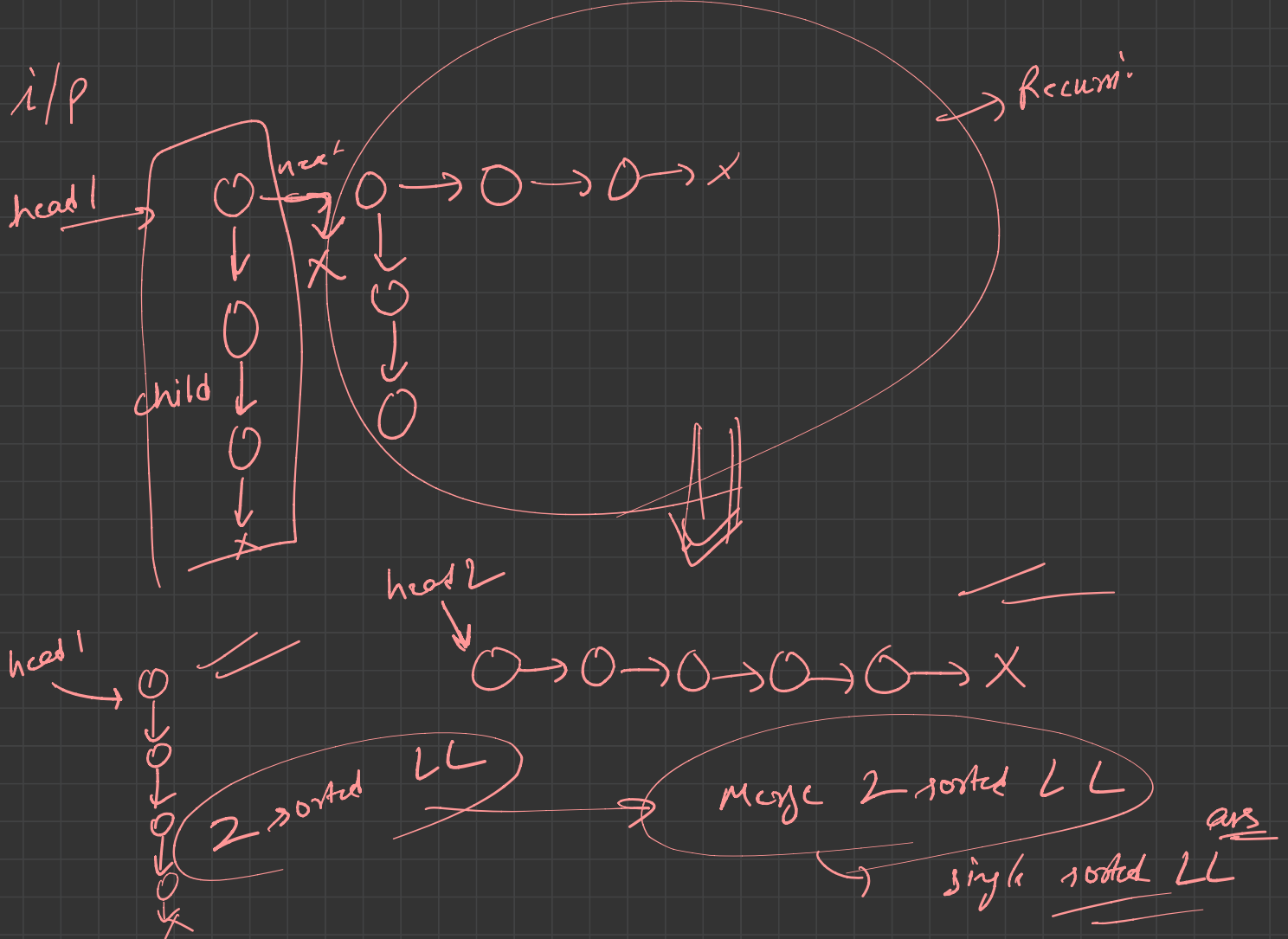
}



Merge 2 sorted Linked list

Exact Code → X

H/W


i/p
head1
next
child
Recursi
head2
head1
2 sorted LL
Merge 2 sorted LL
single sorted LL
ans