

---

---

---

---

---



## Linked List Questions

# → Remove Duplicates

Remove duplicates from a sorted LL  
→ a unsorted LL

i/p

→



o/p →



Approach:-

i/p

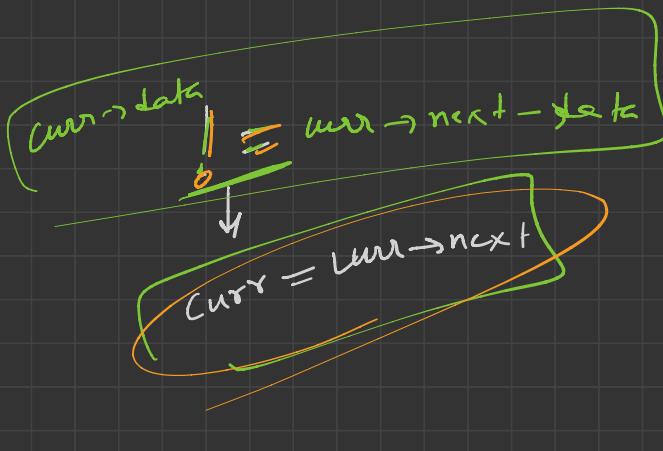
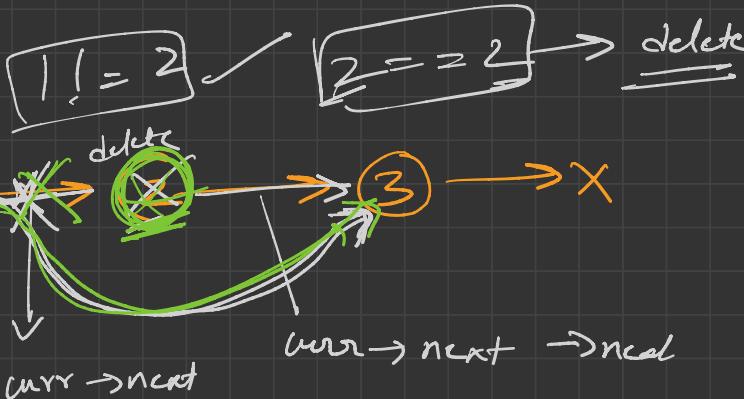
head



1

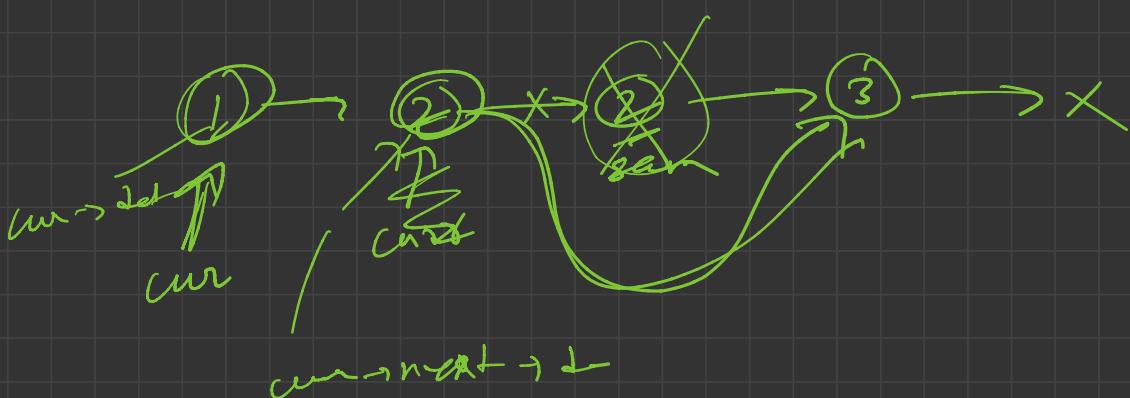
X

curr



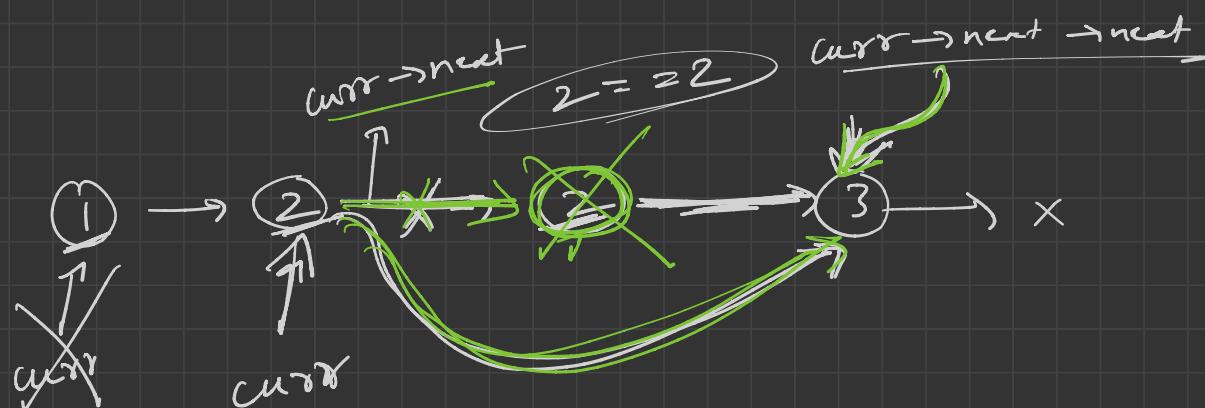
=  
① point to next->next + curr  
curr->next = curr->next->next

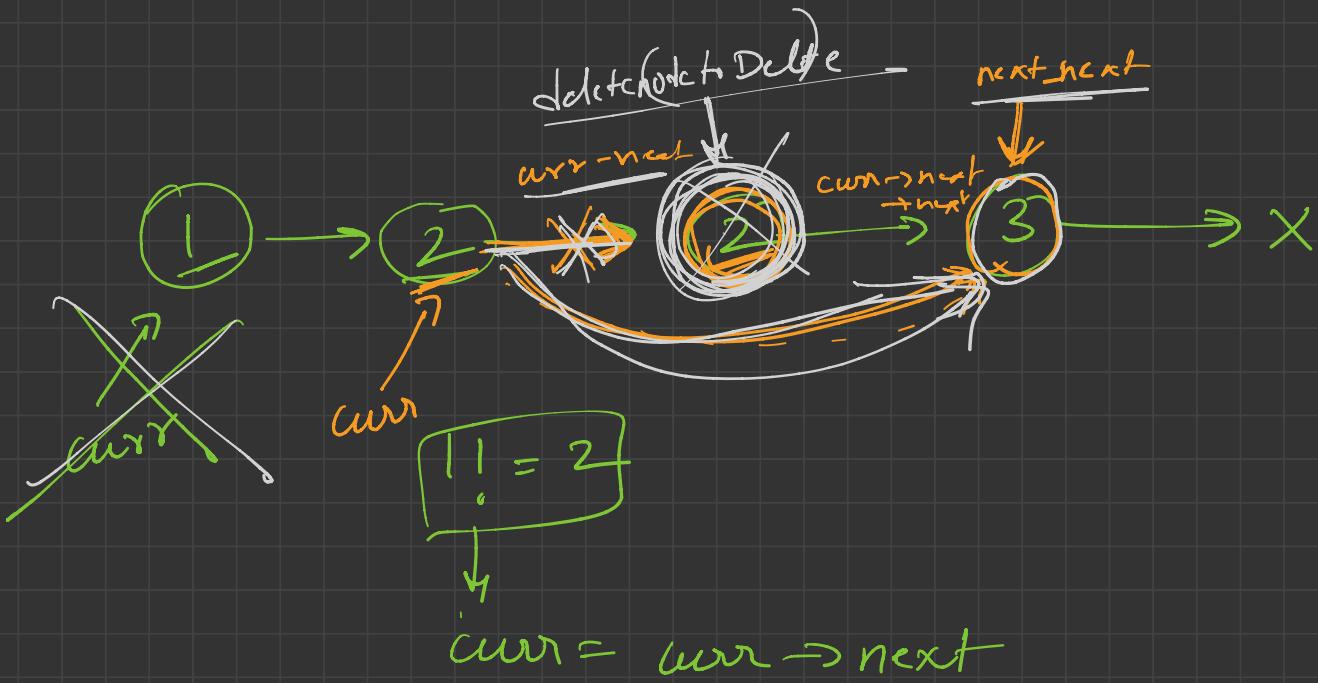
② delete Kawada duplicate work



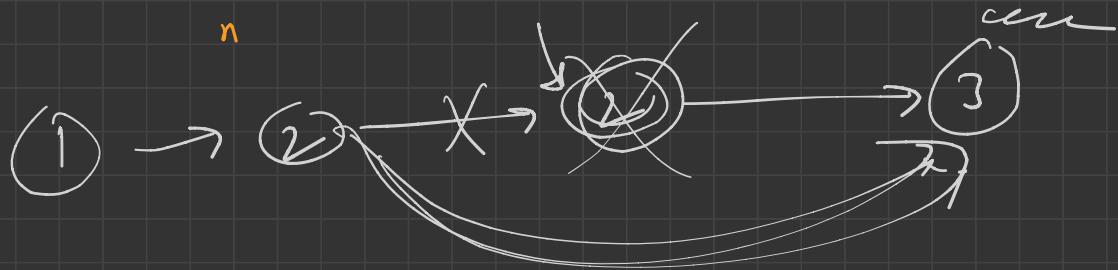
Code:-

$$ii = 2$$





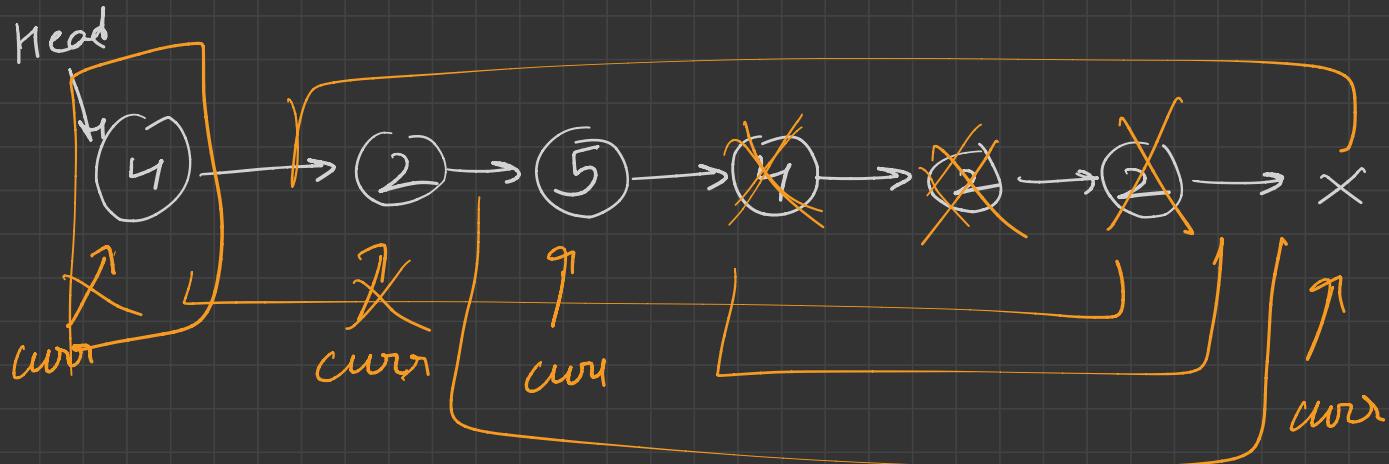
$$2 = \cancel{2}$$



$T \hookrightarrow O(n)$

$S \cdot \hookrightarrow \underline{O(1)}$

→ Remove Duplicates from Unsorted LL



1 element → aage ki LL banane karne ho

curr = head

while (curr != NULL)

{

Node \* temp = curr → next;

while (temp != NULL)

{

if (curr == temp)  
    ↳ data  
    ↳ data

d

deletion

}

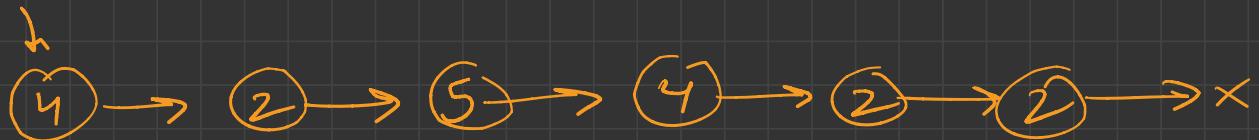
}

Homework

$\{$   
+ -  $\Rightarrow \mathcal{O}(n^2)$

$\leq = \mathcal{O}(1)$

head



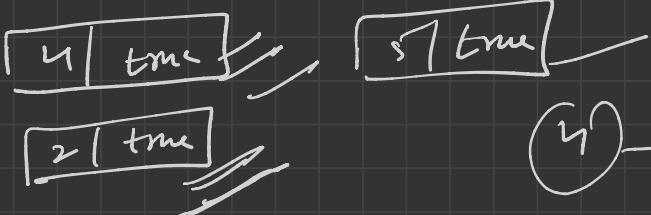
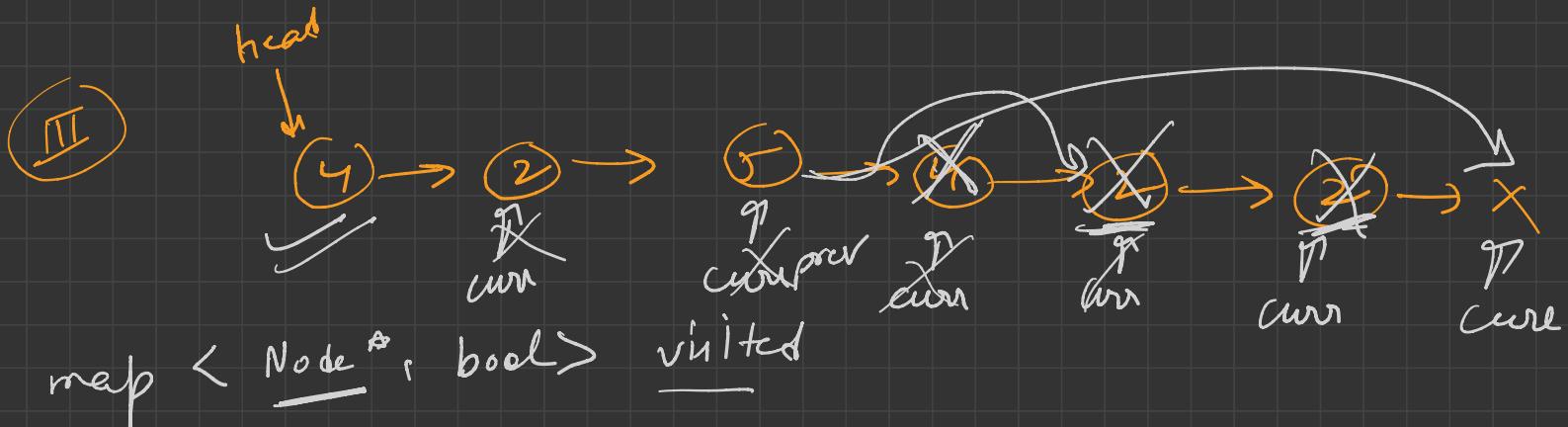
(II)

Step 1  $\rightarrow$  sort LL  $\mathcal{O}(n \log n)$

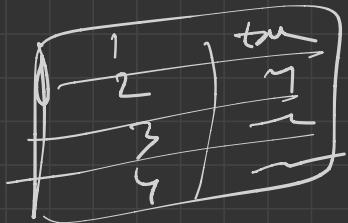
Step 2  $\rightarrow$  use the previous algo  $\mathcal{O}(n)$

$+ \cdot C \rightarrow O(n \log n)$

$S \cdot C \rightarrow \underline{\text{O}(D)}$



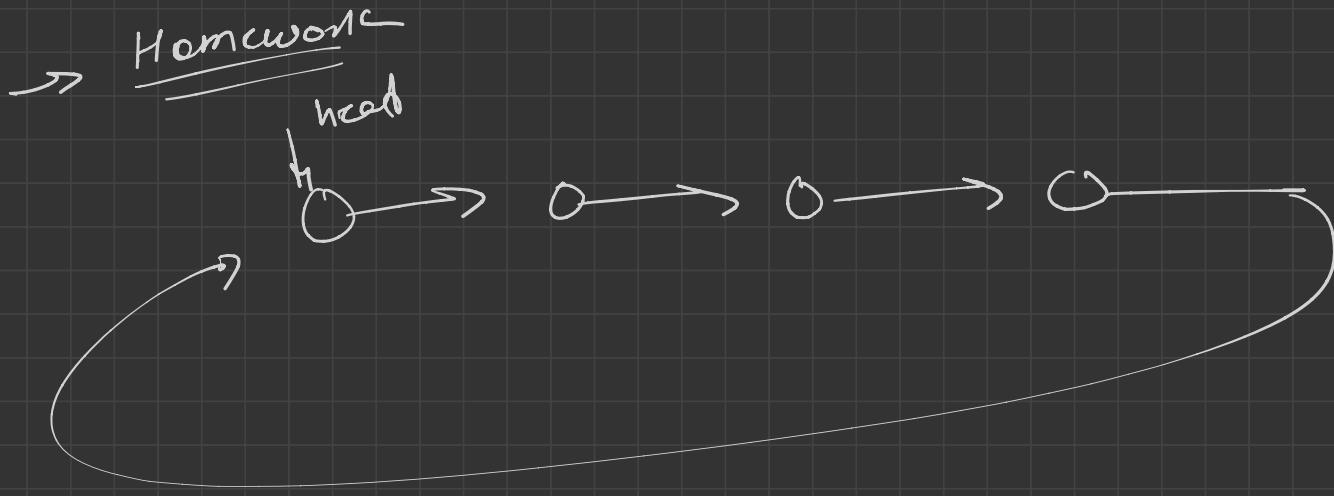
$T.C \rightarrow O(n)$   
 $S.C \rightarrow O(n)$



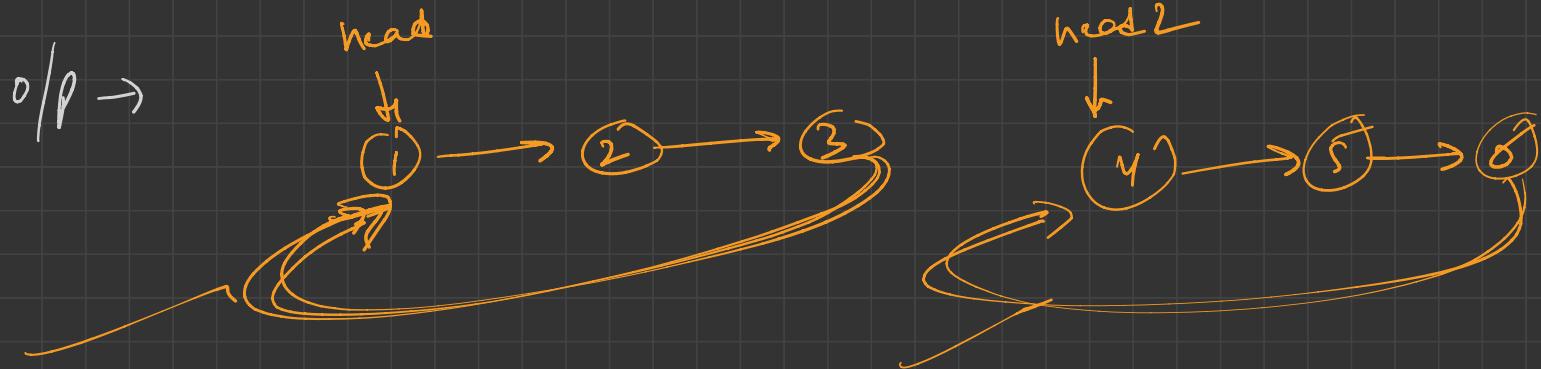
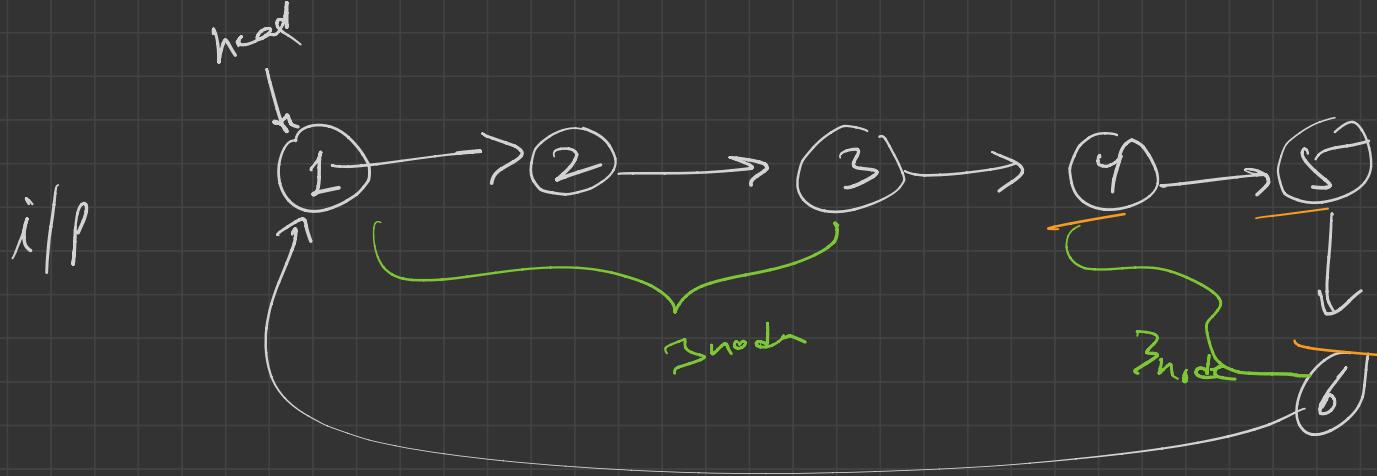
Remove Duplicates from Unsorted LL

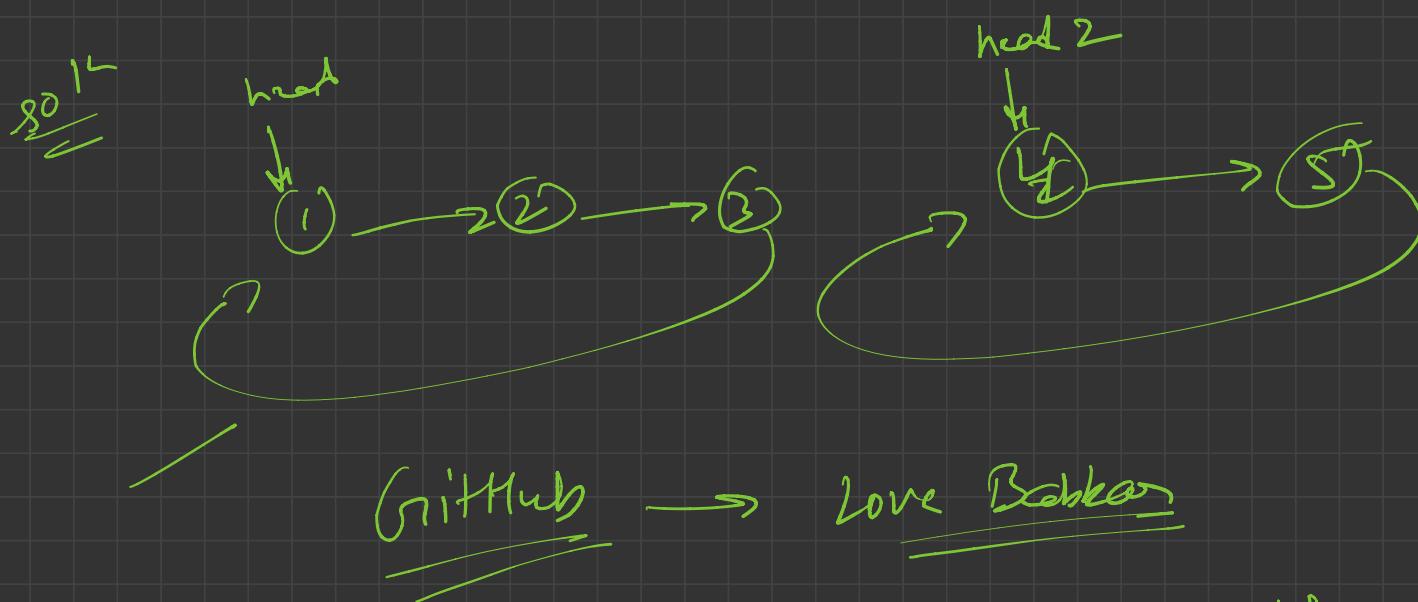
Homework

$O(n^2) \rightarrow$  2 Loops  
 $O(n \log n) \rightarrow$  sort  $\rightarrow O(n)$  when  
 $\rightarrow \text{map} \rightarrow O(n)$



split into 2 halves





Github → Love Bobkom

Contributor

Lecture 48































