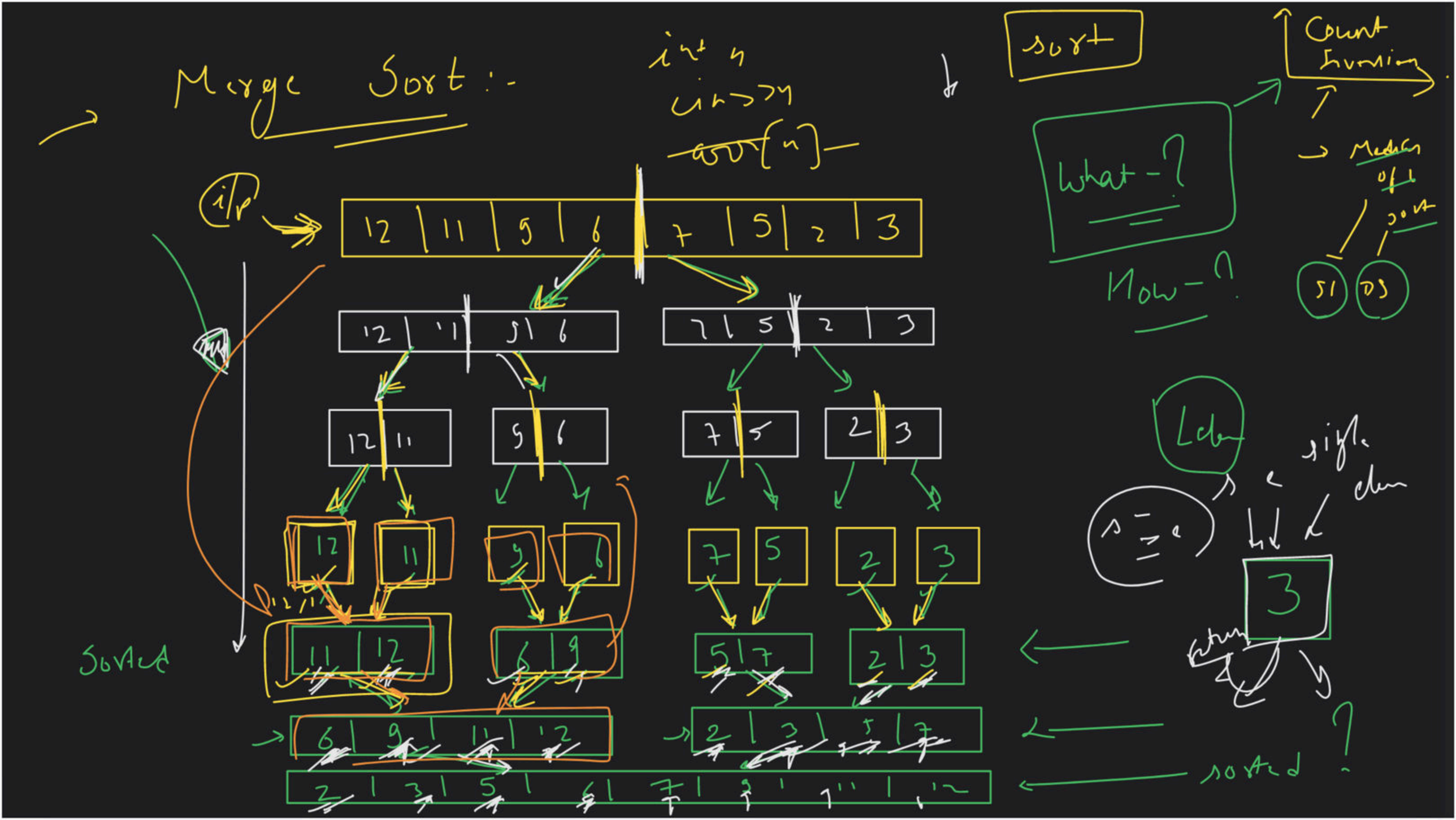
Divide & Conquer - Recursion

Foundation Course on Data Structures & Algorithm - Part I

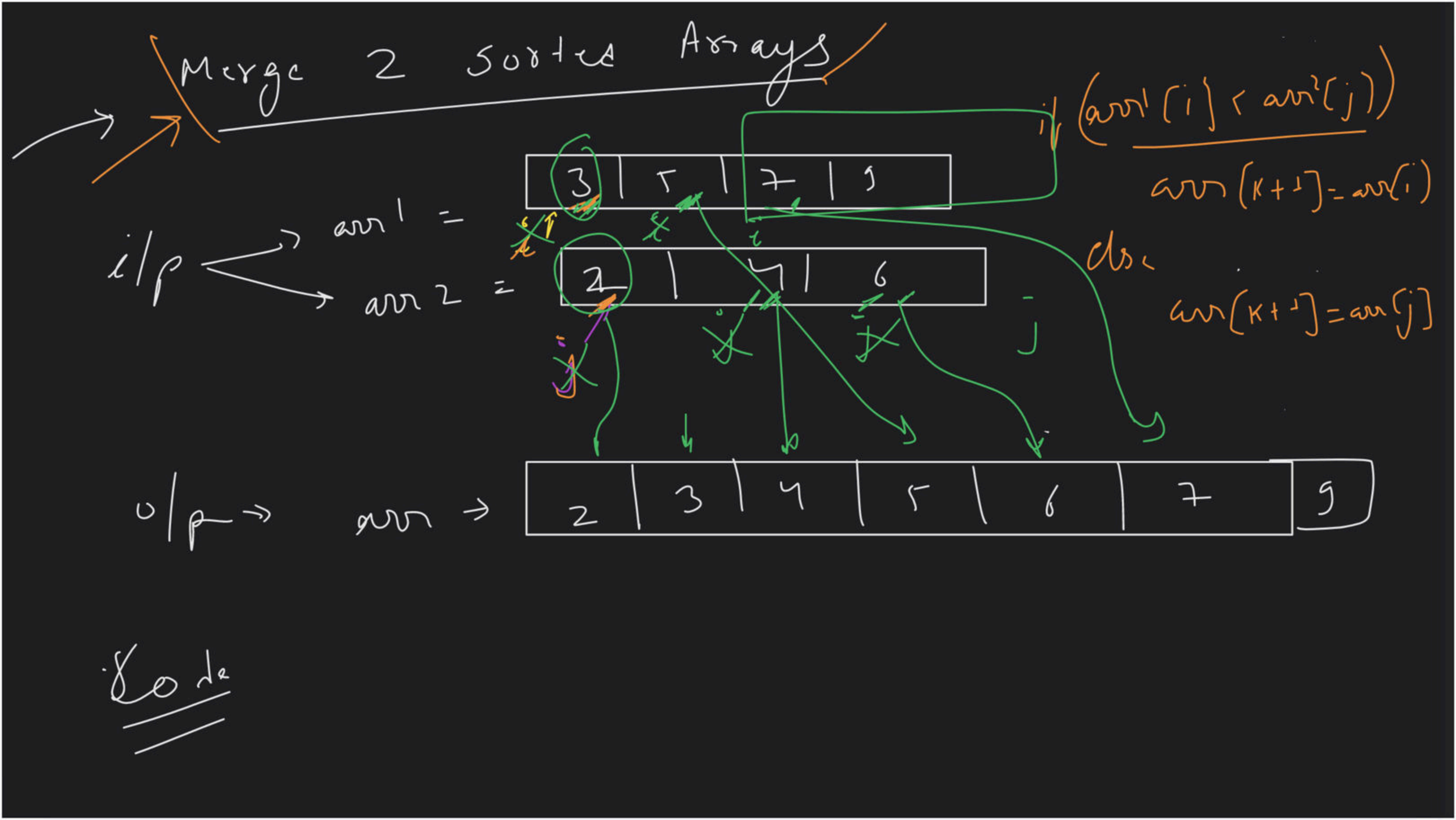
Love Babbar • Lesson 22 • May 31, 2022

→ Merge Sort → "Count Inversion" → Imp

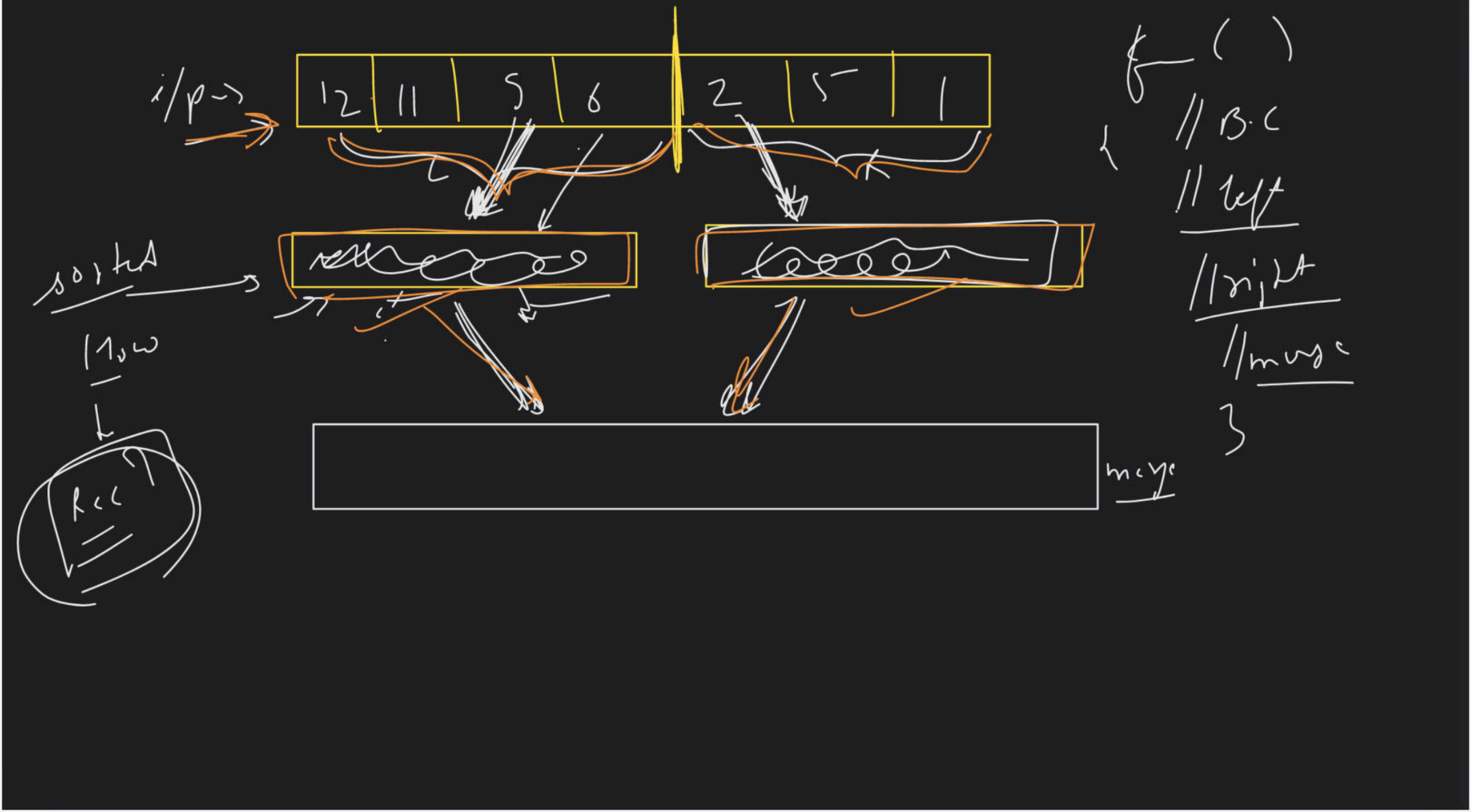
Merge Sort :-

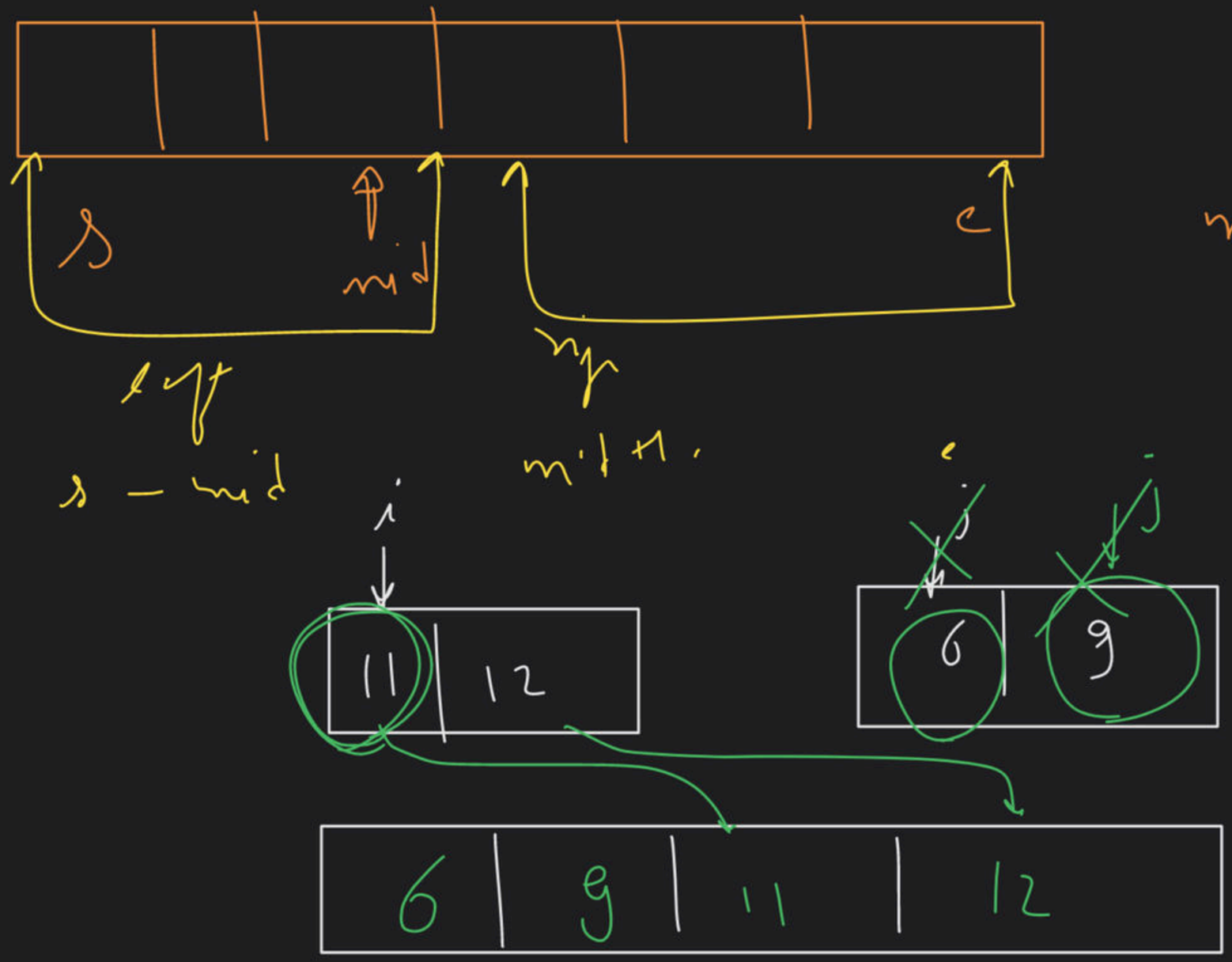

int n
i/p
12 | 11 | 9 | 6 | 7 | 5 | 2 | 3
Sorted
sort
Count Inversion
what -?
How -?
Merge of 2 sort
S1
S2
Left
right
3
sorted
?

Merge 2 Sorted Arrays

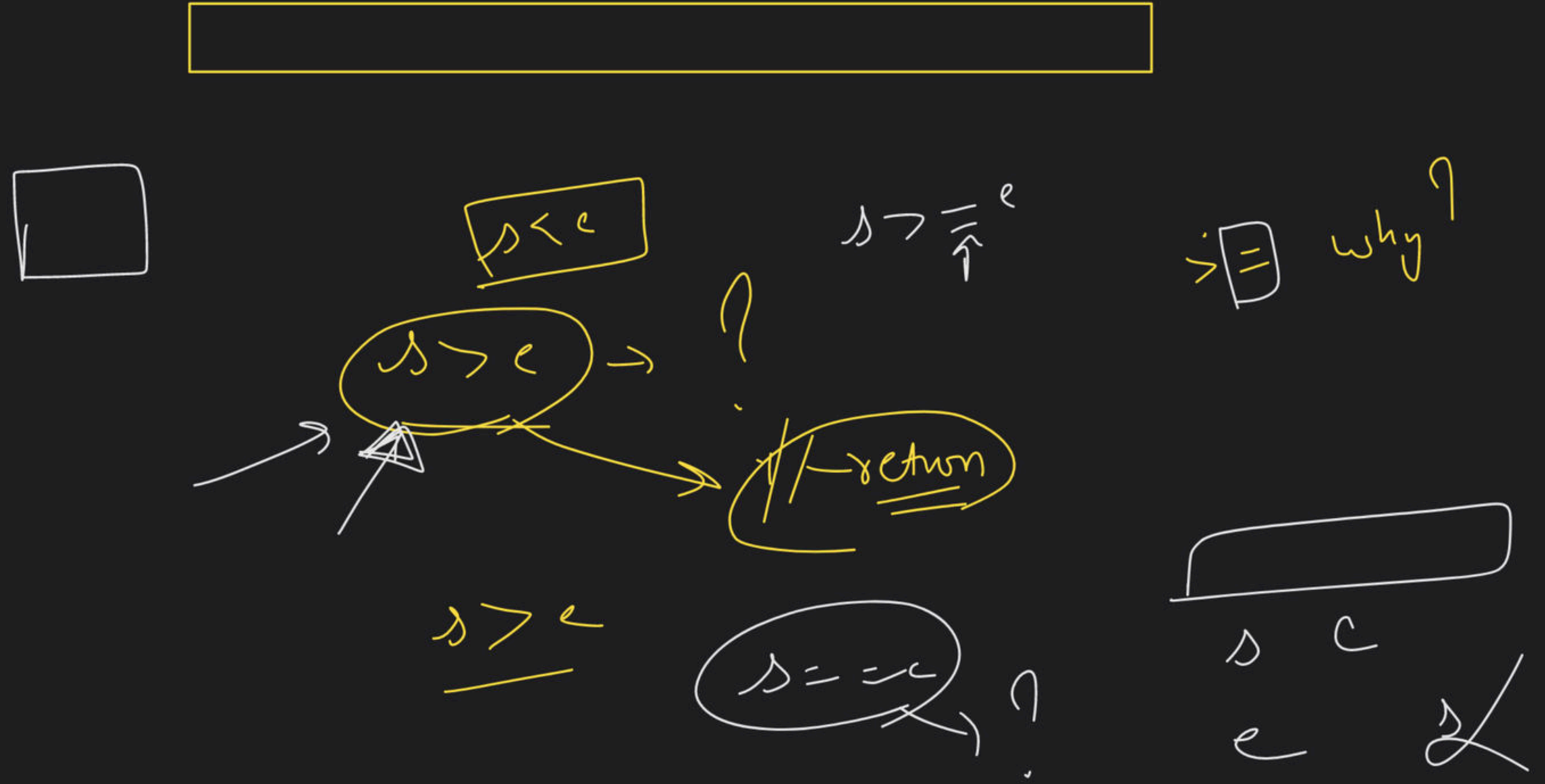


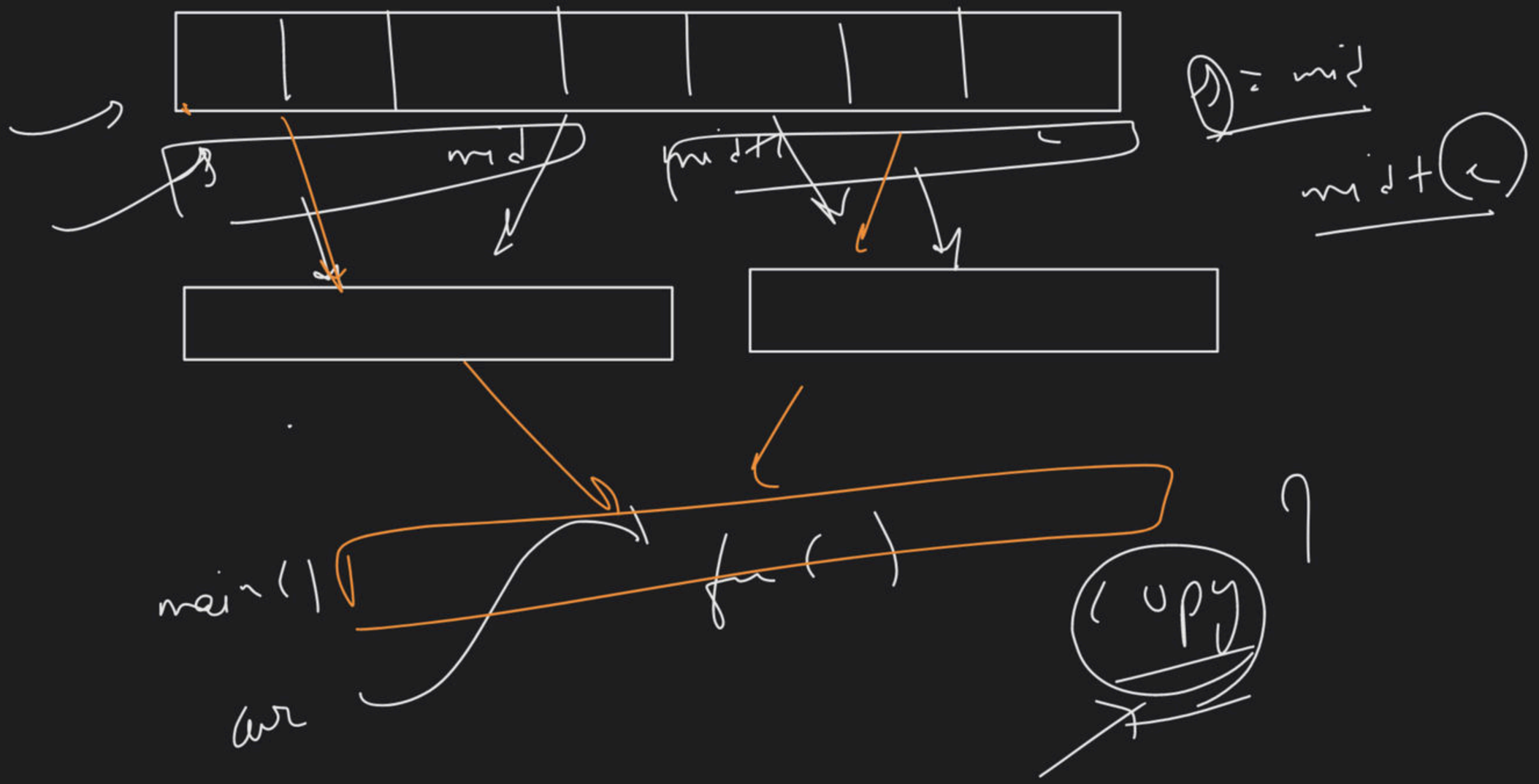
Code

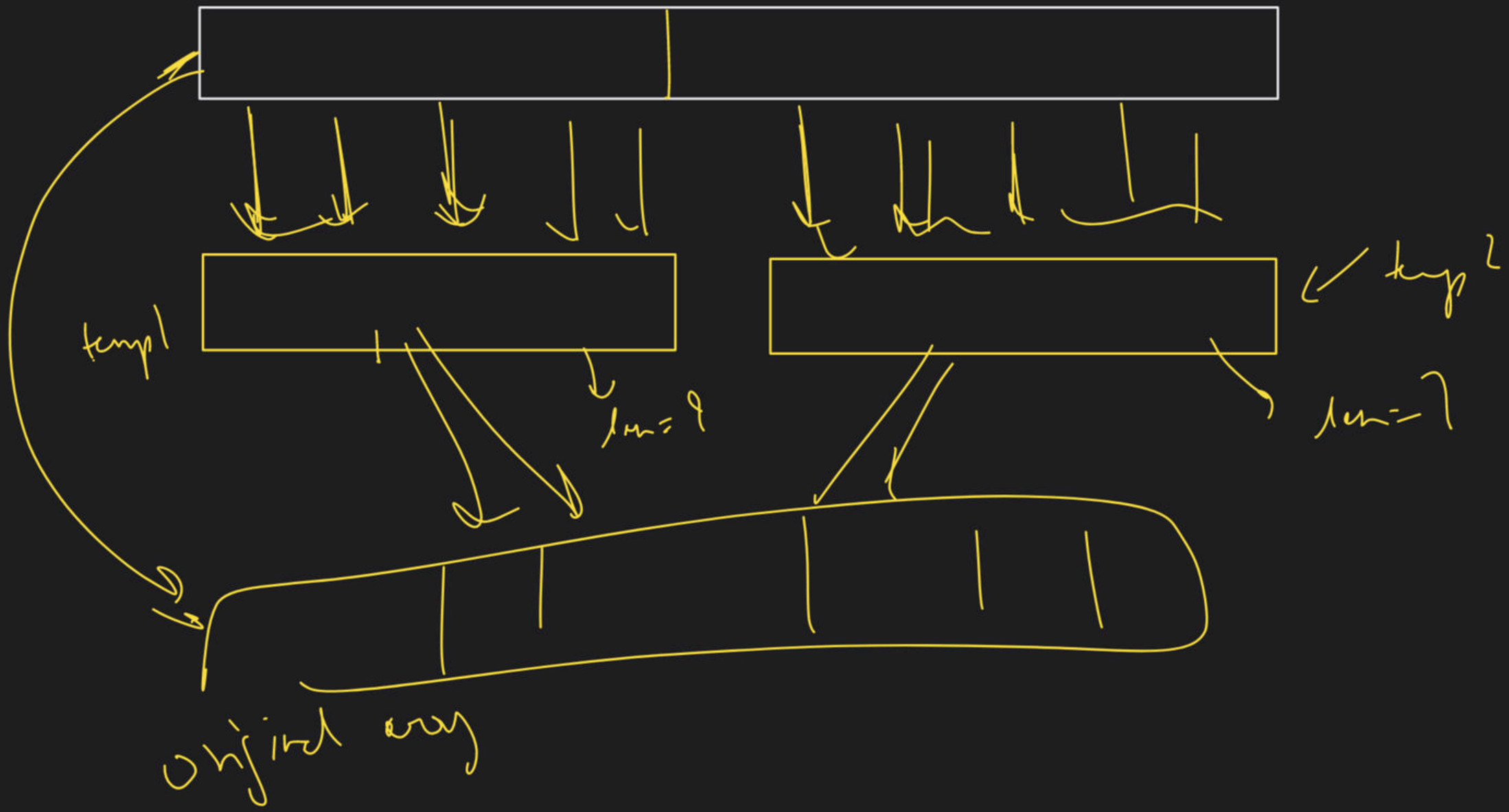

i/p →
12
11
5
6
2
5
1
sorted
low
high
merge
f( )
// B.C
// left
// right
// merge
}

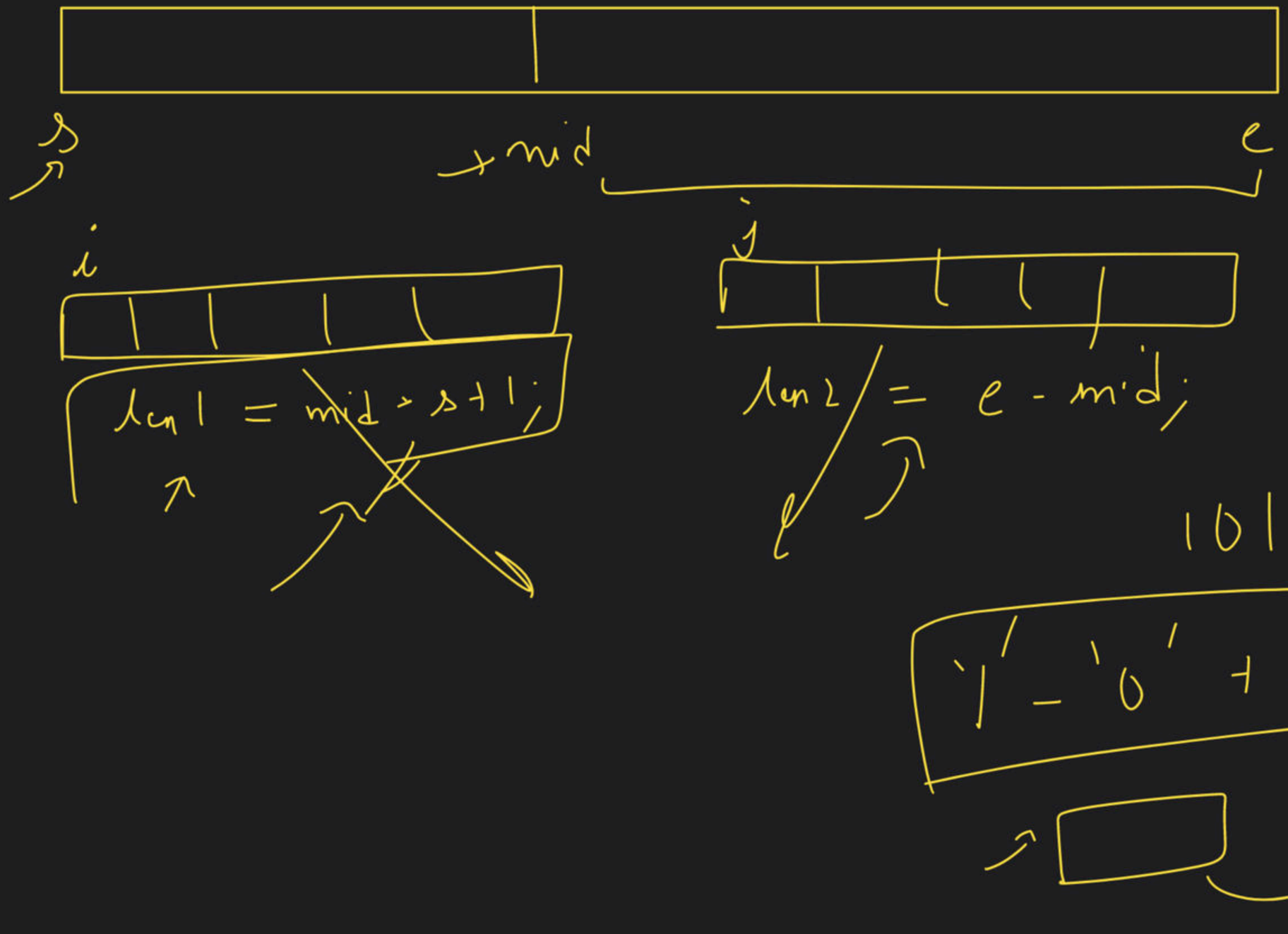


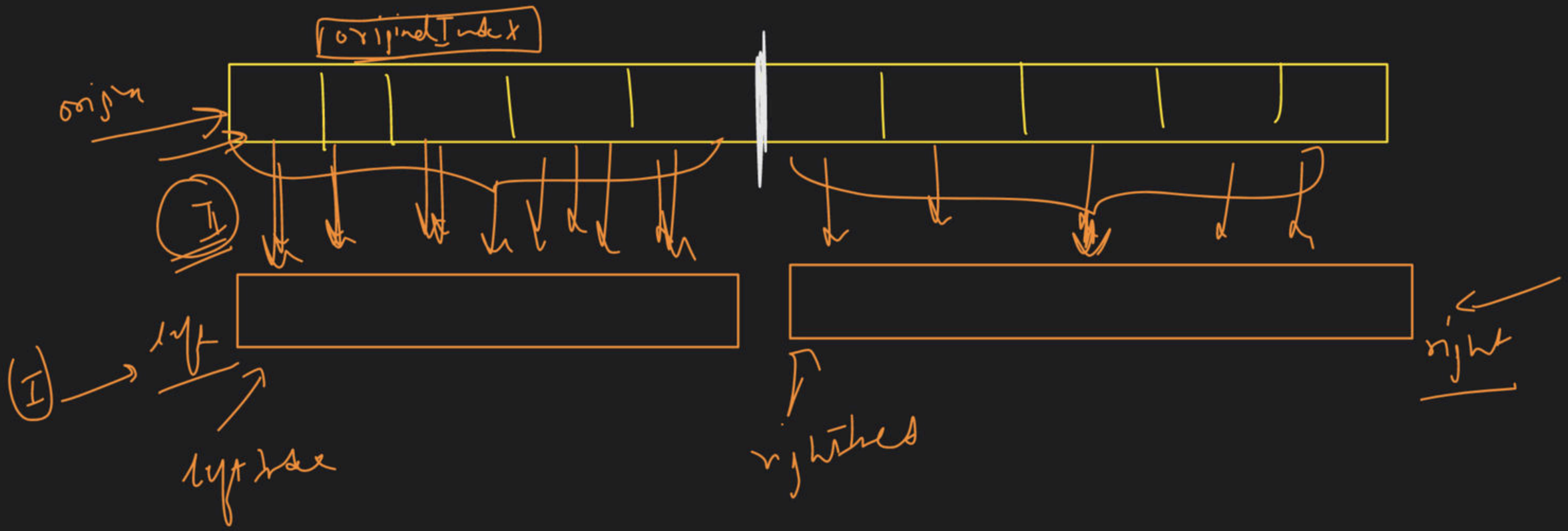
$$mid = \frac{0+5}{2} = 2$$

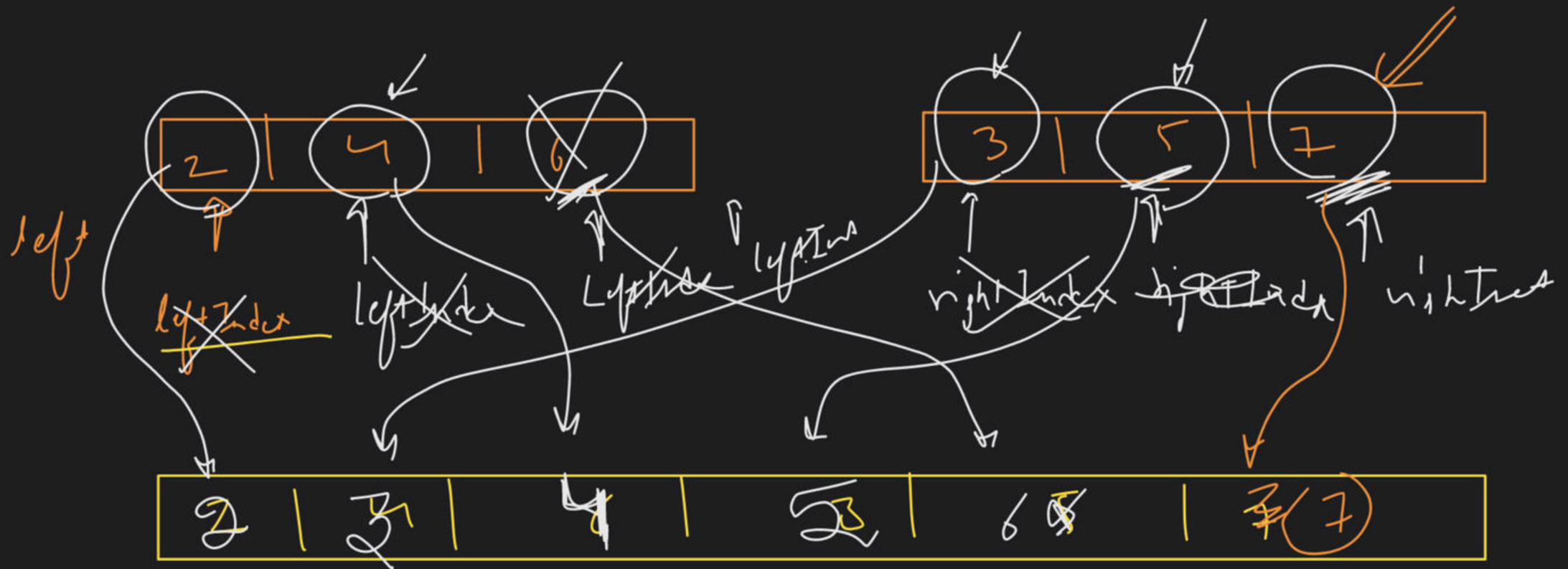

s<e
s>=e
why?
s>e
return
s>e
s==e
?
?
s
e
e
s


mid
mid+1
0 : mid
mid+1
main()
fun()
arr
copy


temp1
temp2
len=9
len=7
object array


s
mid
e
i
j
len1 = mid - s + 1;
len2 = e - mid;
101
'1' - '0' + 1


originalIndex
origin
I
I
left
left index
right
right index
right

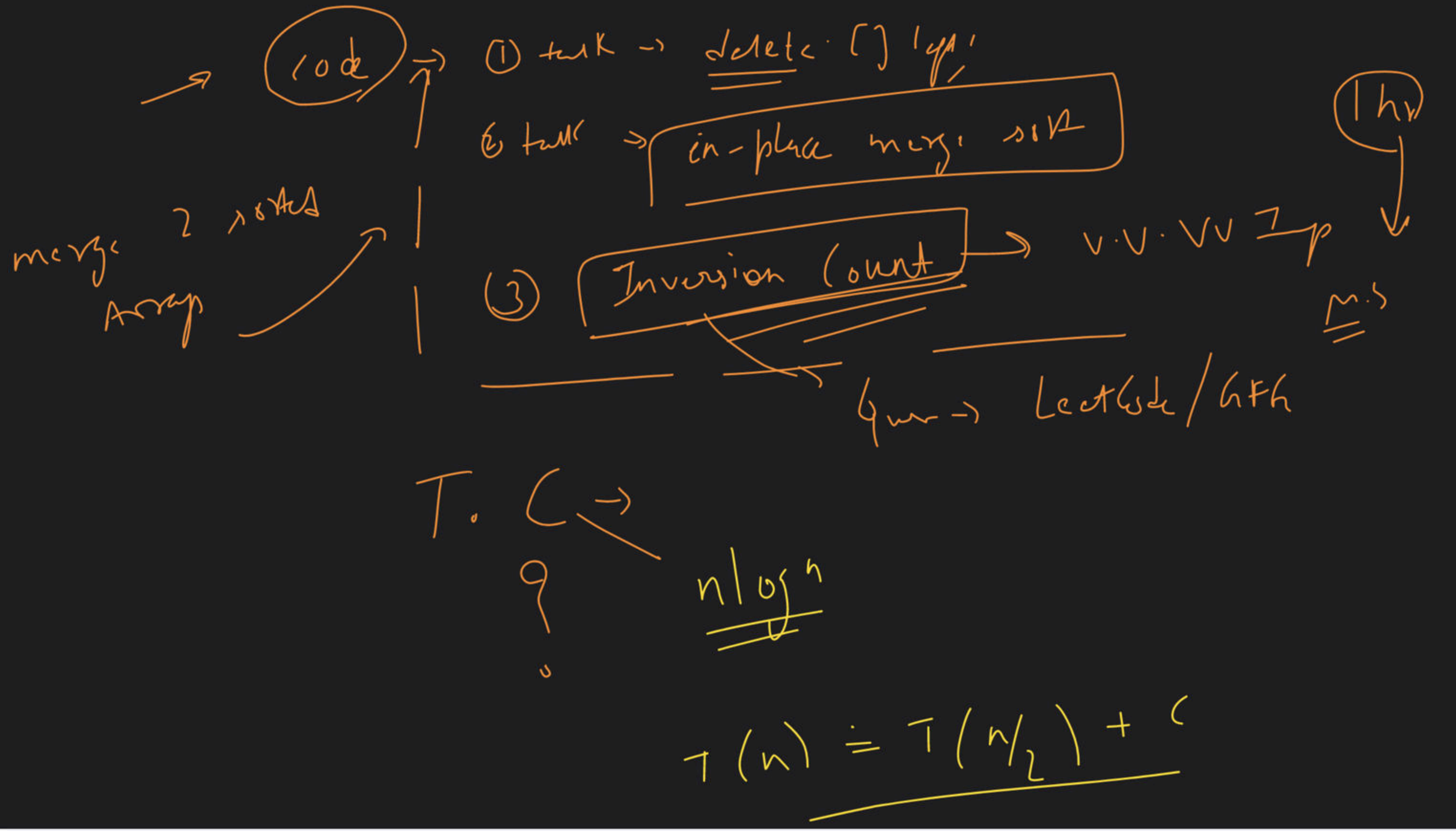


N/W

→ (1) delete [] arr — 1 term

→ (2) in-place merge sort →

No Voice ? Aayi ?


code
① task → delete [] 'ya'
② task → in-place merge sort
merge 2 sorted Array
③ Inversion Count
V.V.VV Zip
1 hr
M.S
Ques → LeetCode / GFG
T.C →
?
n log n
T(n) = T(n/2) + c

Quick Sort → ?

sort

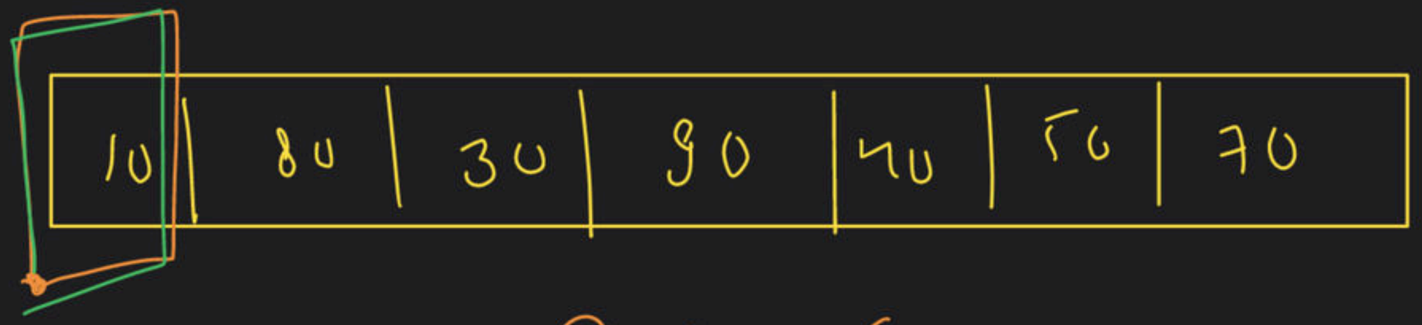
Interview

Student

fail

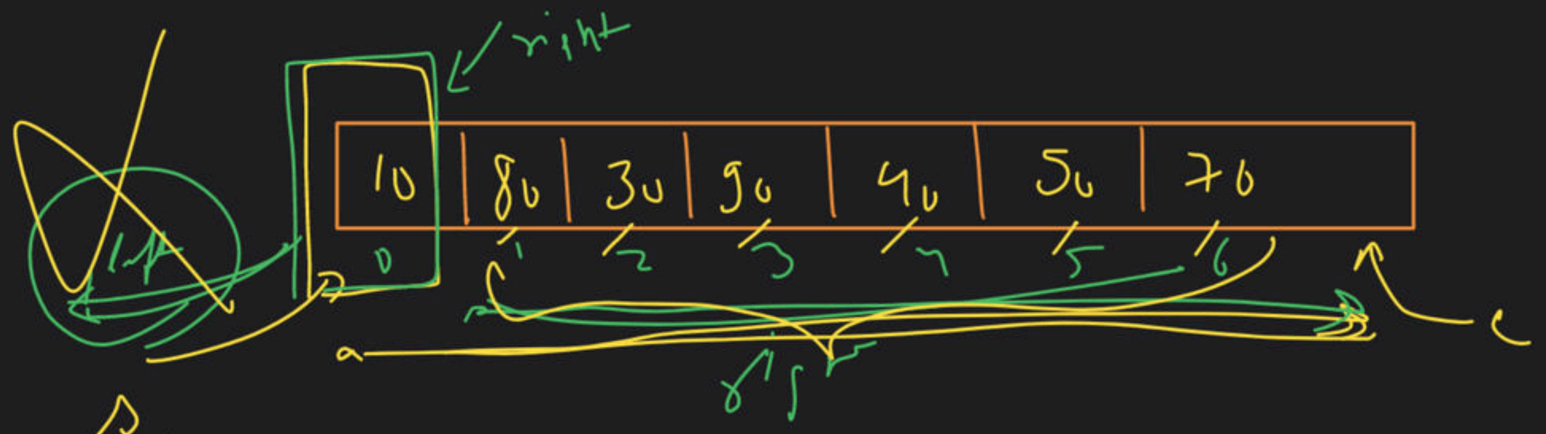
101%

arr =



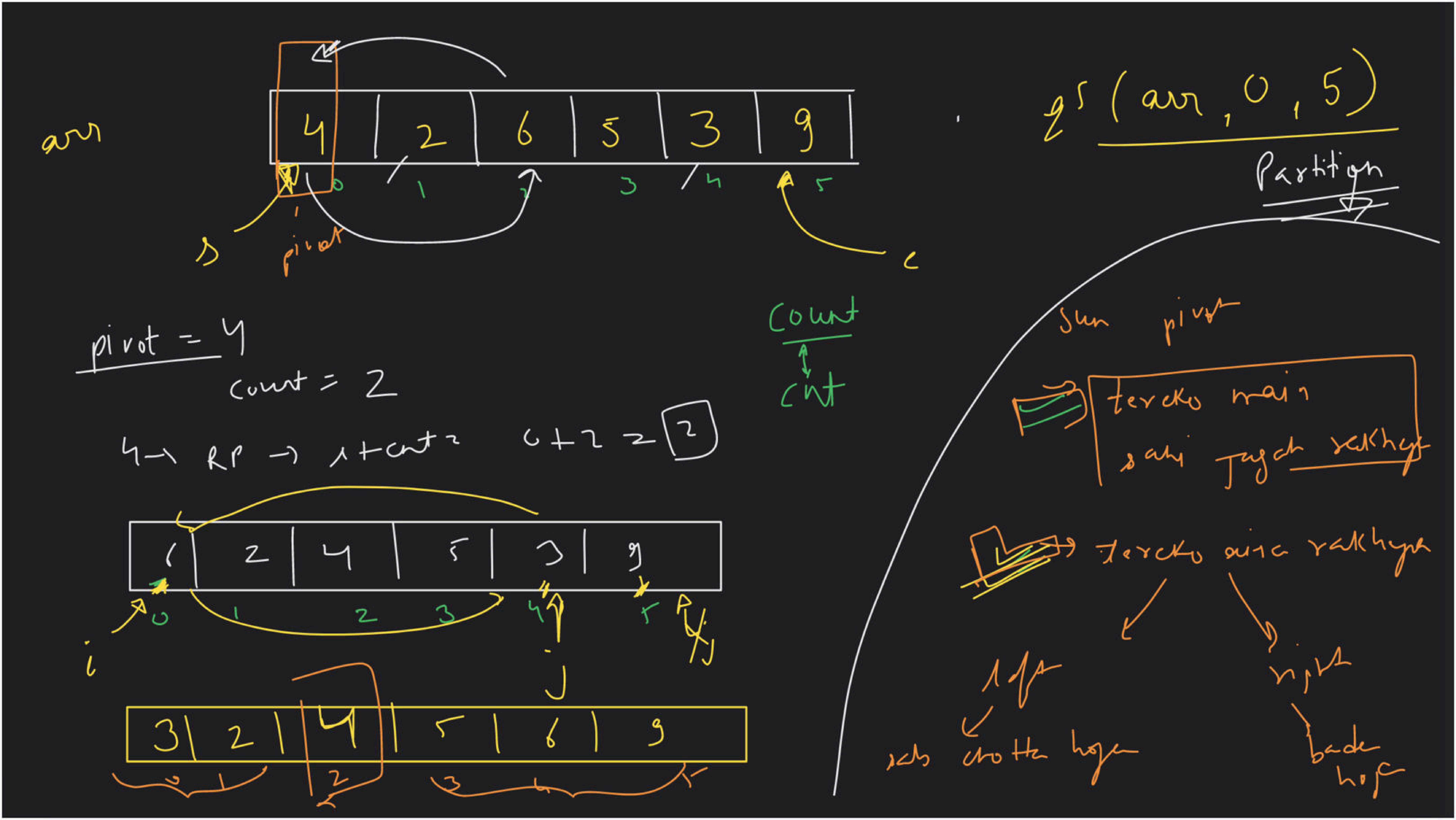
pivot = 10 → सही | गलत ?

right place → सही स्थान



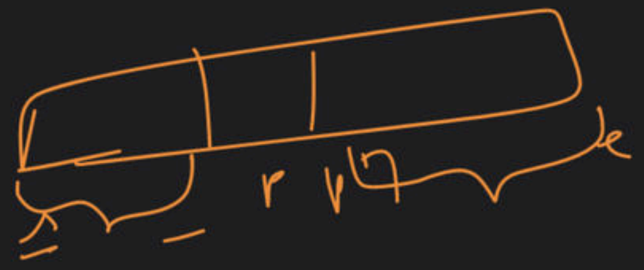
count = 0;

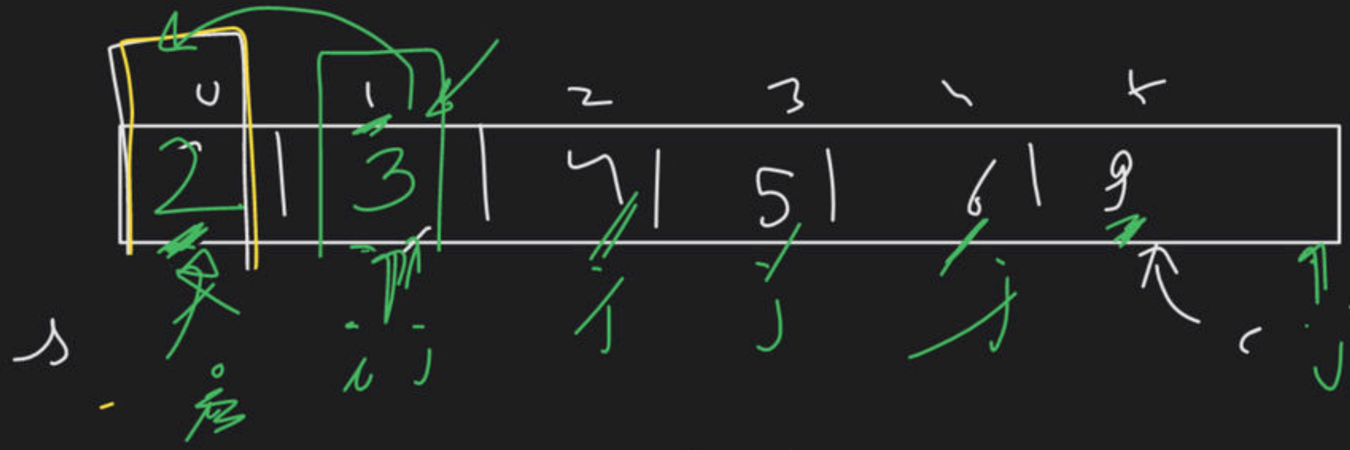
10 = st count = 0 + 0 = 0



arr
4 | 2 | 6 | 5 | 3 | 9
0 1 2 3 4 5
s
pivot
e
qs (arr, 0, 5)
Partition
count
cnt
pivot = 4
count = 2
4 → RP → s+cnt = 0+2 = 2
1 | 2 | 4 | 5 | 3 | 9
i
j
3 | 2 | 4 | 5 | 6 | 9
sun pivot
terko main
sahi jagah rakhyo
terko aisa rakhyo
left
right
sab chotta hoga
bada hoga

```
quicksort (arr, l, e)
{
    // b.c
    int p = partition( );
    // left
    qs(arr, l, p-1)
    // right
    qs(arr, p+1, e)
}
```





i < pivotIndex

j > pivotIndex

(1) 3 ke partition

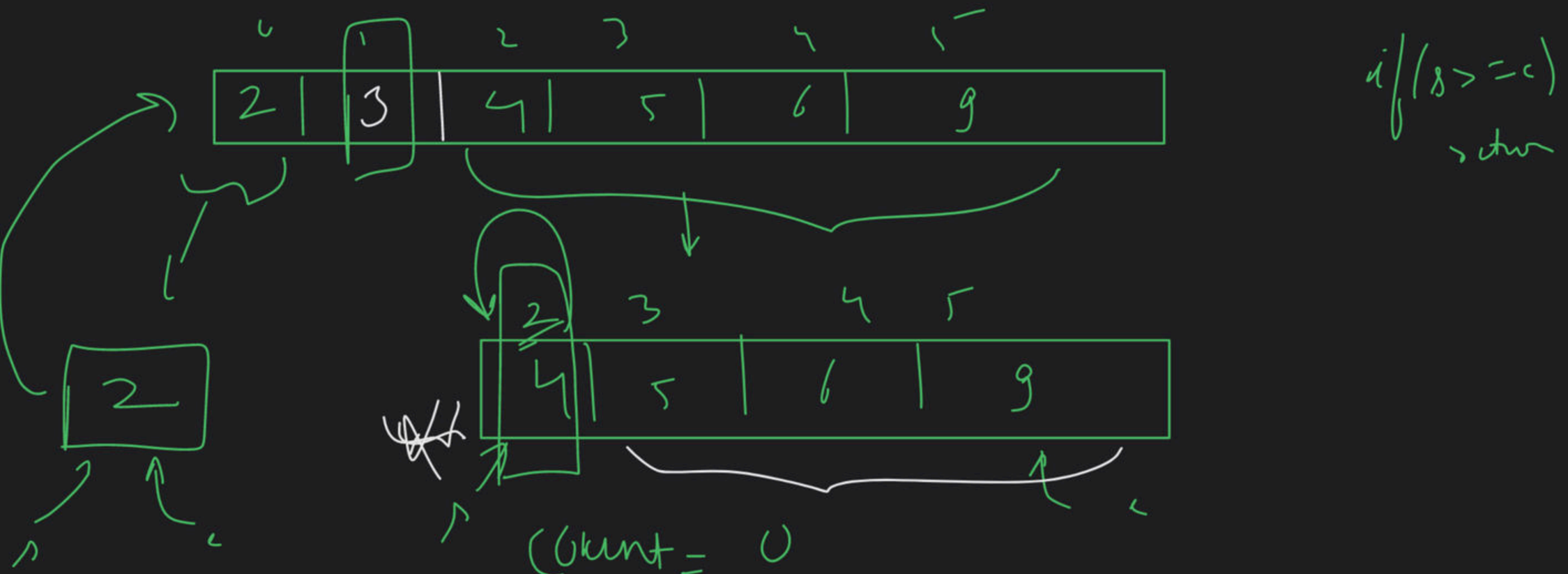
(A) Count element less than 3

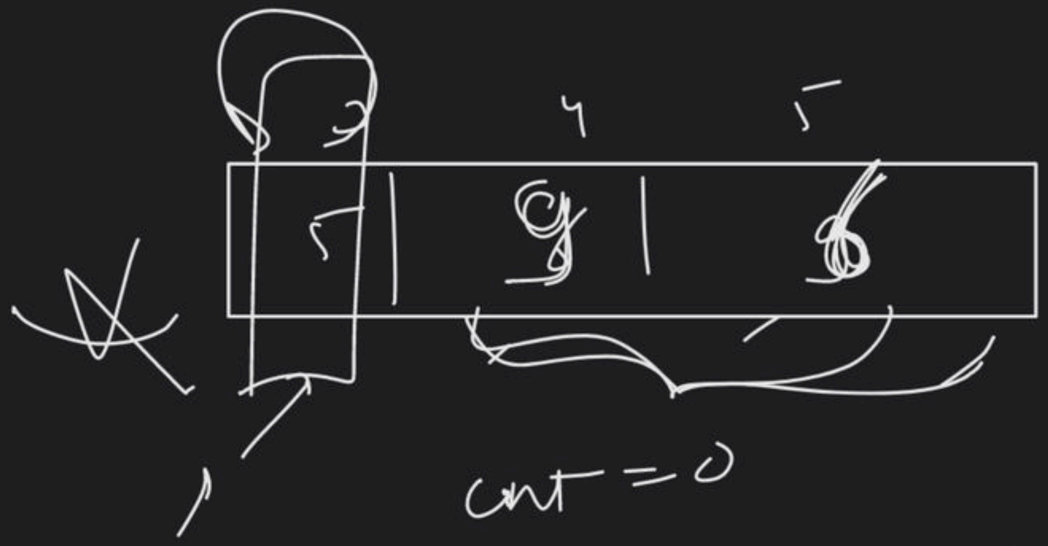
count = 1

pivotIndex → rightPlace = s + count = 0 + 1 = 1

(2) swap

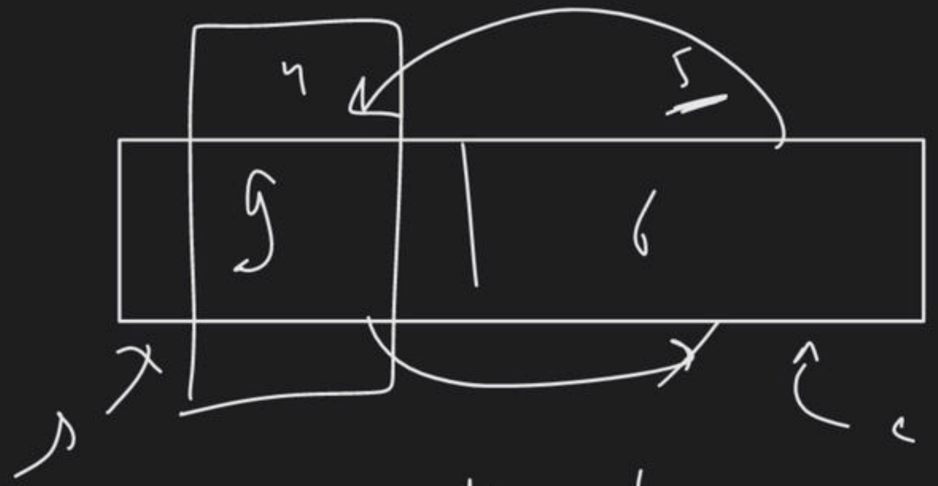
(B) ⟹ TRUE


0 1 2 3 4 5
2 3 4 5 6 9
2
2 3 4 5
4 5 6 9
if(s>=c)
count = 0
pivot index = s + cnt = 2 + 0 = 2



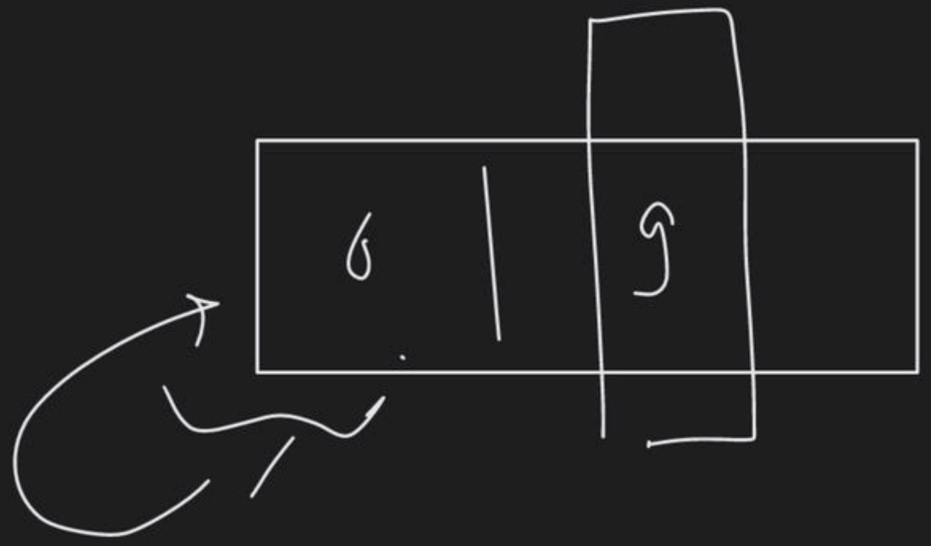
cnt = 0

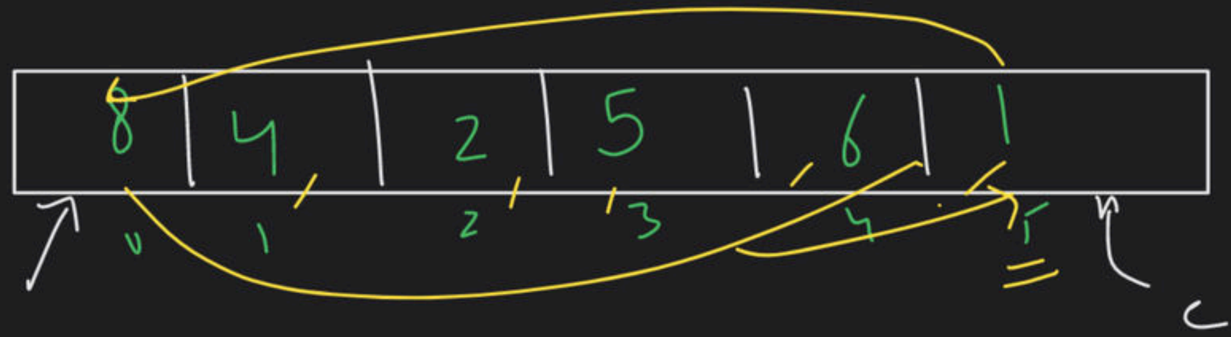
RP = start = 3 tv = 3



cnt = 1

RP = 1 + cnt = 4 + 1 = 5





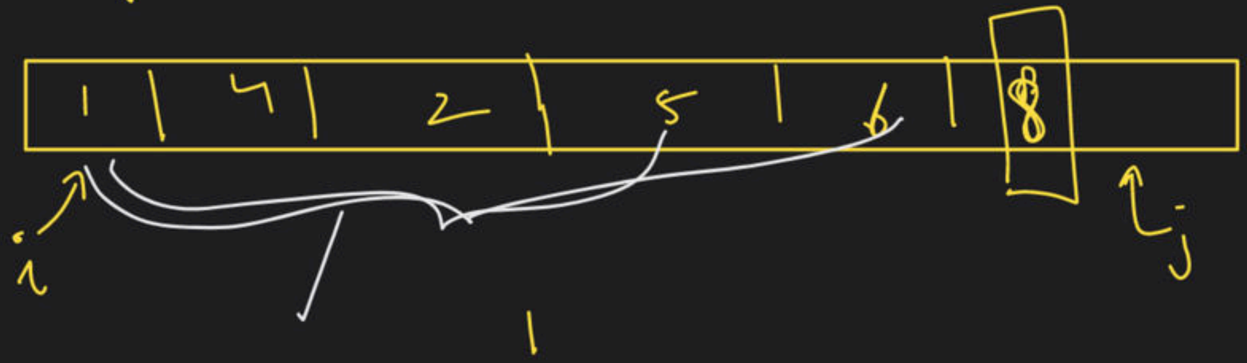
i < pivotIndex < j

j > pivotIndex

pivot = 8

count = 5

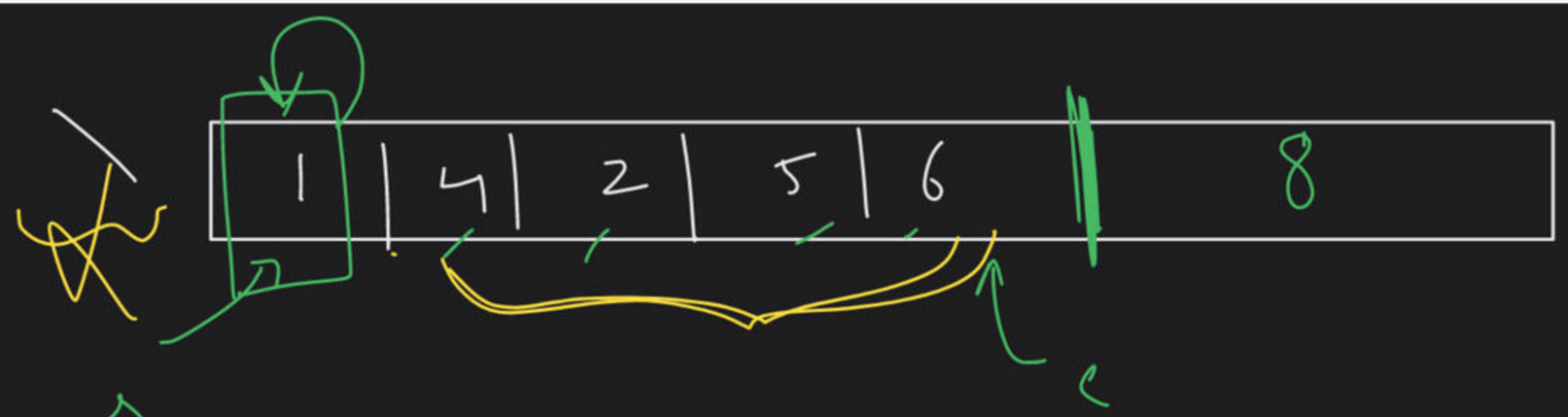
pivotIndex = 1st cut = 0 + 5



i < 5

j > 5

5 > 5 - 1 F



pivot = 1

count = 0

pivotIndex = 1st cnt = 0 + 0 = 0

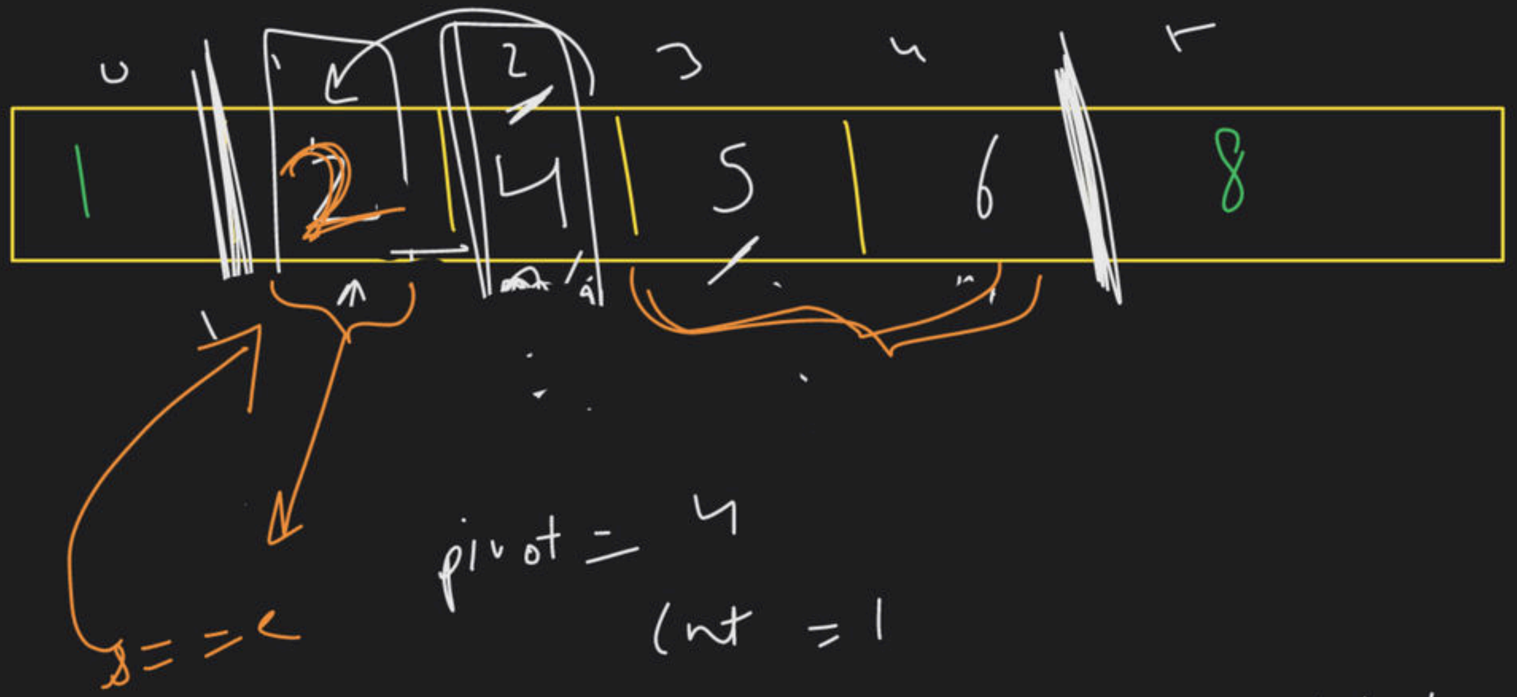
swap

i < pivotIdx

0 < 0

!

F



pivot = 4

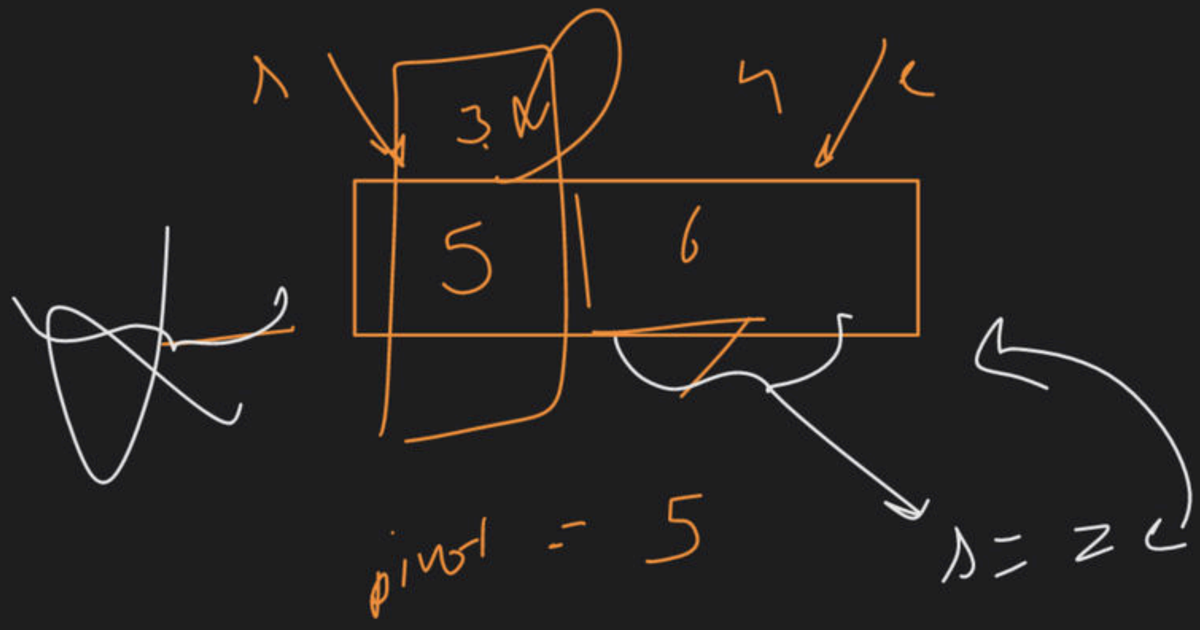
cnt = 1

s == e

pivotIndex = s + cnt = 1 + 1 = 2

Best Case

nlogn



cnt = 0

pivotIndex += s + cnt = 3 to 7

i < pivotIndex

3 < 3 -> F

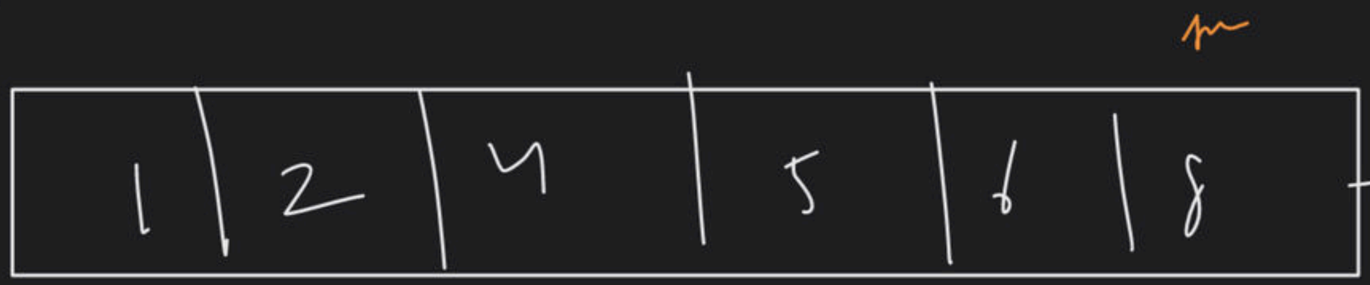
sort ( )

STL

implementation

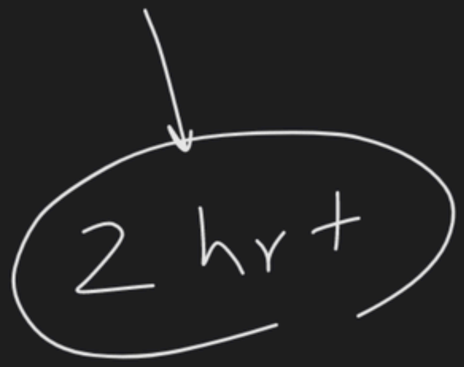
Worst

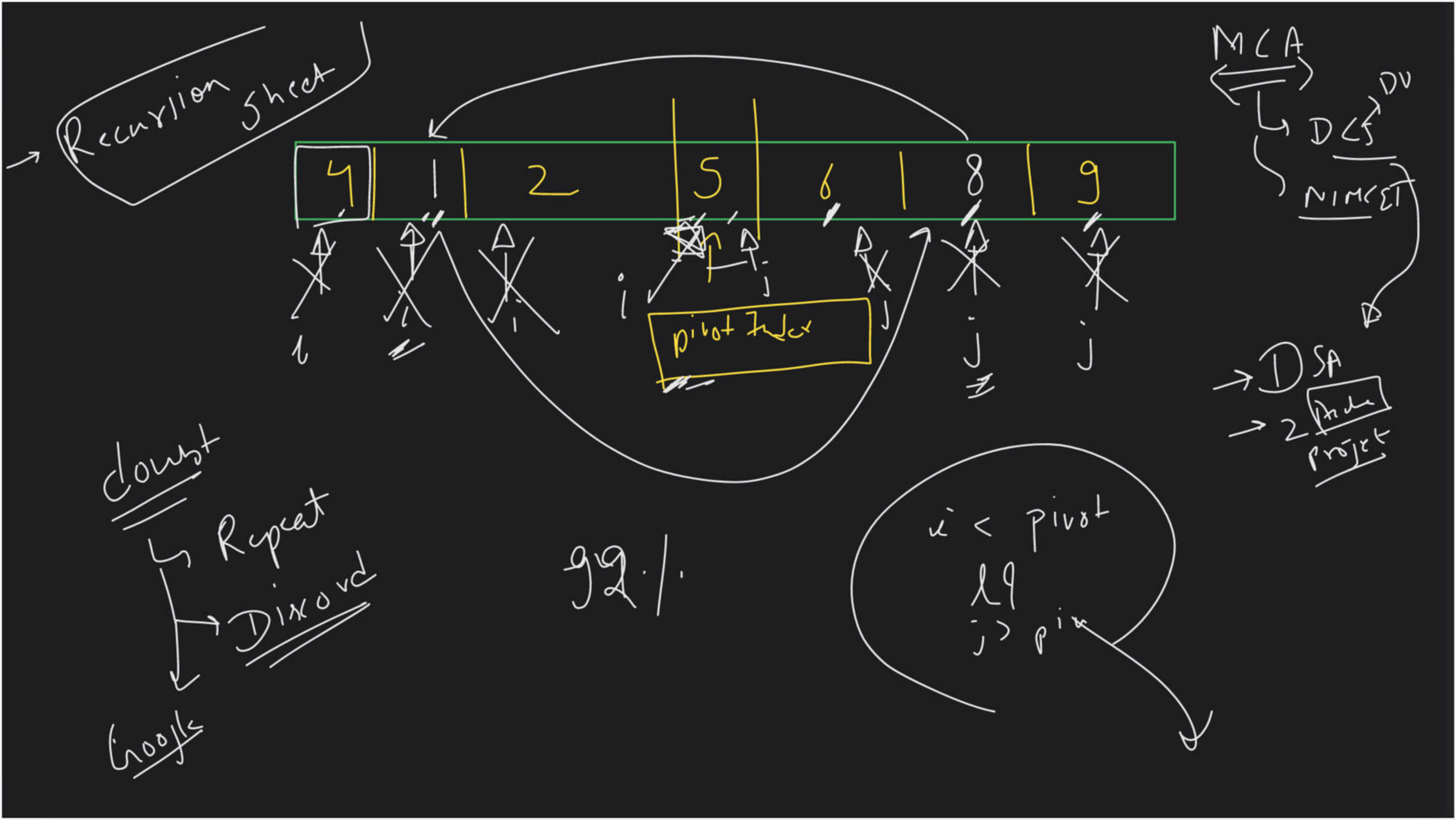
n^2

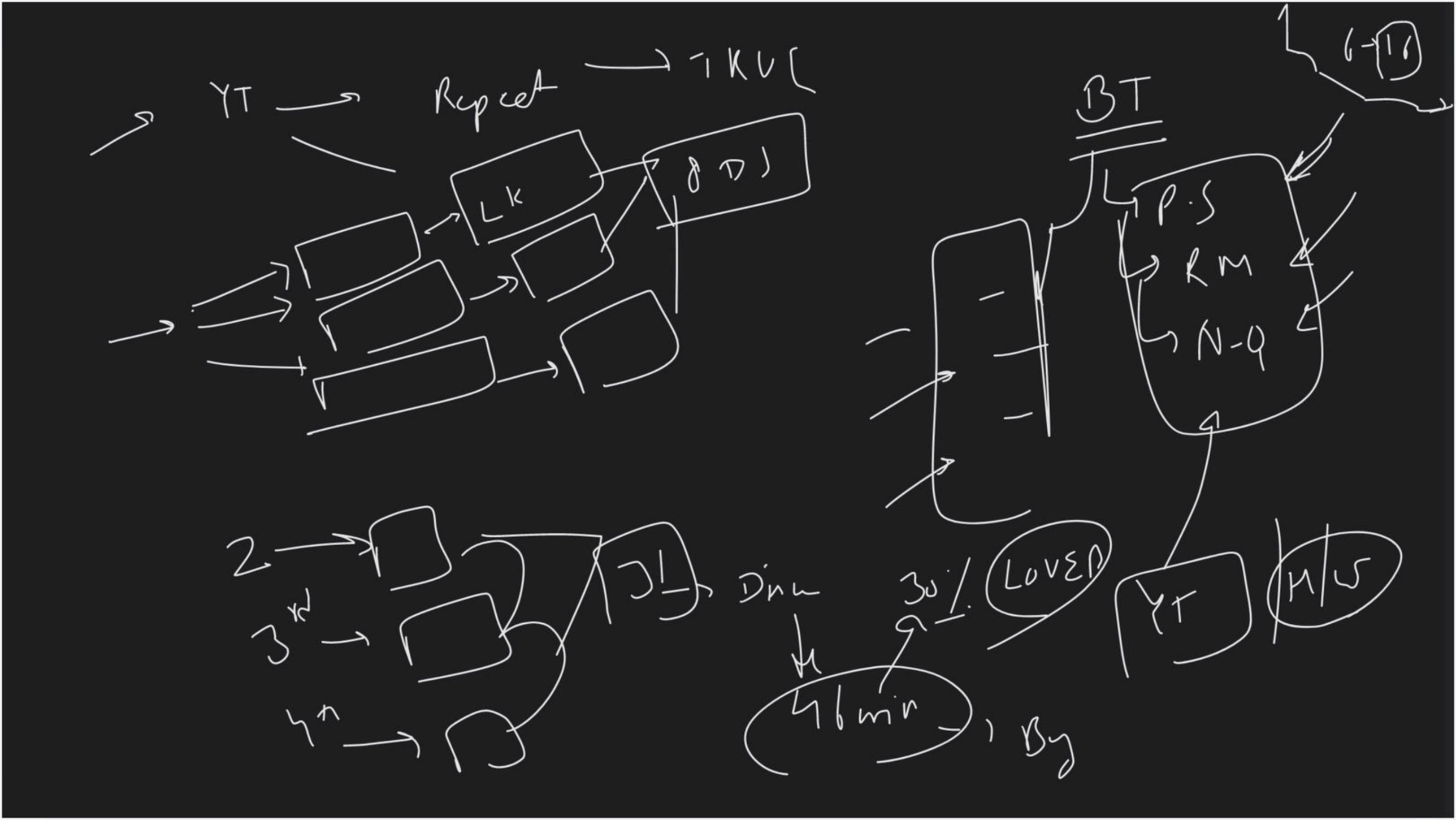


sorted ??

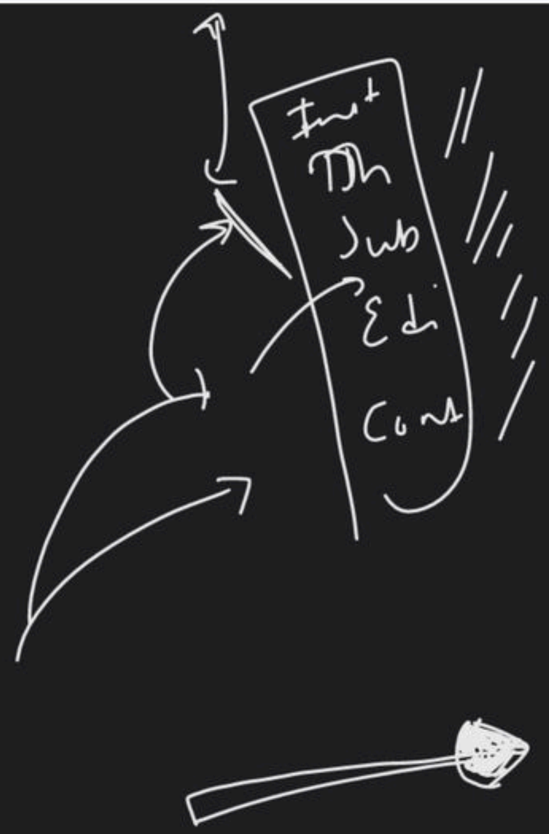

1.5 hr


2 hr +


Recursion Sheet
4
1
2
5
6
1
8
9
i
j
pivot Index
MCA
DCF
DV
NIMCET
DSA
2 Projects
doubt
Repeat
Discord
Google
92%
i < pivot
i9
j> pi


YT
Repeat
LK
DDJ
BT
P.S
RM
N-9
6-11
2
3rd
4th
DJ
Dinner
30 min
LOVED
YT
M/S
By


1.5
2:15 min


Inst
Th
Sub
Edi
Cont