Special Class

24 May 2022 20:08

Q :- [1, 2, 5, 4, 3] }

O/P [1, 2, 3, 4, 5] ✓

reverse any subarray then can you able to sort array?

[1, 3, 4, 10, 9, 8]

[1, 3, 4, 8, 9, 10] ✓

[1, 2, 3, 4, 20, 9, 8]

[1, 2, 3, 4 8, 9, 20] ✓

[1, 2, 3, 4, 20, 9, 10] =

[1, 2, 3, 4, 9, 10, 20]

Q [1, 2, 5, 4, 3]

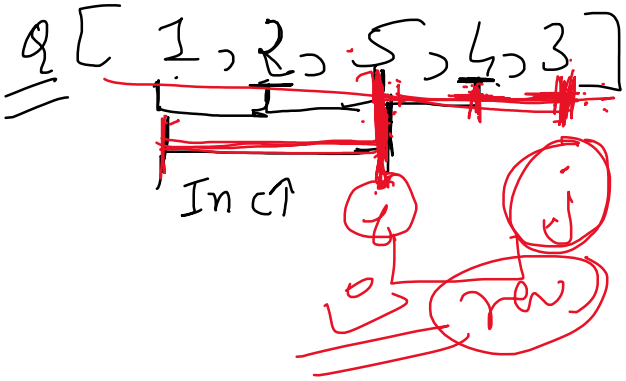
5 mins :-

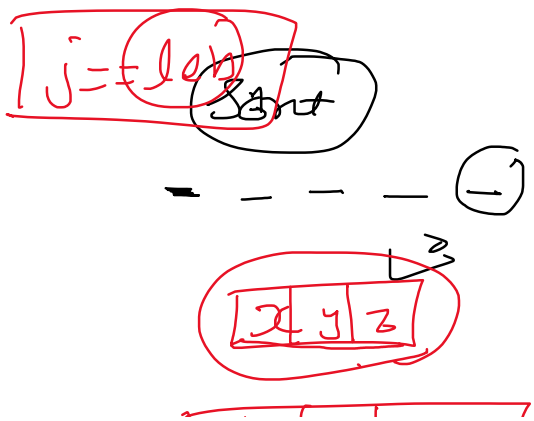
✓

[1, 2, 3, 4, 5]

8:26

8:31





(1) Upto which Point sorted

is incrementing values.

② upto which point array is decrementing values

[45|60|100]

[45|60|15] ✗

[45|30|20]

[1, 2, 5, 4, 3]

```
for(i=1; i<len; i++)
{
    if(arr[i] > arr[i-1])
    {
        if(i == len)
```

i=3

1 → ihs

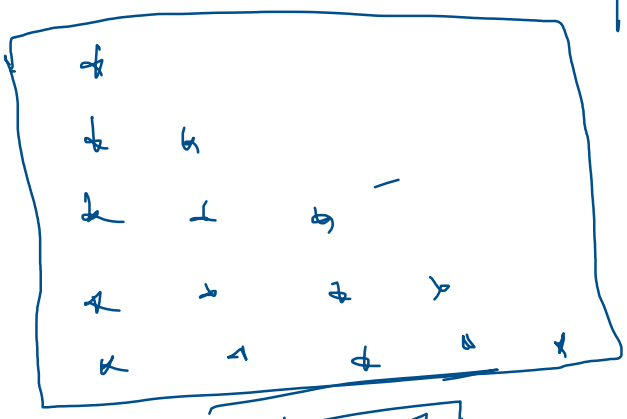
2>1

5>2

4>5

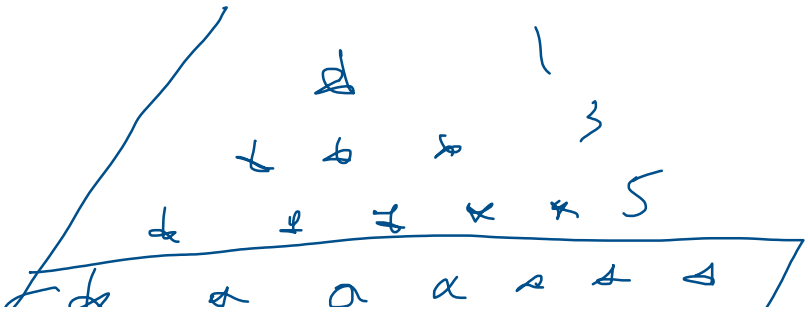
⊛ break & continue:-

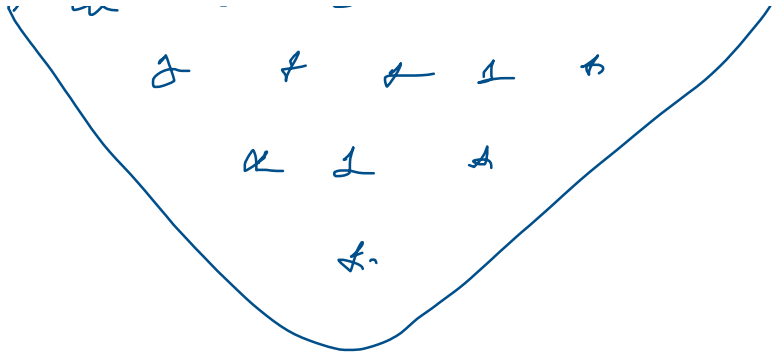
slice()



5 mins

22:12





1 2 3 4 5 6

2 3 4 5 6

3 4 5 6

4 5 6

5 6

6

1520 ⇒ 9

1+5+2+0+1

sum

i+0

num%10 0

Pureset(num/10)

1520

num>0

num=0

[10, 50, 100, 250, 0]

min
max

{0,1}

Array methods:-

1. sort()

sort is function

sort( )

function.

sort((a,b) => a-b)

sort((a,b) => b-a)

callback

higher order

Advanced JS

Arguments ⟹ logic

⇒ Any fn accepting another fn

callback fn

2. Slice

① reverse subarray problem

② Pattern problem ⇒ Pascal Pyramid Numbers

③

3. sum of digits / maximum max
4. array methods
5. watch / watches
6. callback fn :-

---

* Problems with Array :-

Q. Sparse Array :-

$$\begin{bmatrix} 0 & 1 & 2 \\ 5 & 0 & 6 \\ 1 & 0 & 0 \end{bmatrix}$$

9

100 %

51 %

49 % $\left(\frac{9}{2}\right)$ > 4

no. of zeros :- 4

total no. of elements => 9

$\left(\frac{9}{2}\right) \Rightarrow 4$

$\left(\frac{\text{total no. of element}}{2}\right)$ > 4

Q.

$$\begin{bmatrix} 0 & 1 & 3 \\ 5 & 0 & 6 \\ 1 & 0 & 0 \end{bmatrix}$$

5 mins

20 : 21

20

this Sparse or not?

Q :- [1, 0, 0, 1, 1, 0]

2 pointers make 0 -> left

3

1 → right

[2, 0, 1, 0, 0, 1, 2]

Sort

3 Pointers

0 ← 1 2 →

Q. [3, -2, 5, -4, 1, 6]

[-2, -4, 3, 5, 1, 6]

-ve ← +ve

Approach

Partial

Pivot => 0

Partition algo

7 mins

arr=[3, -2, 5, -4, 1, 6]

i=-1, j=0 3 pivot=0

Step 1:-

① arr[0] => 3

3 < 0 } => no/false => j++

② arr[1] => -2

-2 < 0 } => yes => i++ => i=0

now, a[i], a[j] => swap => j++

[-2, 3, 5, -4, 1, 6]

i=2, i=0

arr[2] => 5

5 < 0 => j++

j = 3 & 5 i = 0 1

a[4] => -4 < 0 } true i++

swap a[i], a[j] && j++

[-2, -4, 5, 3, 6]

1 < 0 } j++

-ve | pivot | +ve

-1

i = -1, j = 0

6 < 0 } j++

1. a[j] < pivot } i++

swap a[i] & a[j] & j++

2. a[j] > pivot } j++

I/P [1, 2, 3, 4] => Cycle rotation

Swing

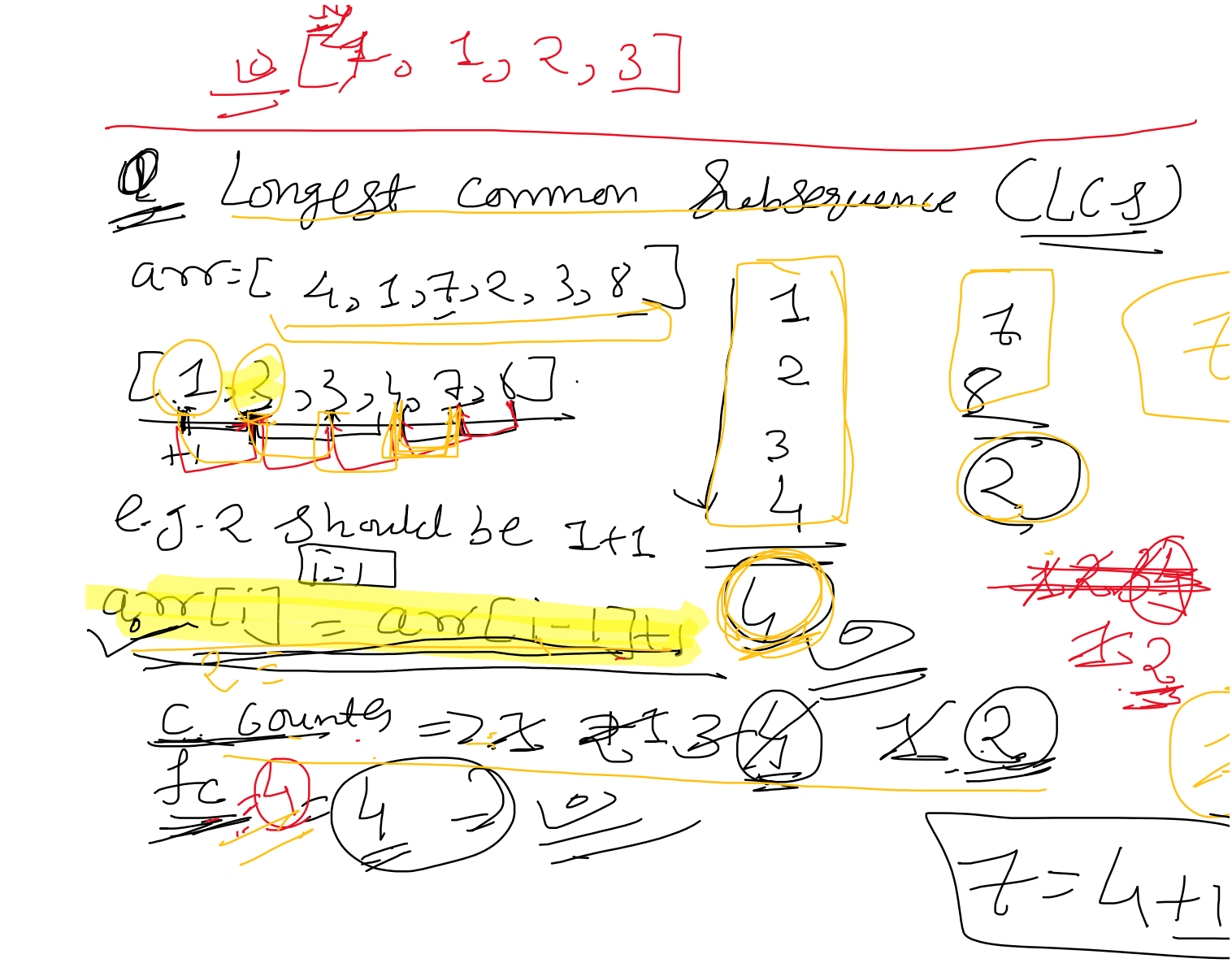
Intermediate

O/P [4, 1, 2, 3]

Self logic

temp = 4

22:18



1. Sparse array :-
2. Shift -ve to left +ve to right
3. Cyclically rotate array 1
4. Longest consecutive seq
5. Swapping → destructuring

20 – ①

Doubt

2D → ②

Adv Javascript – 1

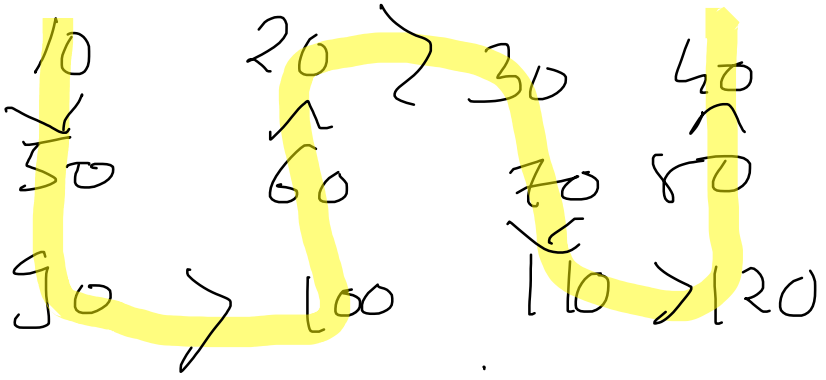
)L — — 2

— — 3

Doubt

Wave f

Q



O/P

10, 50, 90, 100, 60, 20, 30, 70, 110

80, 40

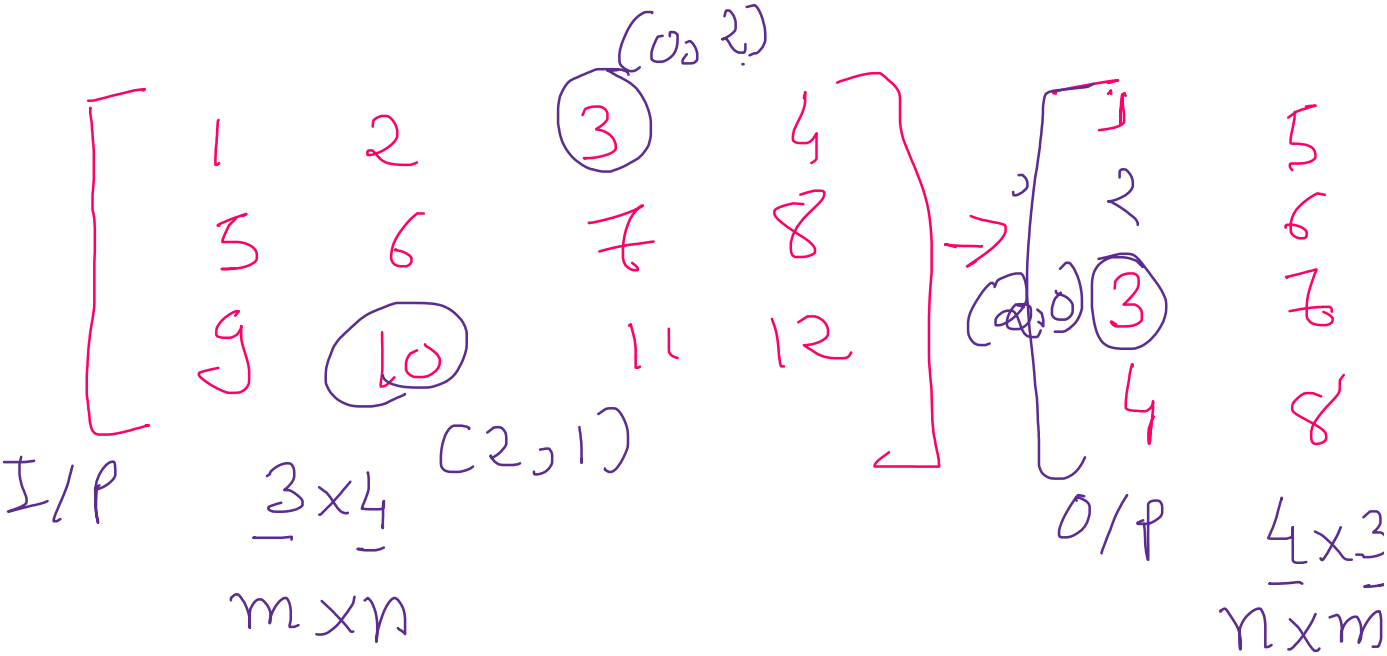
	Col 0	Col 1	Col 2	Col 3
row 0	10	20	30	40
row 1	50	60	70	80

row 2 | 90 ... 100 110 ... 120

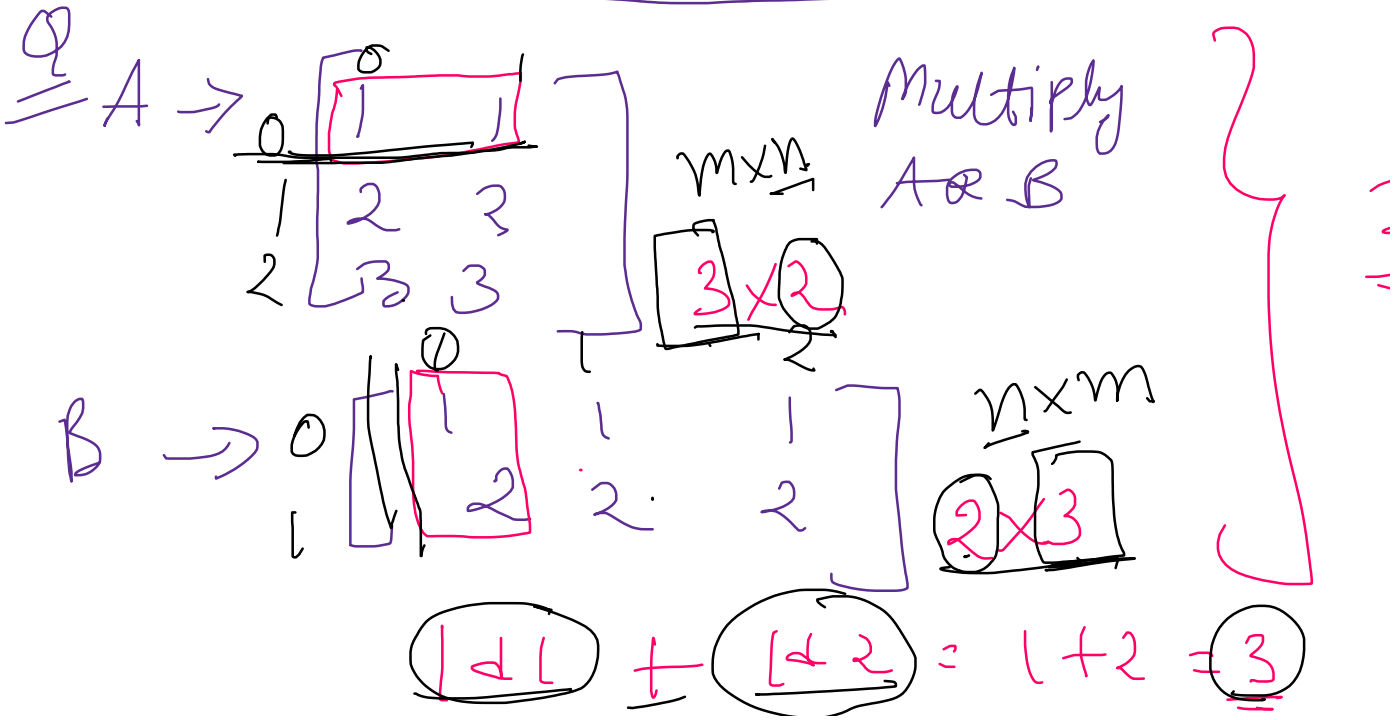
Whenever col → even we are moving

Col → odd we are moving

Q Transpose matrix :- exchanging values with corr. col values :-



Q



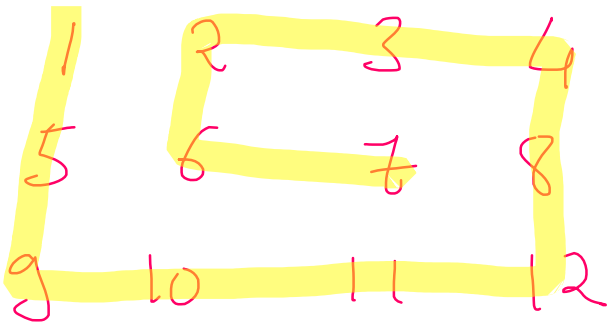
Ol/p

$$\begin{bmatrix} 3 & 3 & 3 \\ 6 & 6 & 6 \\ 9 & 9 & 9 \end{bmatrix}$$

3x3

2

Q Print matrix in Spiral form.



O/P:- 1, 5, 9, 10, 11, 12, 8, 4, 3, 2, 6, 7

colmin

colmax

rowmin

1	2	3	4
5	6	7	8
9	10	11	12

rowmax

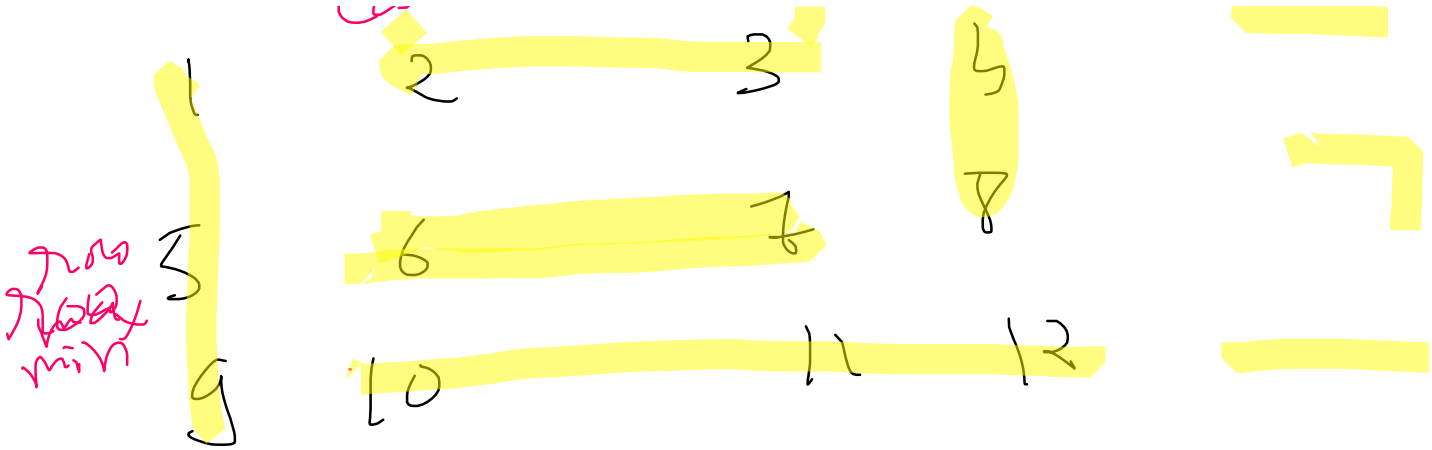
(1) [rowmin → rowmax] → [colmin → colmax]

Print all element rowmin → rowmax

colmin++

colmin

colmax



(2) [colmin → colmax] [rowmax

rowmax --

(3) [row min → rowmax] [colmax

colmax --

(4) [rowmin] [colmax → colmin

rowmin++;

Start at 11:10

[ ]

1D new Array() empty

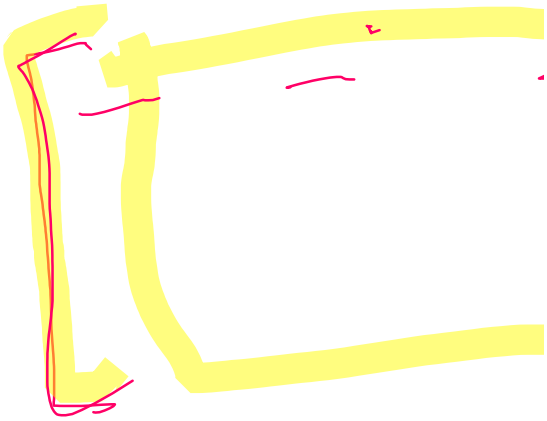
2D } ⇒ custom dimension

rows ? [ ] Array(2)]

Cols]

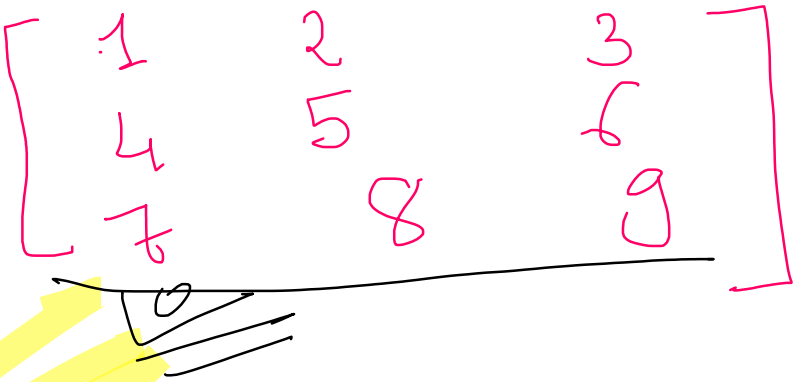
arr [0].

blank array



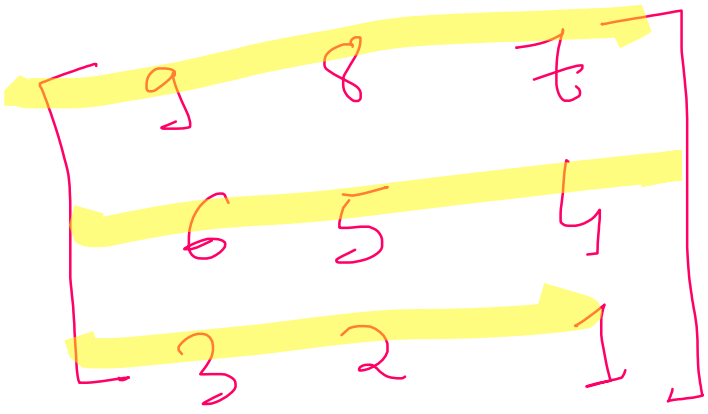
Q.

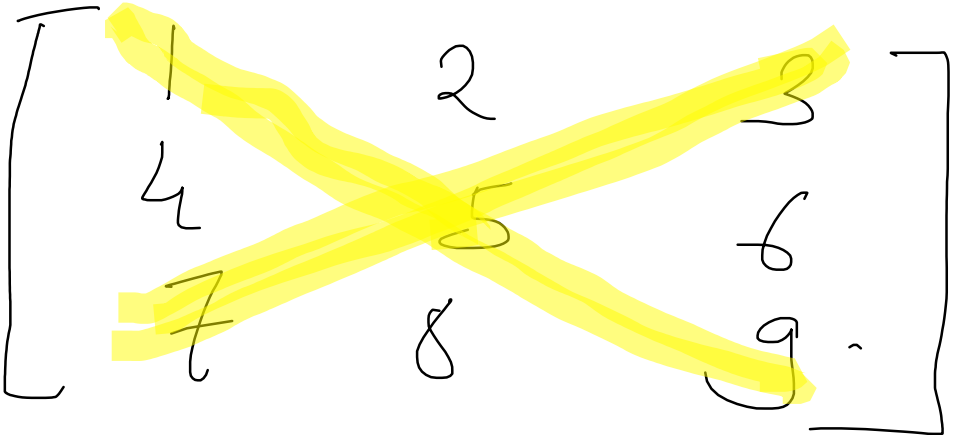
I/P

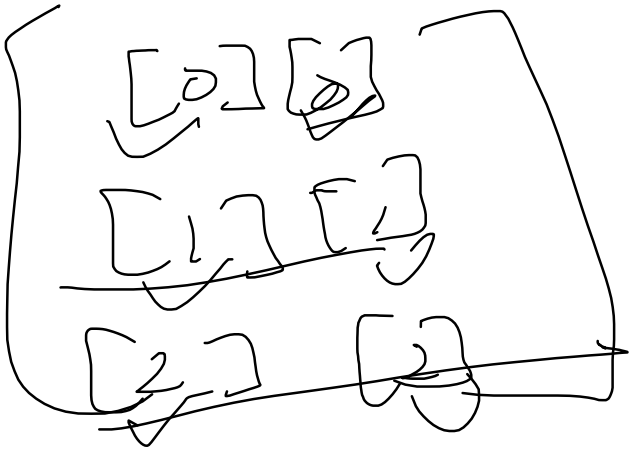


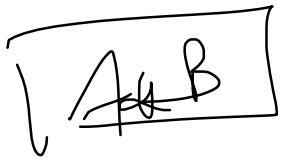
Rotate 180

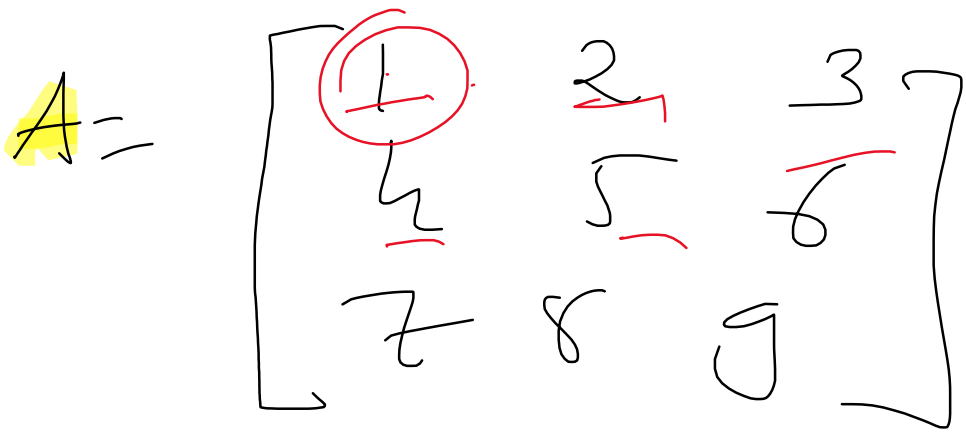
O/P

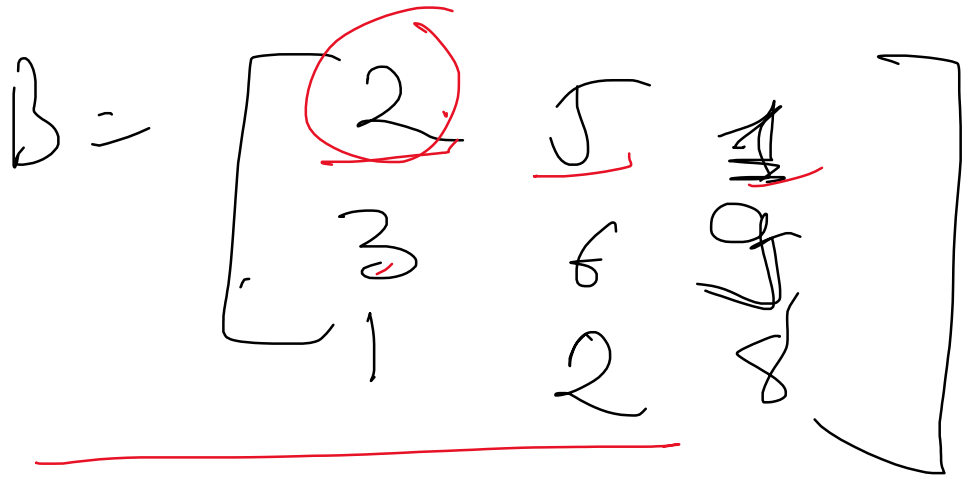


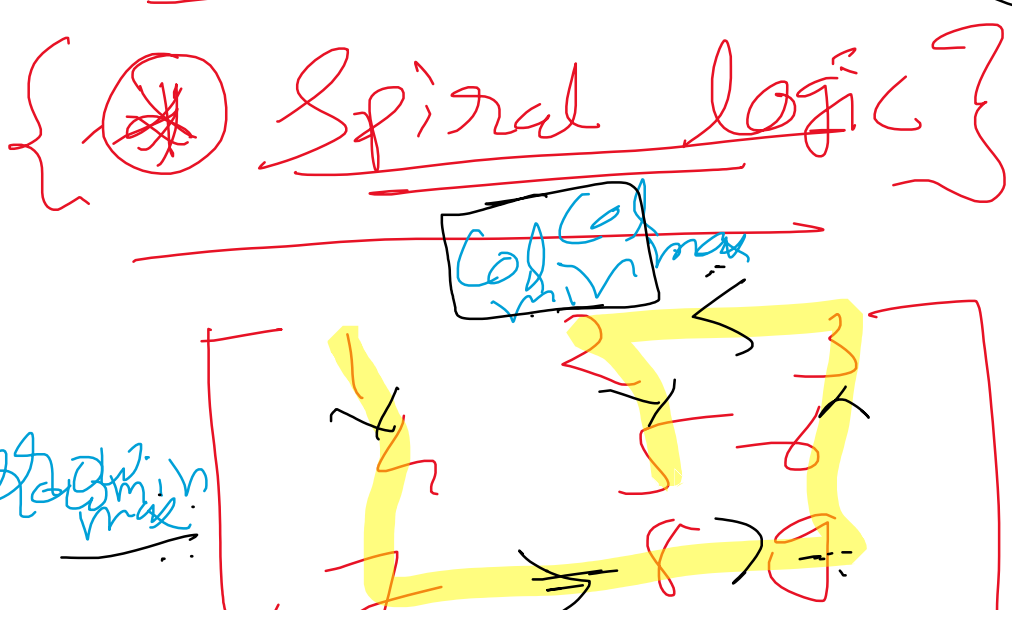












4

2D Array - 2

Q

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

I/P :-

O/P :- Yes / No

Diagonal matrix?

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Q Sum of matrix

1+4+7 = 12
15
18

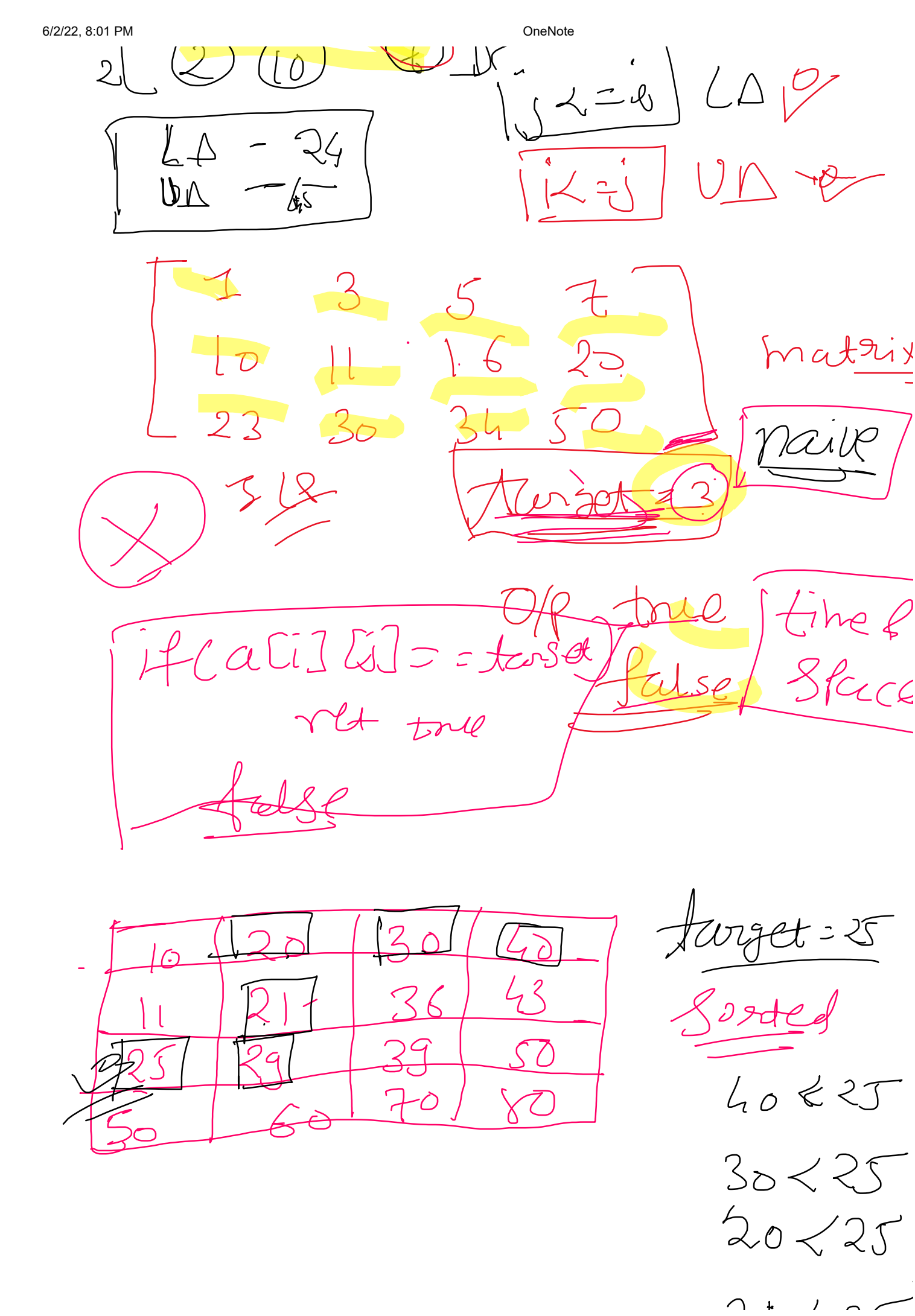
[12, 15, 18]

Q

	0	1	2
0	2	5	4
1	1	2	5

find Sum of lower t

upper t


LA - 24
UA - 45
i j k = 0
LA
UA
i k = j
1 3 5 7
10 11 16 25
23 30 34 50
matrix
pair
Target = 3
O/P true
false
time &
Space
if(a[i][j] == target)
ret true
false
target = 25
Sorted
40 < 25
30 < 25
20 < 25

21 < 25

29 < 25

Q // Print unique elements :-

$$\begin{bmatrix} 11 & 2 & 2 & 20 \\ 5 & 16 & 20 & 7 \\ 1 & 13 & 5 & 16 \\ 6 & 7 & 18 & 15 \end{bmatrix}$$

11 13

1 6

Q =

$$\begin{bmatrix} 2 & 14 & 15 & 18 \\ 10 & 18 & 14 & 22 \end{bmatrix}$$

4 21 LJ

10 14 21 28

O/P 2 28

Starting at ⟹ 8:07

ADVANCED JS

① Pass by Value & Pass by Refer

In Javascript, all function argument

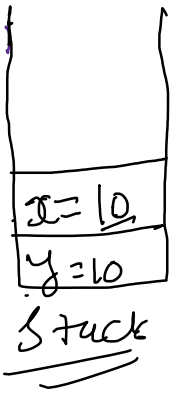
Pass by Value:

↓

copy the value of the variables in the function arguments.

```
y = 10;
Square(y)
```

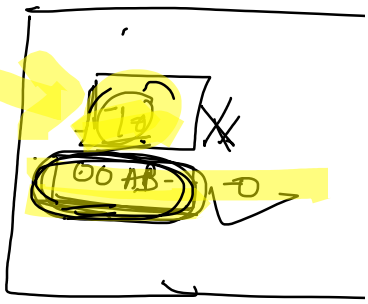
Square(10)




```
function square(x){
  x=x*x;
  return x;
}

let y = 10;

let result = square(y);
console.log(result);
console.log(y);
```

let y = 10 //00AB

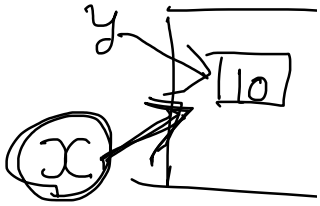
y = 100 //00AB



Square(y) // Pass by valu

Square(10)

e.g. Square(x)



↳ let y = 10;

Square(y)

fn Square(x)

{

}

Square(x)

{

x = x*x; //100

return x;

}

y = 10;
result = square(y) // 100
result

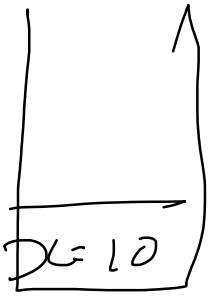
Pass by value as a concept is not supported by all datatype in JS.

Supported
1. Number
2. String
3. Boolean
} Primitive

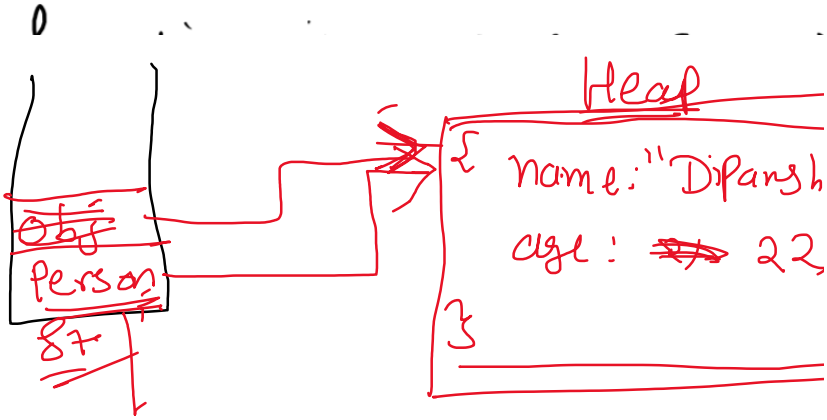
Pass by reference

```
let person = {
    name: "Dipanshu",
    age: 21
}
function increaseAge(obj) {
    obj.age++;
    obj = { name: Pr
}
increaseAge(person)
```

=> DC=10



let person = {
name: "Dipans
age : 21
}

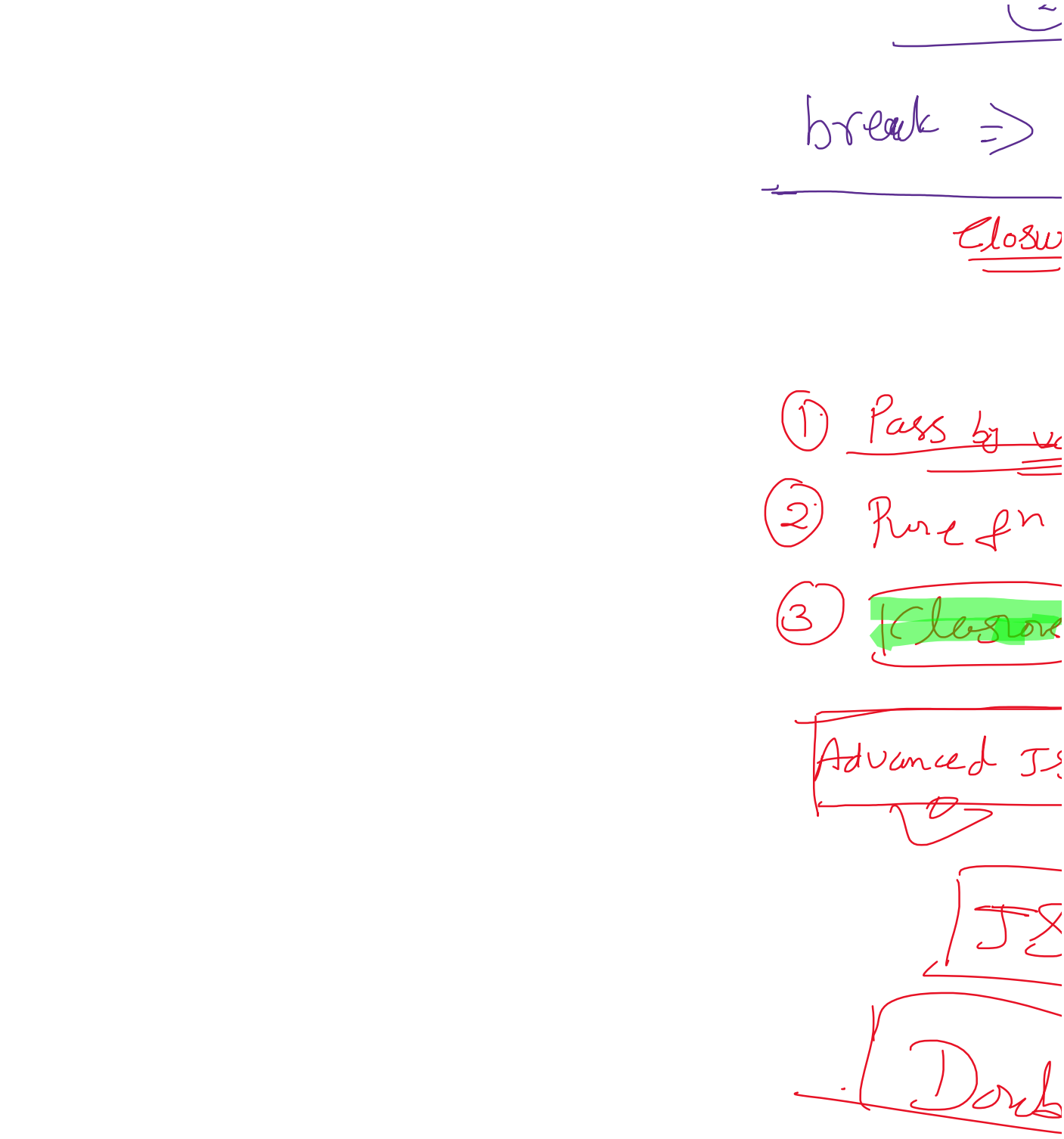



```
let
  nam
  age
}

funct
  obj
  }
  obj
  cor
}

incre
consc
```

Pure fn &

Pure :- (1)
(2) H

Impure- (1)
(2)


break =>
Closu
1 Pass by
2 Pure fn
3 Closure
Advanced JS
JS
Dark