Assignment Problem Class 10_06

10 June 2022 20:01

Q:- arr = [7, 3, 5, 4, 5, 3, 4] } I/P

O/P:- 7

Q:-

i = 0 j = i+1

a[i] a[j]

① naive approach

② XOR

OR || &&

Exclusive OR

XOR - bitwise operators

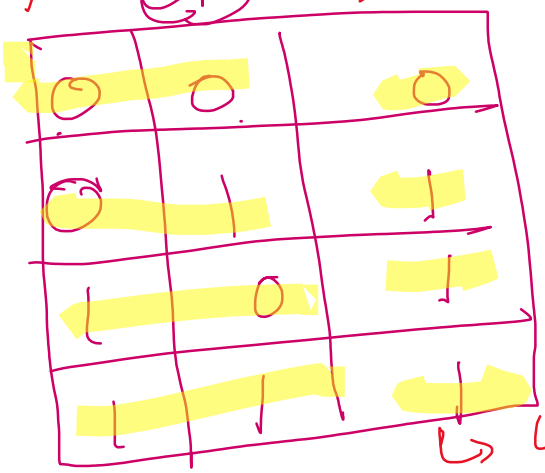
| OR

&

8 →

8 || 2 } →

toggle OR Imp



65

95

1000

0010

1010 (bit OR)

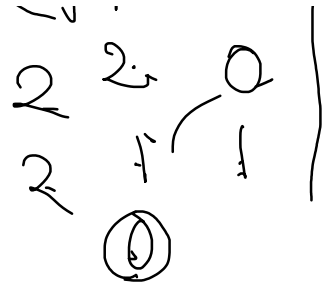
↳ any of I/P 1

↳ ① decimal

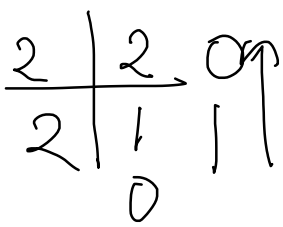
8 || 2:-

2 | 8 0

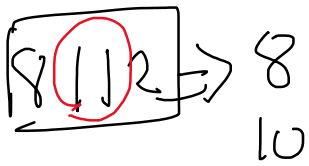
2 | 4 0



9 → 1000
2 → 10



1 0 1 0 (bit)
2^3 2^2 2^1 2^0
8 + 0 + 2 + 0 ⇒ 10



8 | 2 ⇒ 10

a + b → 0 Received
↳ 0 Received
2 + 3 → 3 + 2

a + b
&
b + a

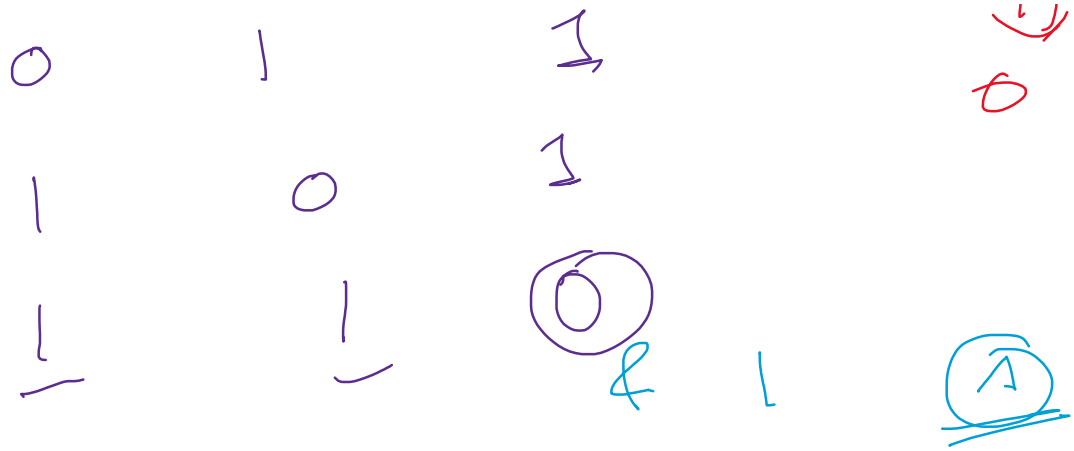
XOR
↳ Commutative → a + b
↳ Associative :-

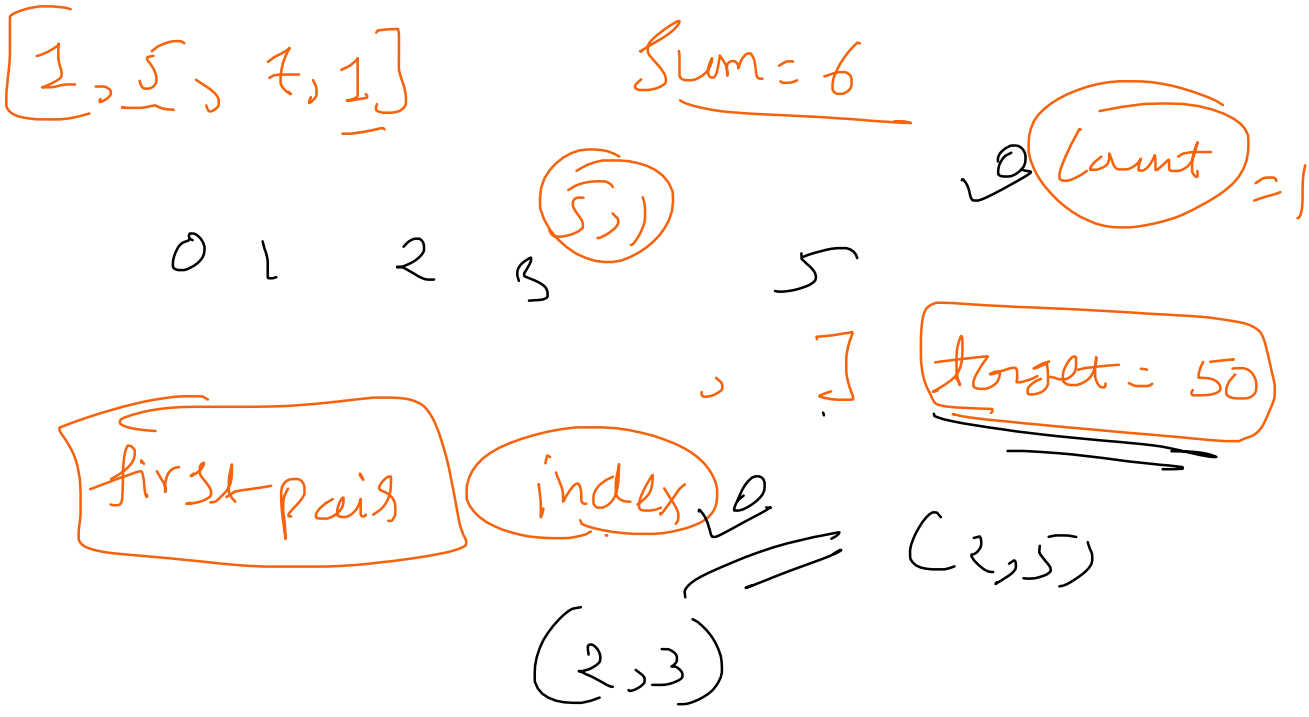
(a + b) + c ⇒ a + (b + c)

(a × b) × c ⇒ a × (b × c)

I/P → 0 0
O/P → 0

both I/P same


0 1 1
1 0 1
1 1 0
& 1
1
5 ^ 3 ^ 4 ^ 5 ^ 2 ^ 3 ^ 4
(5 ^ 5) ^ (3 ^ 3) ^ (4 ^ 4) ^ 2
0 ^ 0 ^ 0 ^ 2
0 ^ 0 ^ 3
0 ^ 2
2


[1, 5, 7, 1]
Sum = 6
count = 1
(5, 1)
0 1 2 3 5
]
target = 50
first pair
index 0
(2, 5)
(2, 3)

b.f. { i=0 { j=i+1 } } { arr[i] + arr[j] == target

a [10, 20, 10, 70, 50, 40] => 50

result

return_result

target - elem

50 - 10 = 40

if (result[10]) {

return_result.push(10)

}

else {

result[4] = [illegible]

}

| 10 | 20 | 10 | 70 | 50 | 40 | => 50

result[target-sum] return_result

[10

10 + x = 50

x = 50 - 10

= [target - curr]

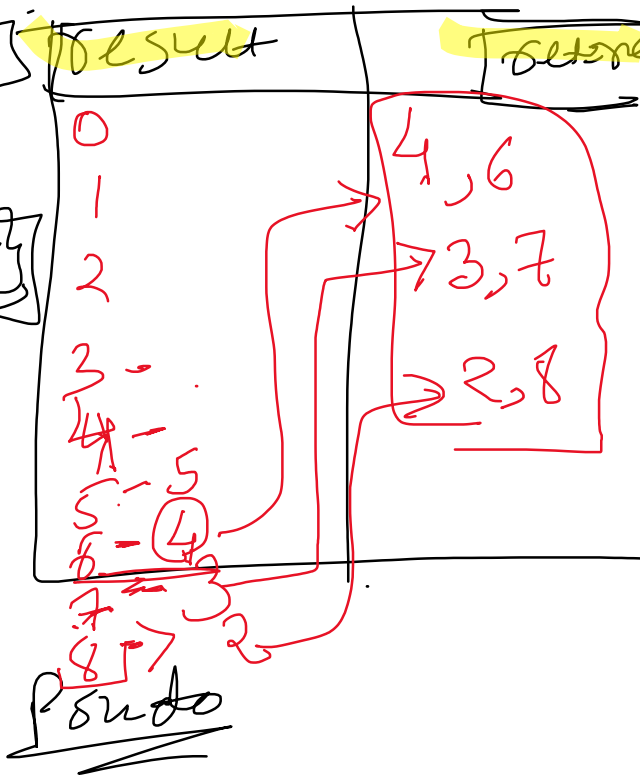
[2, 3, 4, 5, 6, 7, 8] 2,8 4,6 => 10

result = [2 3 4 5]
8 7 6 5

ret_result = [[4,6],[3,7],[2,8]]

result[index]

result[7]



Pseudo

result
ret_result

if(result[curr])
then

ret_result{curr, result[curr

else

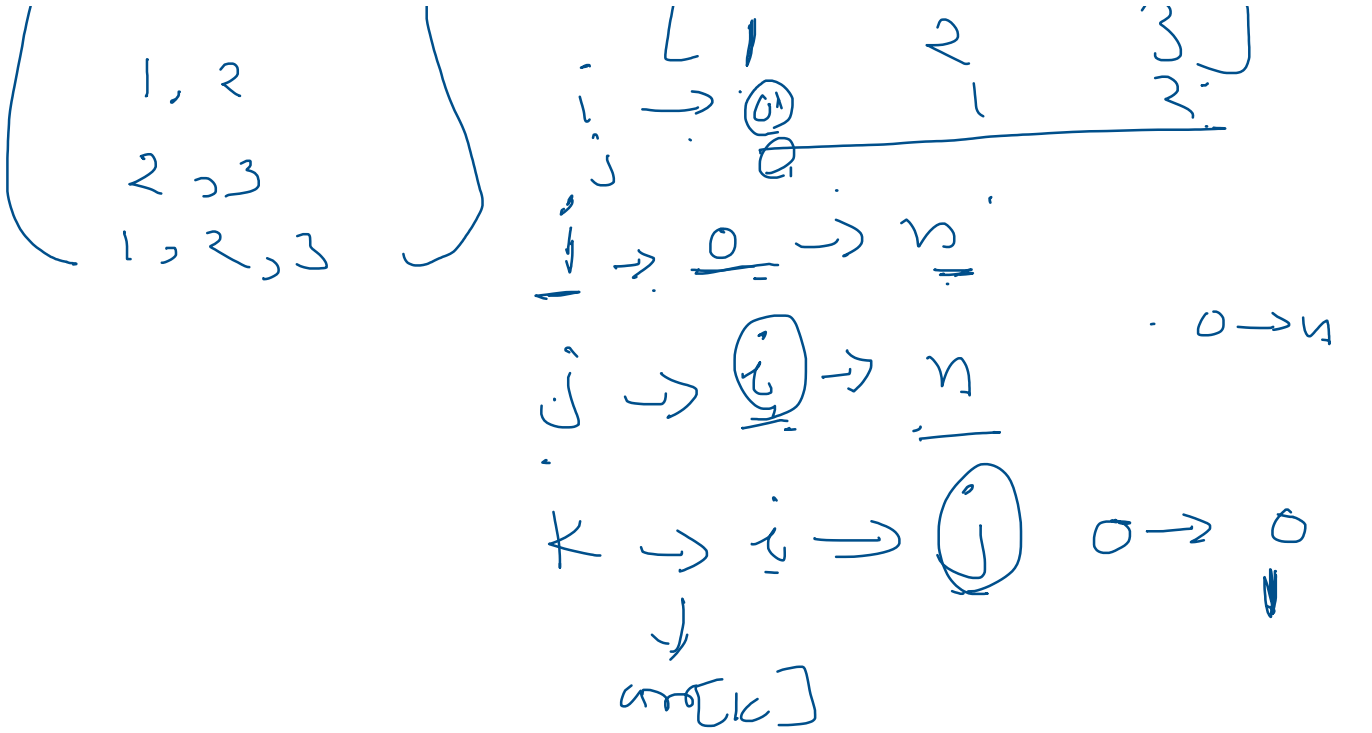
result[sum-curr]=curr

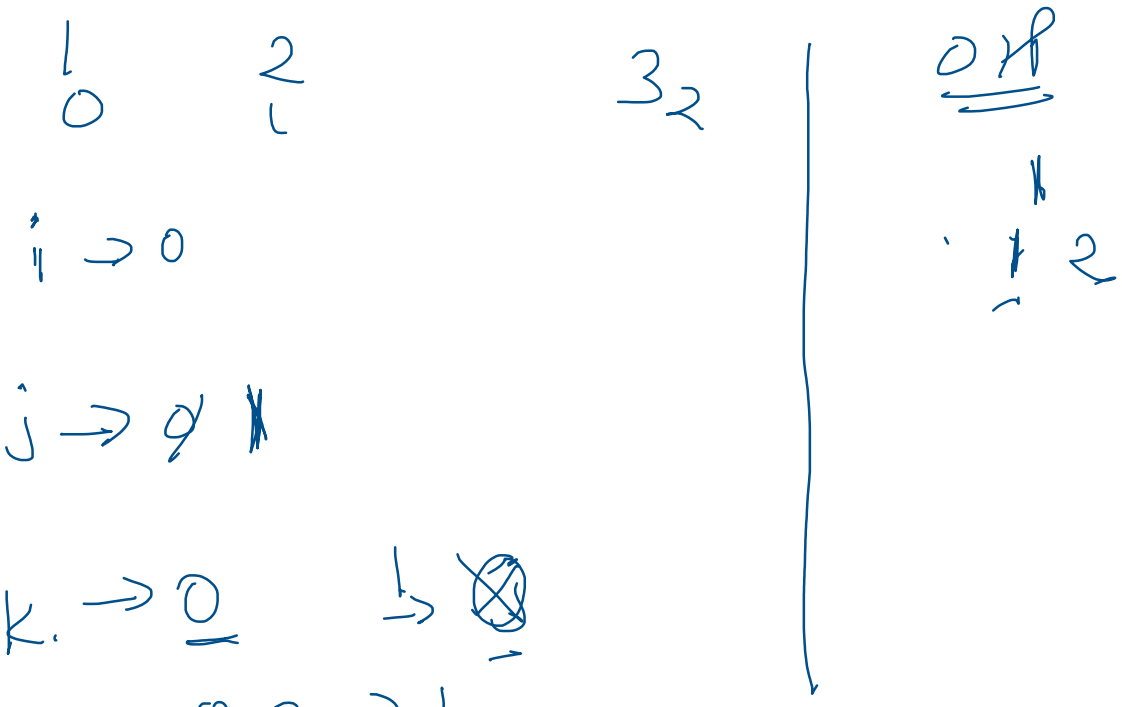
Q

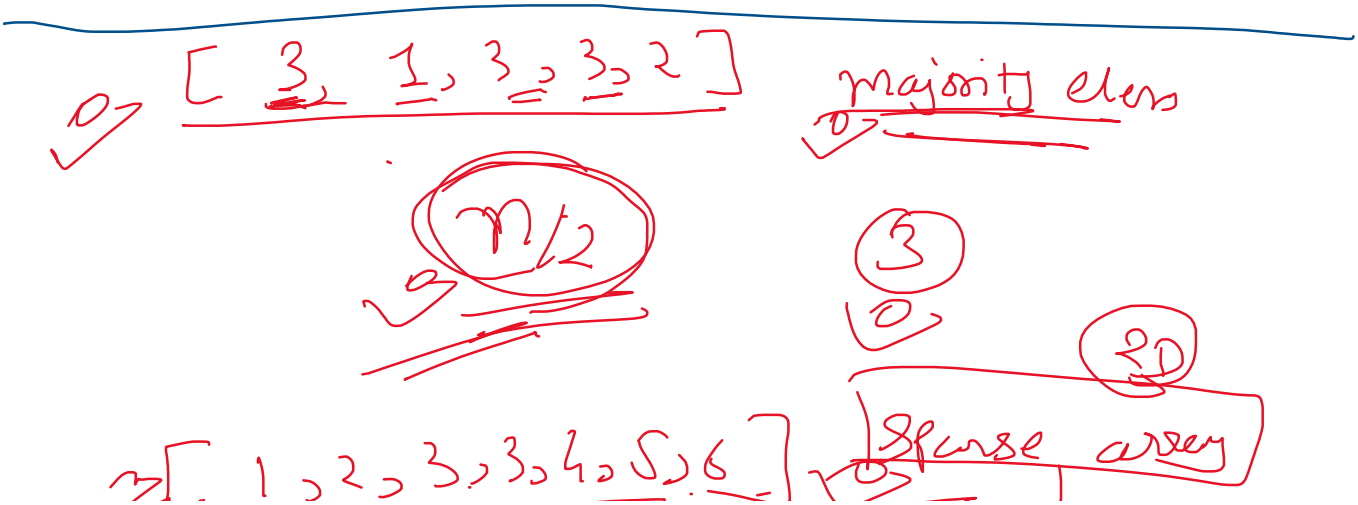
[1,2,3] I/P

1,
2
3

O/P


1, 2
2, 3
1, 2, 3
i → 0 → n
j → i → n
k → i → j
ans[k]


0 1
1 2
2 3
i → 0
j → 0
k → 0 → 1
0, 0 → 1
0, 1 → 2
0 1
1 2


[3, 1, 3, 3, 2]
majority elem
n/2
3
2D
Sparse array
[1, 2, 3, 3, 4, 5, 6]

7

no maj. elem

10/5 > n/2

7/2 => 3

2 counters :-

2 loops :-

count = 1

max count = 0

i -> 0 -> n

j -> 0 -> n

{ arr[i] == arr[j]

count++ }

{ count > max count

update maxcount = count

index = i }

if maxcount > n/2 } => arr[index]

else (not) =>

A [1 2 3 3 4 5 6]

B [3 3 5]

O/P [3 3 5]

=> sorted

Intersection

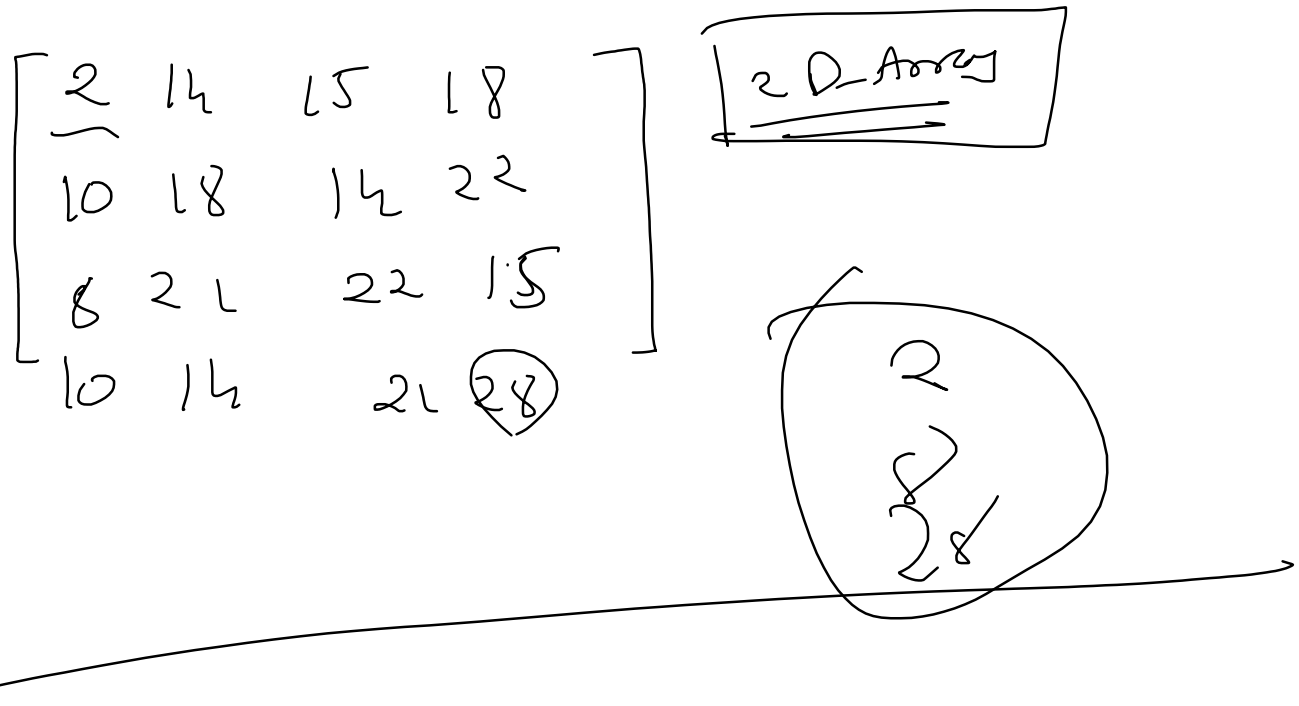
fc < A.length

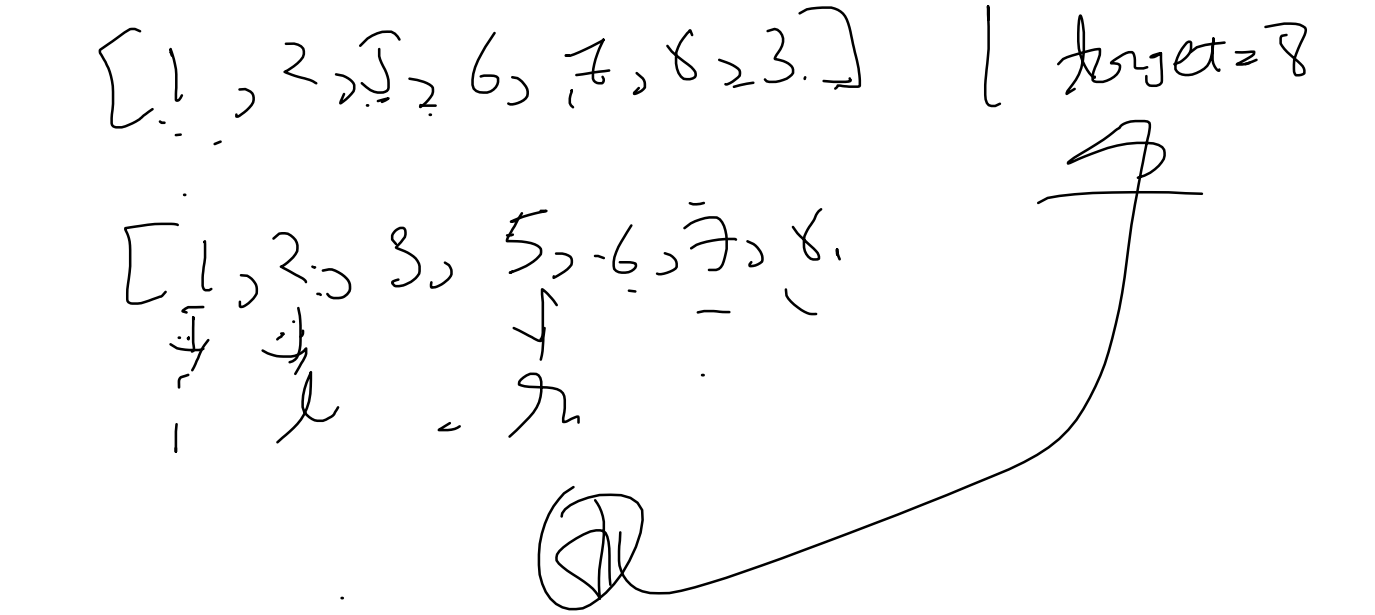
fc < B.length

1 < 3

2 < 3

3 < 3


2D Array


target=8