Problems on hashing

Q.1 Given array of distinct elements, find 4 numbers such that:

$A + B = C + D$, (all are distinct)

eg: [3, 4, 7, 1, 2, 9, 8]

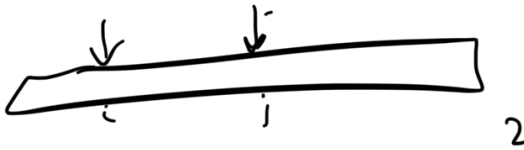
$3 + 7 = 1 + 9$

$4 + 1 = 3 + 2$

(1) Brute force $\Rightarrow$ T.C $\Rightarrow O(N^4)$

```
for( i -> 0-N)
  for( j -> i+1 -> N)
    for( K -> j+1 -> N)
      for(p -> K+1 -> N)
        a[i]+a[j] = a[K]+a[p]
```

or



$A_i + A_j = K$

$A_j = K - A_i$

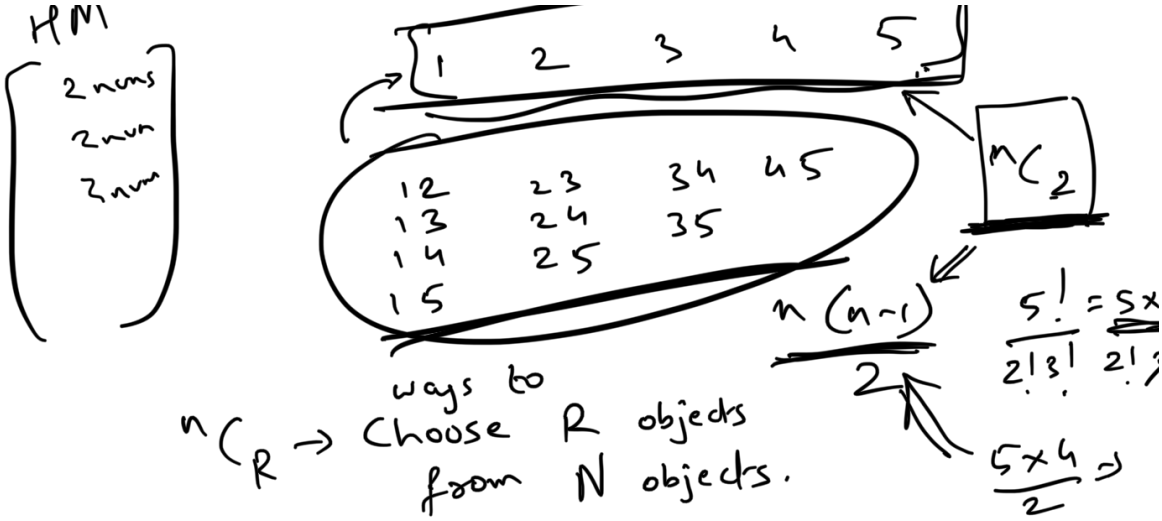
k

$a + b = K$

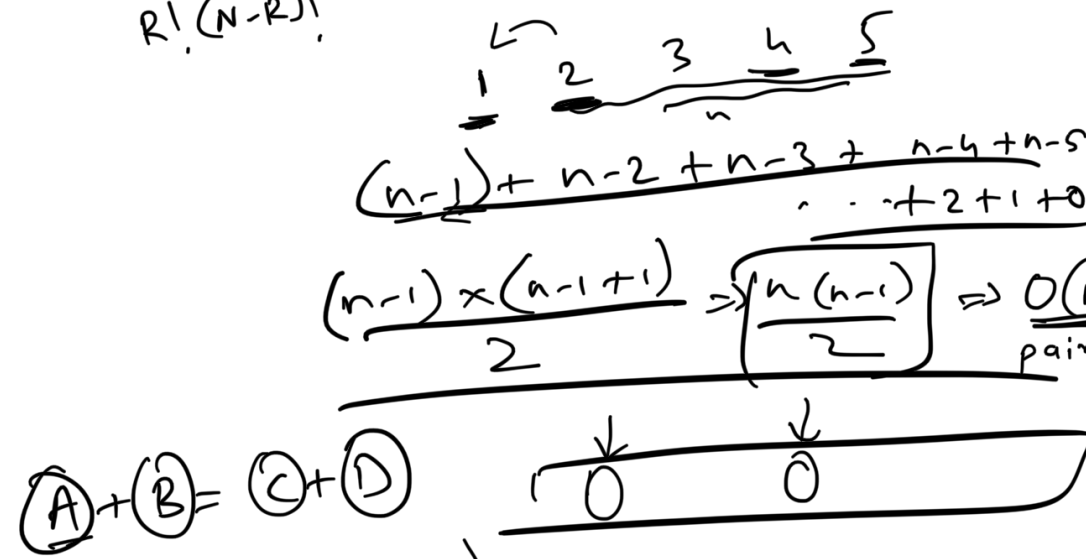
2

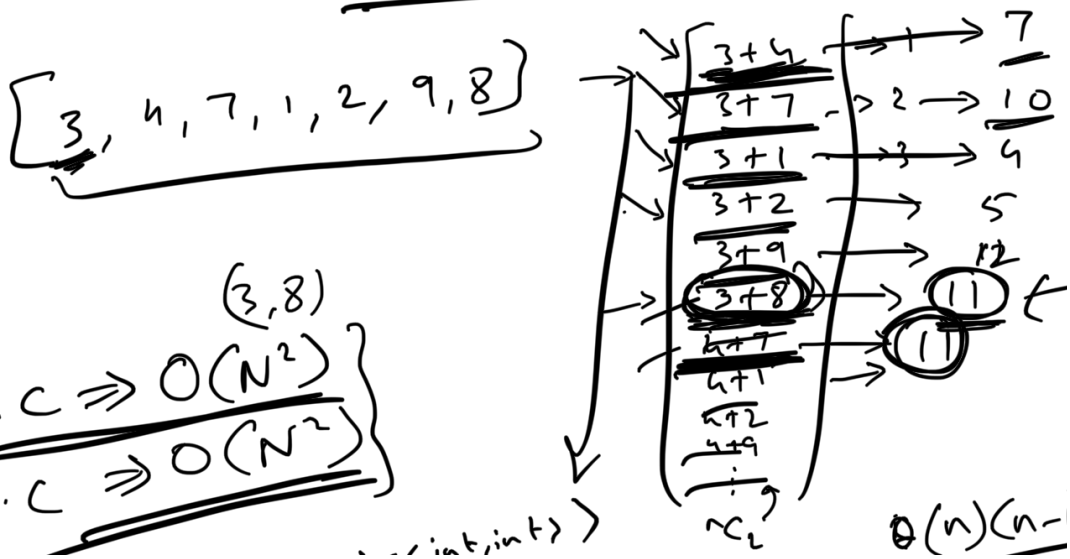
k

(2) We use HM to store something, & use it for finding pairs.

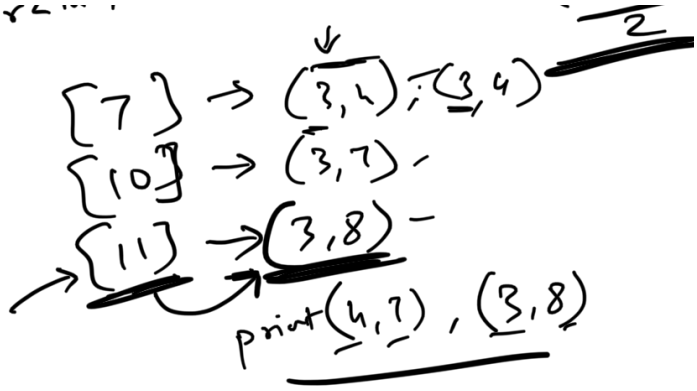

HM
2 nums
2 nums
2 nums
1 2 3 4 5
12 23 34 45
13 24 35
14 25
15
nC2
n(n-1)/2
5!/2!3! = 5x
5x4/2 =>
ways to
nCR → Choose R objects from N objects.

$$\frac{N!}{R!(N-R)!}$$


1 2 3 4 5
n
(n-1)+n-2+n-3+n-4+n-5
...+2+1+0
(n-1)x(n-1+1)/2 => n(n-1)/2 => O(
pair
A + B = C + D
for(i→0→N)
for(j→i+1→N)
HM
[ ]
H.M. insert (A[i]+A[j])


[3, 4, 7, 1, 2, 9, 8]
3+4 → 7
3+7 → 2 → 10
3+1 → 3 → 4
3+2 → 5
3+9 → 12
3+8 → 11
4+7 → 11
4+1
4+2
4+9
nC2
(3,8)
T.C ⇒ O(N²)
S.C ⇒ O(N²)
O(n)(n-

HM< int, pair<...

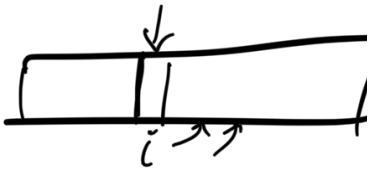


Q.2 Given array, check if subarray exi
Amazon. M.S.
with sum = 0.

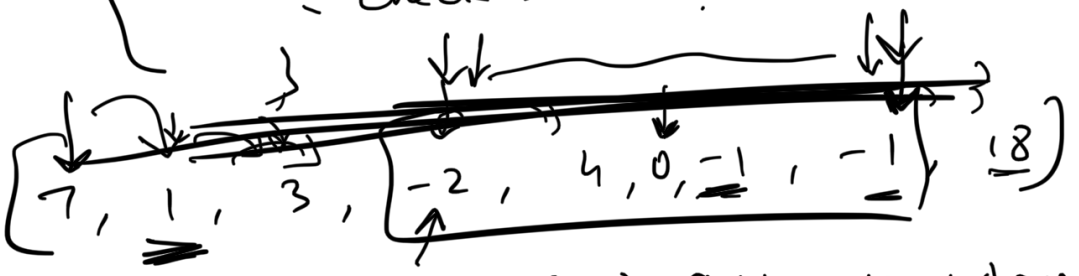
eg. A => [7, 1, 3, -2, 4, -1, -1, 18

Brute → check all subarrays.

S.C => O(1) . T.C. => $O(N^2)$




```
for (i=0 -> N)
    Sum = a[i]
    check(sum==0)
    for(j=i+1 -> N)
    {
        Sum += a[j]
        Check sum==0?
```



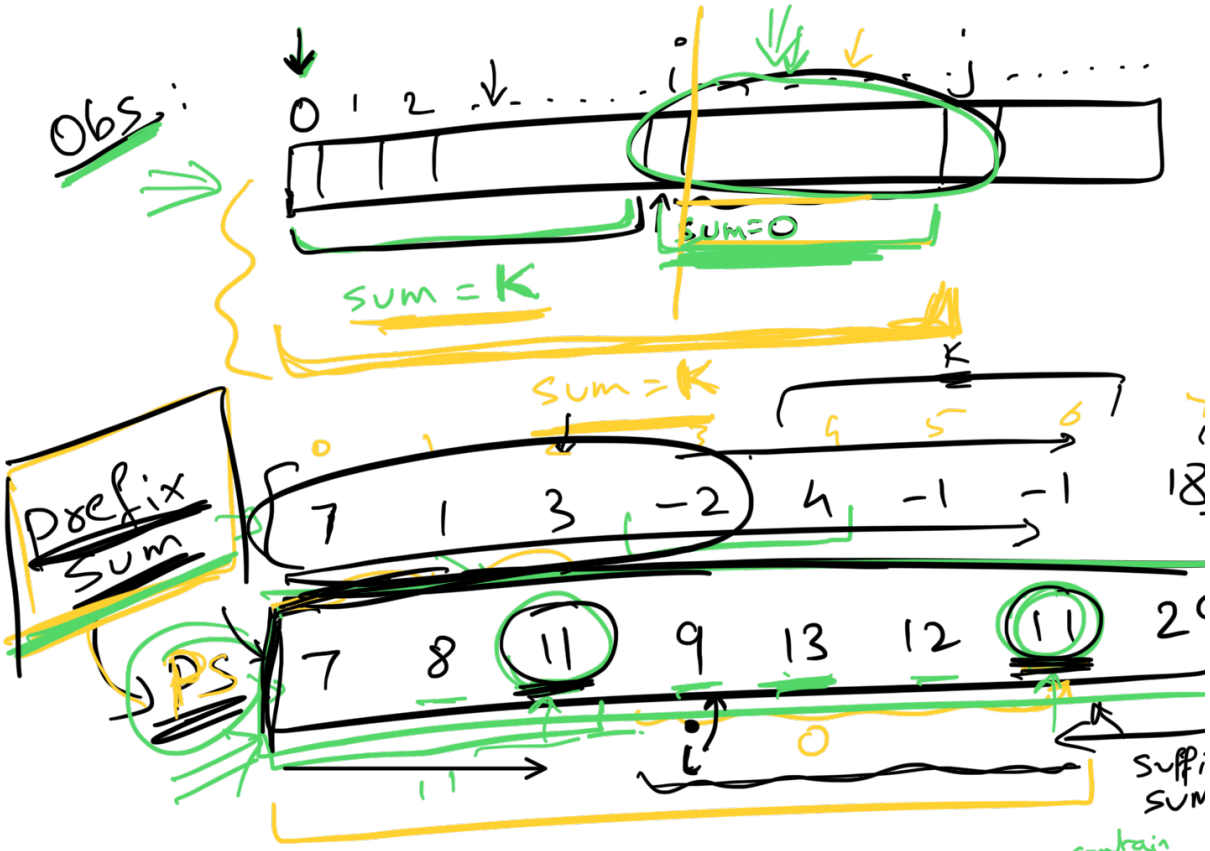
7+1=8+3 = 11 +(-2) = 9+4 = 13+(-1) = 12 +(-1 = 11 +1 => 29

1+3=4+(-2) = 2+4 = 6+(-1) = 5+(-1) = 4+18

3 + (-2) - - - -

-2 + 4 = 2 - 1 = 1 - 1 = 0

2 + 0 = 2

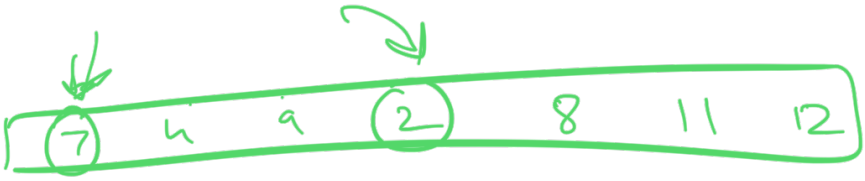


Array. contain

O(N)

T.C ⇒ O(N)

S.C ⇒ O(N)



diff = K :

A + B = K

B = K - A

A - B = K

B = A - K .

2 + 5

2 - 5

K = 5

Q.3 Given an array, count no. of subar
with sum = K

eg. [10, 2, -2, -20, 10]

10 + 2 - 2 - 20 = -10

2 - 2 - 20 + 10 = -10

-20 + 10 = -10

Ans = 3.

K = -

Sub-problem → Sum of [L, R] → O(1).

Real-life.

nag - Kasol → ?

nag → Kheer → ?

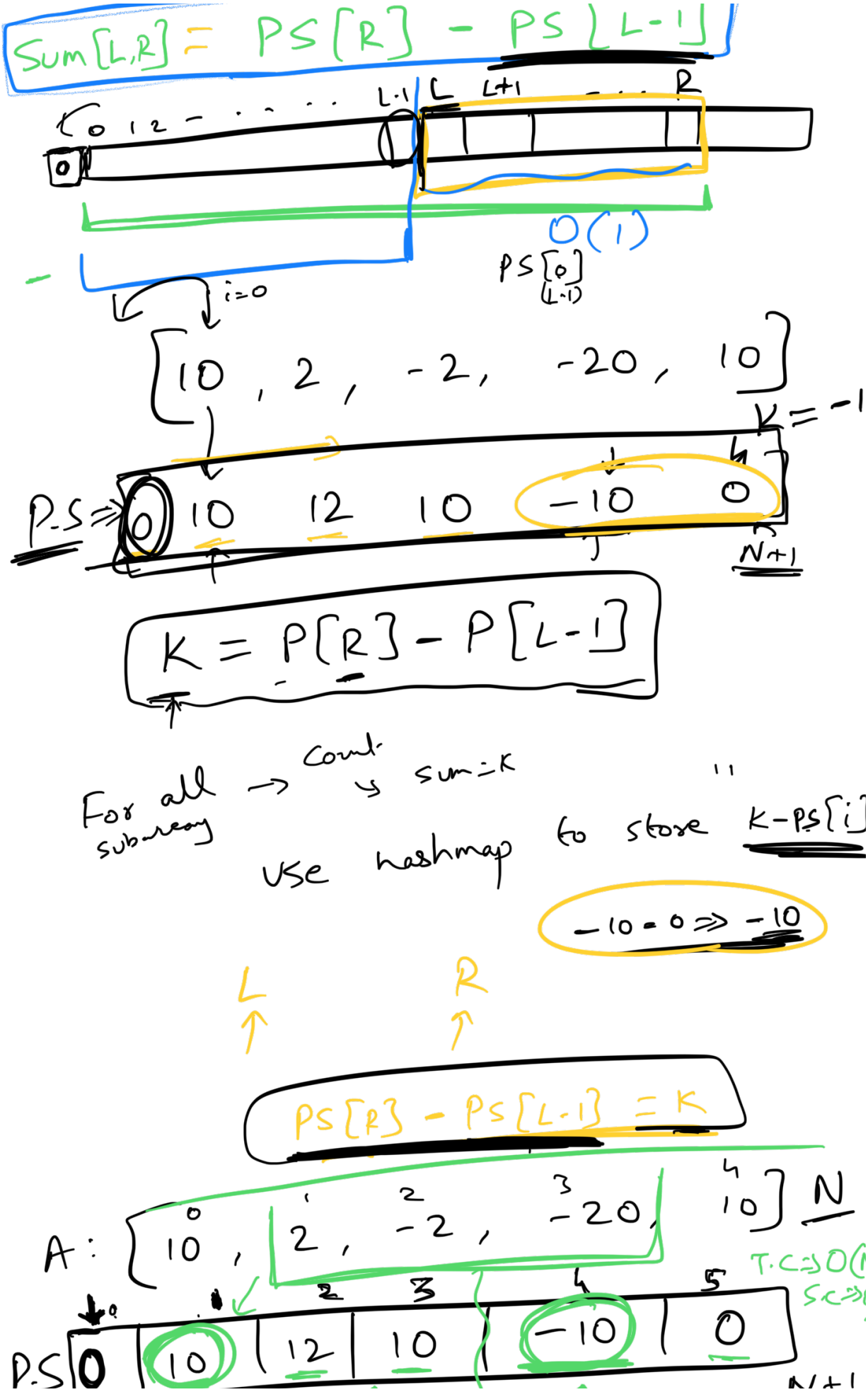
Pune → Kasol → ?

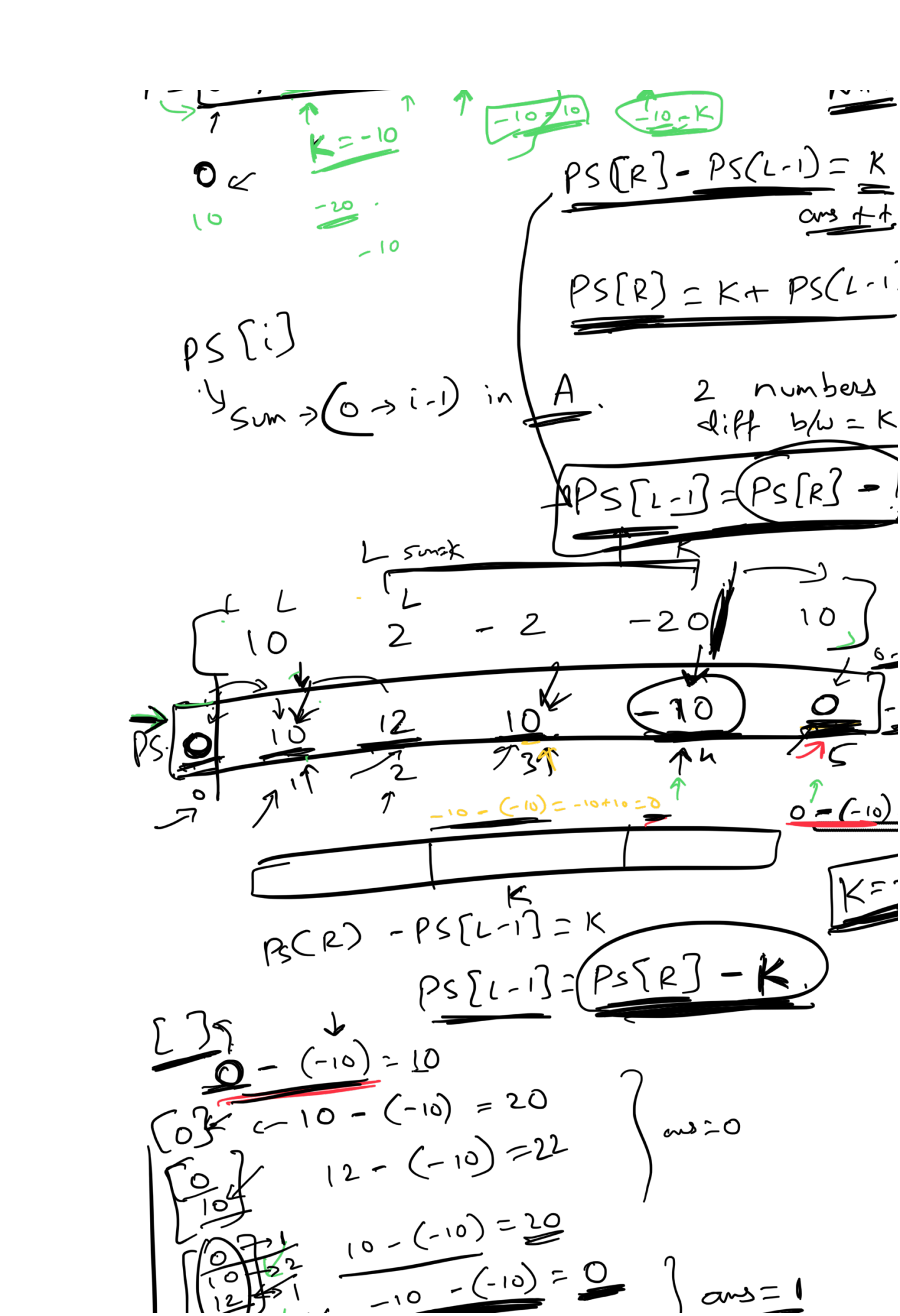
Pune → nag → 20
Pune → Del → 10
Pune → Kasol → 13
Pune → Kher → 135

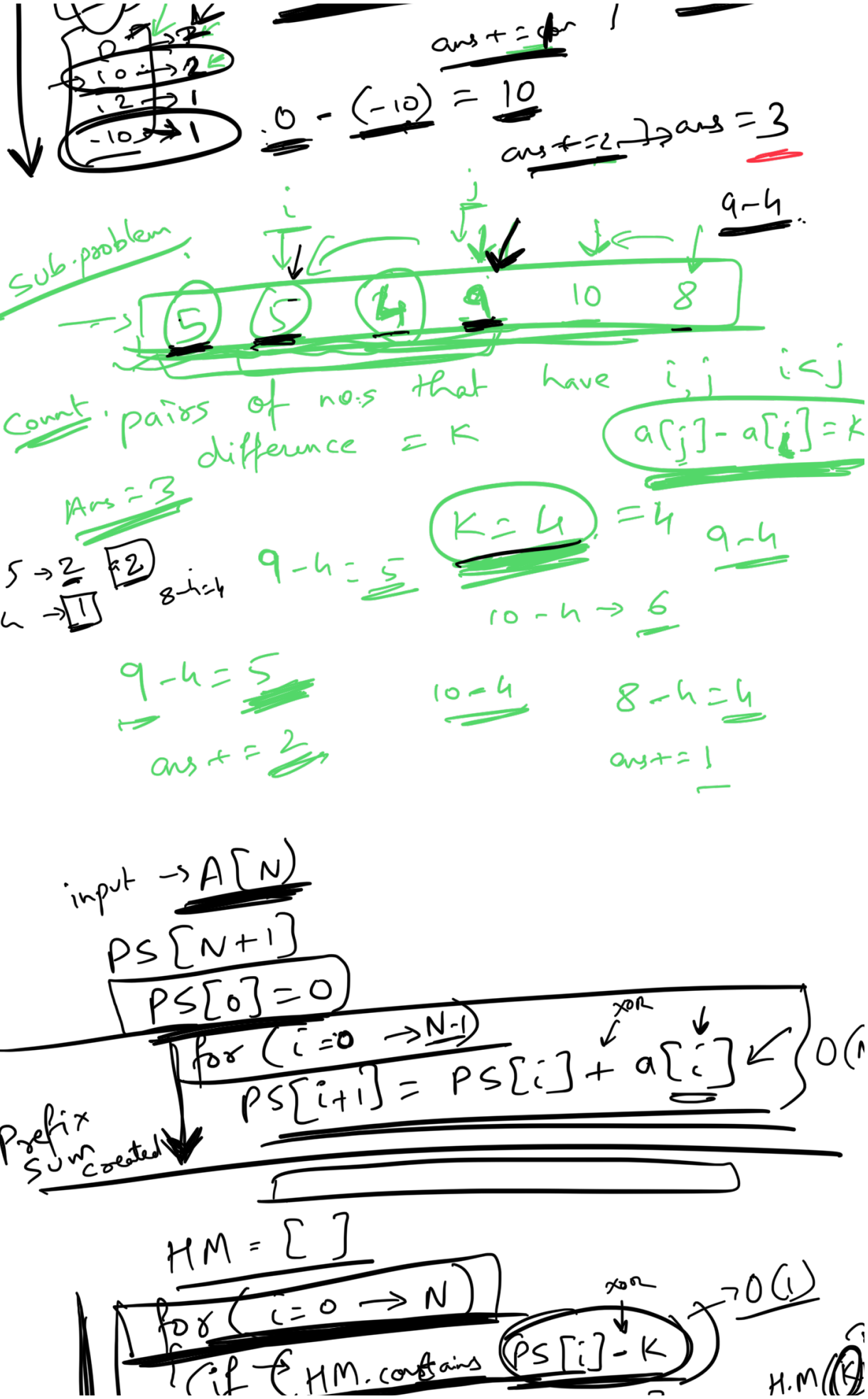
Prefix Sum

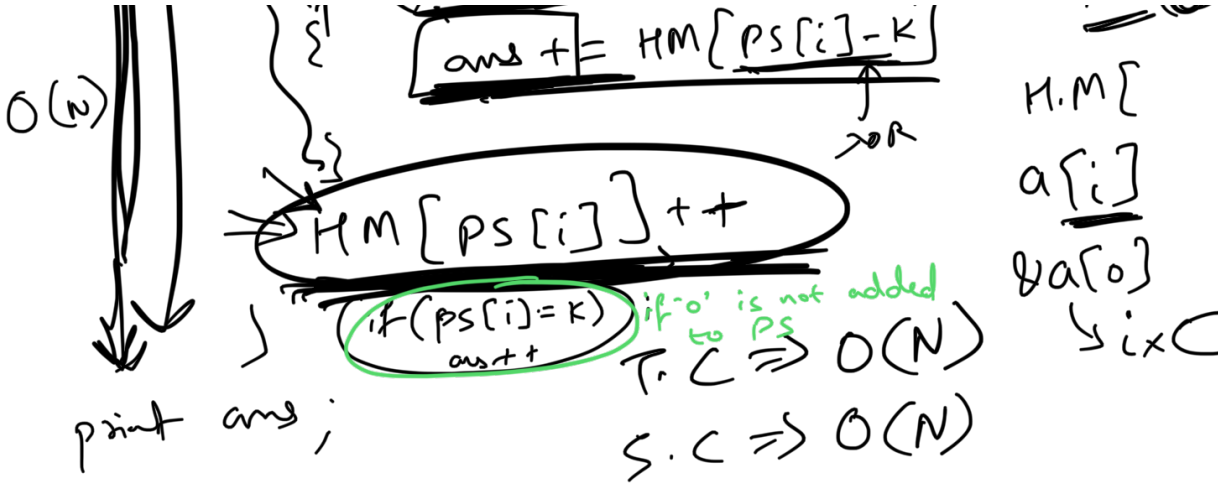
PS [200, 1000, 1300, 1350]

→ O(1)


Sum[L,R] = PS[R] - PS[L-1]
0 1 2 . . . . . L-1 L L+1 . . . R
0
O(1)
PS[0]
(L-1)
i=0
[10, 2, -2, -20, 10]
K = -1
P.S
0 10 12 10 -10 0
N+1
K = P[R] - P[L-1]
For all subarray
Cont. y sum=K
Use hashmap to store K-PS[i]
-10 - 0 => -10
L
R
PS[R] - PS[L-1] = K
A: [10, 2, -2, -20, 10] N
0 1 2 3 4
T.C=>O(
S.C=>
P.S 0 10 12 10 -10 0
N+1


K = -10
-10 = 10
-10 = K
PS[R] - PS[L-1] = K
ans ++
PS[R] = K + PS[L-1]
PS[i]
sum → (0 → i-1) in A.
2 numbers
diff b/w = K
PS[L-1] = PS[R] -
L sum K
10
2
-2
-20
10
PS:
10
12
10
-10
0
-10 - (-10) = -10+10 = 0
0 - (-10)
K
PS[R] - PS[L-1] = K
PS[L-1] = PS[R] - K.
0 - (-10) = 10
-10 - (-10) = 20
12 - (-10) = 22
ans = 0
10 - (-10) = 20
-10 - (-10) = 0
ans = 1


0 → 2
10 → 2
12 → 1
-10 → 1
0 - (-10) = 10
ans += 1
ans += 2 → ans = 3
Sub-problem
i
j
9-4
5 5 4 9 10 8
Count pairs of nos that have i, j i<j
difference = K
a[j] - a[i] = K
Ans = 3
5 → 2
4 → 1
2
8-4=4
9-4 = 5
K = 4
=4
9-4
10-4 → 6
9-4=5
10-4
8-4=4
ans += 2
ans += 1
input → A[N]
PS[N+1]
PS[0] = 0
for (i=0 → N-1)
PS[i+1] = PS[i] + a[i]
xor
O(
Prefix sum created
HM = [ ]
for (i = 0 → N)
if (HM.contains
PS[i] - K
xor
→ O(1)
H.M



PS ⇒ [5 8 3 6 11 9]

8 - 3 ⇒ 5

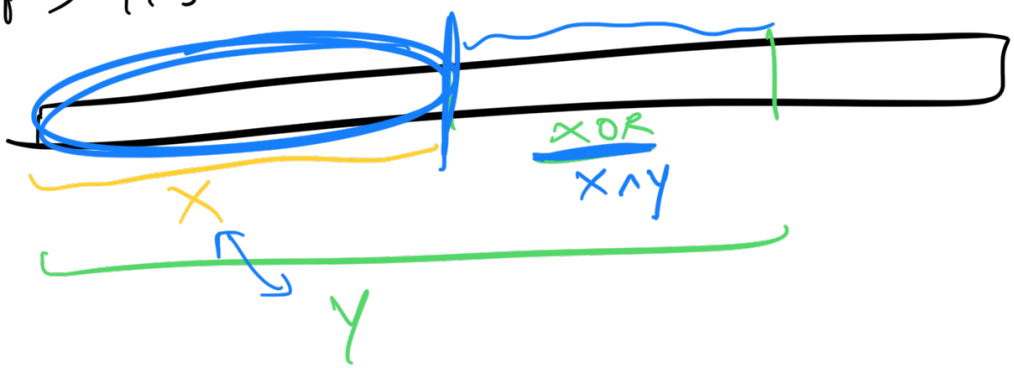
K = 3.

Q.4 Given array, count no. of subarr
where XOR = K. K.

Instead
prefix sum → prefix XOR

PS[i] = a[0] ∧ a[1] ∧ a[2] ··· a[i]

PS[i] = a[0] + a[1] + a[2] ··· a[i]



$PX[L \to R] = PX[R] \wedge PX[L-1]$

Q.5 Given array, count no. of sub-arrays with exactly K odd nos.

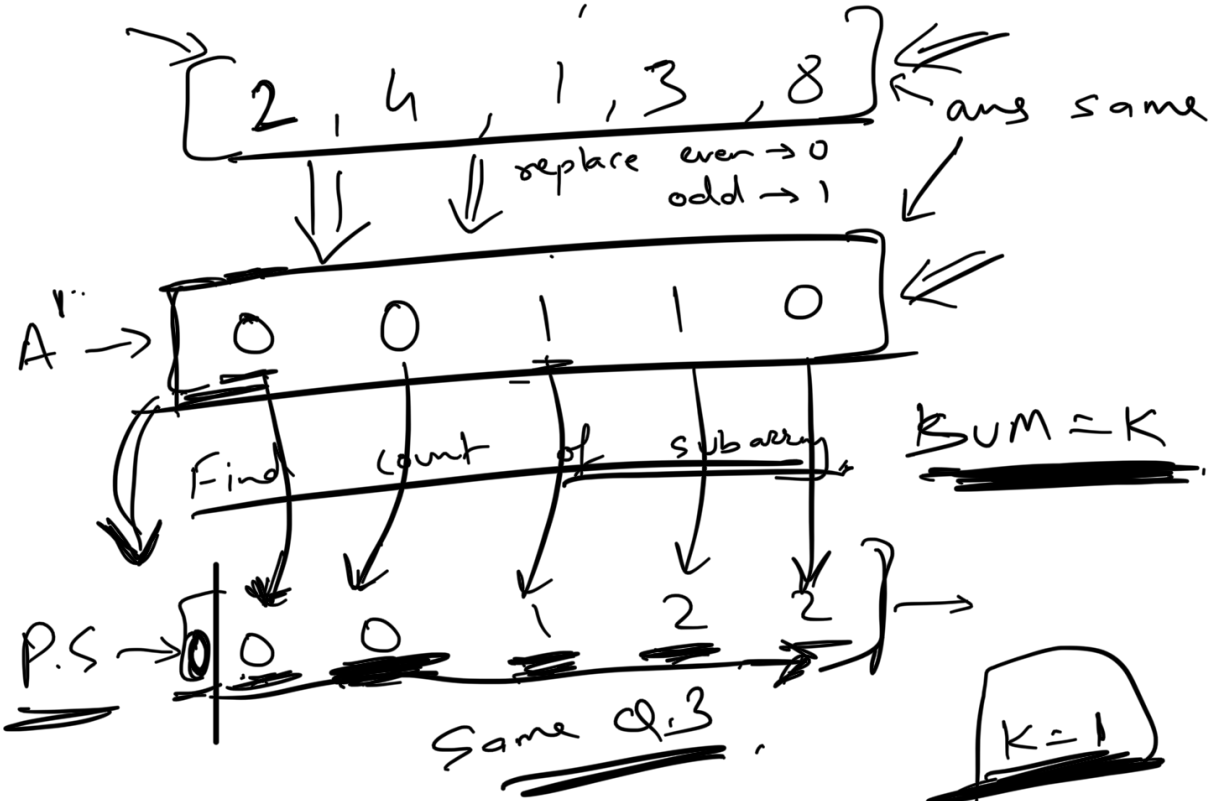
eg. [2, 4, 1, 3, 8] K = 1

Ans = 5

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

Ans = 6 K = 2

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



-1

0 → 0-K? ans = 0

0 → 0-1 ans = 0

1 → 1-1 ⇒ 0 ans += 3 ⇒ 3

2 → 2-1 ⇒ 1 ans += 1 ⇒ 4

2 → 2-1 ⇒ 1 ans += 1 ⇒ 5

A [0 0 1 1 0]

ps [0 0 0 1 2 2]

K = 2

[ ]

0 → 0-2 ⇒ -2 ans = 0

[0 → 3]

-2 ⇒ -1

1 → 1

2 → 2

2-2 ⇒ 0

2-2 ⇒ 0

ans = 0

ans += 3 = 3

ans += 3 = 6.

Q.6 Given array of positive integers, find length of longest consecutive sequence which can be formed using no. from array.

Amazon
Google

eg. [9, 3, 10, 4, 20, 2]

Ans = 3 → (2, 3, 4)

(9, 10)

① Brute → sort → find by iterating

T.C ⇒ O(n log N + N)

X (2, 3, 4, 9, 10, 20) →

② → optimise → O(N).

Doubts

abc

bca