Accenture Coding Round Practice Problems

— Shaurya Awasthi

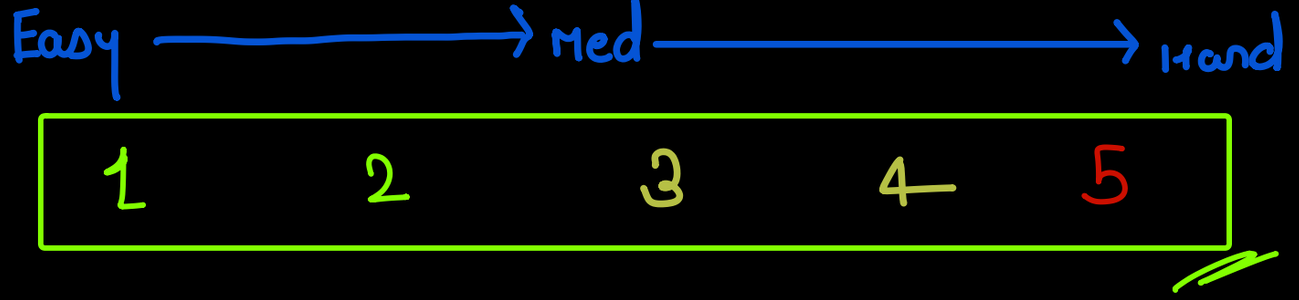
Instructions :-

- Program should take input from standard input and print output to standard output.
- Your code is judged by an automated system, do not write any additional welcome/greeting messages.

"Save and Test" only checks for basic test cases, more rigorous cases will be used to judge your code while scoring.

Additional score will be given for writing optimized code both in terms of memory and execution time

⇒ Level Rating



Q-1 Sum of odd Integers (L-1)

Problem statement

An odd number is an integer which is not a multiple of 2.

You are required to implement the following function:

Int SumOddIntegers(int arr[], int n);

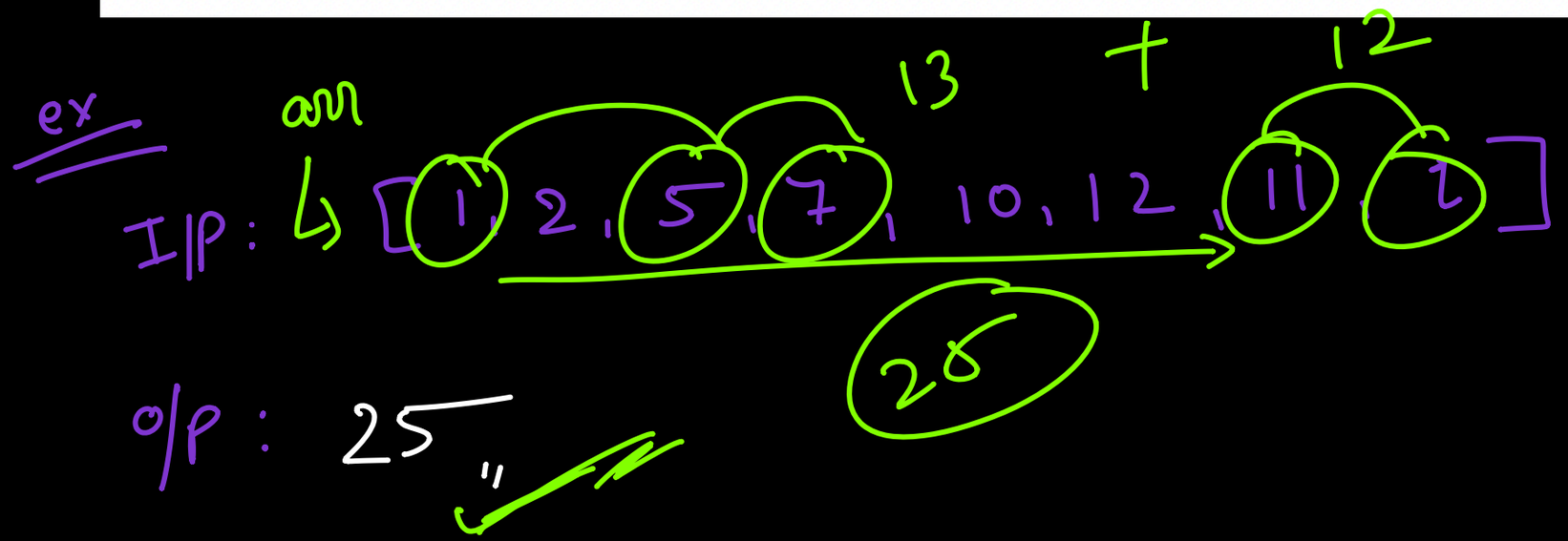
The function accepts an integer array 'arr' of length 'n' as its argument. You are required to calculate the sum of all odd integers in an array 'an' and return the same.

Note:

Array can have negative integers

n > 0

Computed values lie within integer range



Q-2 : Single Digit (L-1) (1st problem of a set - 10 sept 2024).

- Given a number N your task is to make N a single digit number by performing the following operations:
  - If N is ODD make it floor (N/2).
  - If N is even make it floor ((N-2)/2).
  - If N is already a single digit print it as it is.

Example -

I/P → 10

O/P → 4

Example -

I/P → 12

O/P → 5

239

↓

ex n = 10

```
while (n > 9)
{
    (10-2)/2  => 8/2 = 4
}
Ans = 4
```

Q-3 Superior Array Element (L-2)

Problem statement

In an array a superior element is one which is greater than all elements to its right. The rightmost element will always be considered as a superior element.

You are given a function,

Int FindNumberOfSuperiorElements(int* arr, int n);

The function accepts an integer array 'arr' and its length 'n'. Implement the function to find and return the number of superior elements in array 'arr'.

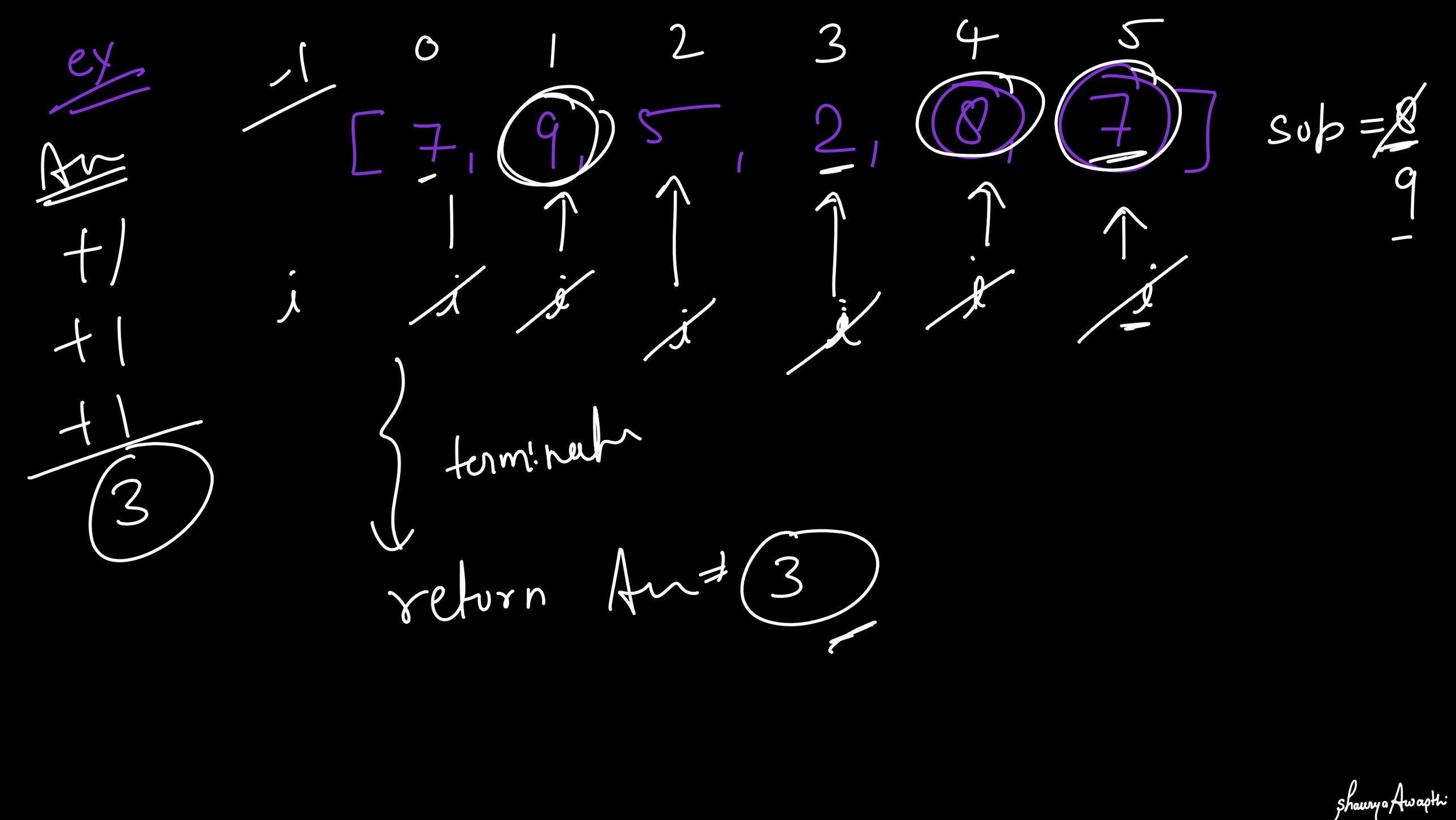
Assumptions:

1. N > 0.
2. Array index starts from 0.

ex

I/p : Arr = [7 9 5 2 0 7]

O/p : 3 ,



Q-4 Edward's Birthday (L-3)

Problem statement

It is Edward's birthday today. His friends have bought him a huge circular cake.
Edward wants to find out the maximum number of pieces he can get by making exactly N straight vertical cuts on the cake.
Your task is to write a function that returns the maximum number of pieces that can be obtained by making N number of cuts.

Note: Since the answer can be quite large, modulo it by 1000000007

Input Specification:

input1: An integer N denoting the number of cuts

Output Specification:

Return the maximum number of pieces that can be obtained by making N cuts on the cake

ex
I/P => 4 (n) cuts
O/P => 11

ex
I/P - 1
O/P - 2

Logic

{ if a chord of circle cuts n other chords then it will introduce (n+1) new parts.

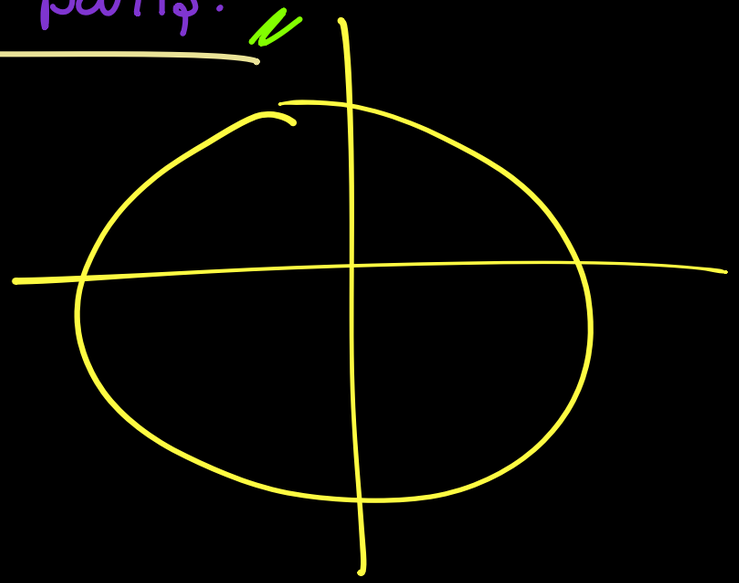
n = 5

1 + 1 + 2 + 3 + 4 + 5

+1

||

16



Ans = sum of n no. + 1

$$= \frac{n(n+1)}{2} + 1$$

Q-5 Regions on a plane (L-3) (Same as Prob-4)

Problem statement

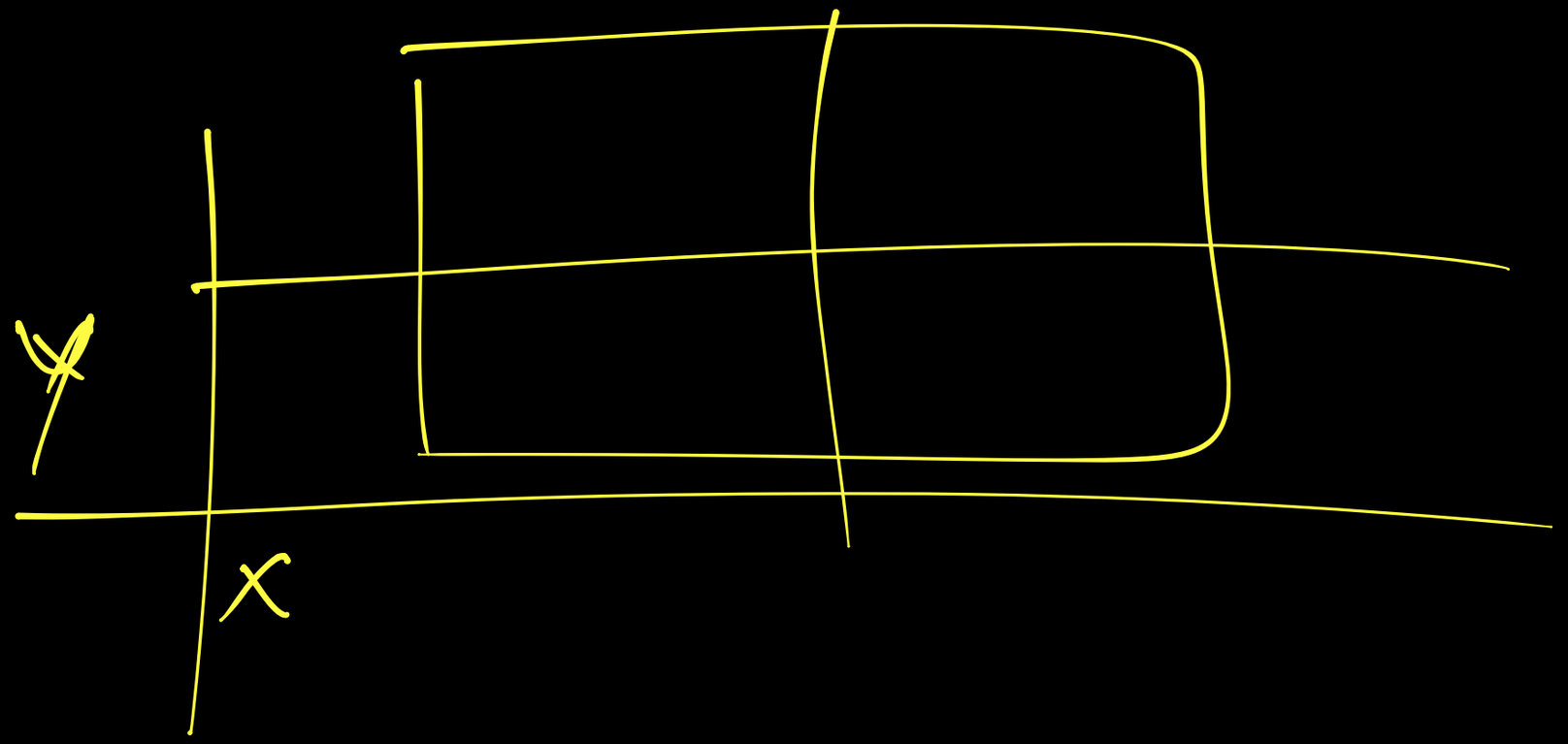
Mr. Professor is a great scientist, but he is not able to find a solution to one problem. There are N straight lines that are not parallel, and no three lines go through the same point. The lines divide the plane into M regions. Write a function to find out the maximum number of such regions he can get on the plane.

Input Specification:

input1: An integer N representing the number of straight lines (0 <=N<= 100)

Output Specification:

Return the maximum number of regions



Q-6 Inversion Count (L-4)

Problem statement

Let j and k be two indices in an array A.

If j < K and A[j] > A[k], then the pair (j,k) is known as an "Inversion pair".

You are required to implement the following function:

int InversionCount(int *A, int n);

The function accepts an array 'A' of 'n' unique integers as its argument. You are required to calculate the number of 'Inversion pair' in an array A, and return.

Note:

If 'A' is NULL (None in case of python), return -1

If 'n' <2, return 0

example

I/P: n=5 , A = [1, 20, 6, 4, 5]

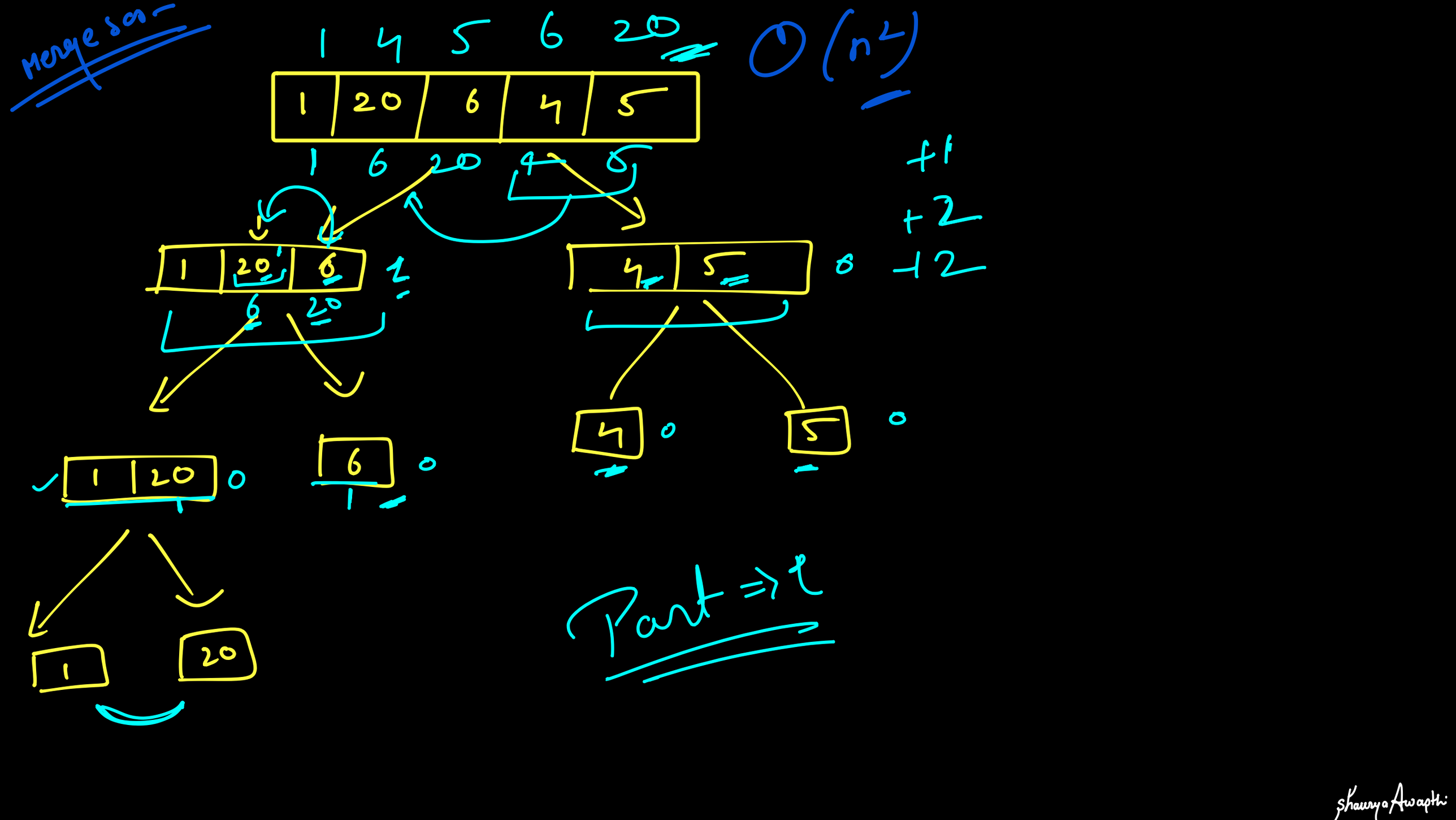
0 1 2 3 4

O/P: 5

(20,6) (20,5) (20,4) (6,4) (6,5)

1 2 3 4 5

Merge sort -



Part ⇒ 1