Q1 You have an array of size N. All the elements are either zero or one. We need to arrange the elements such that all the zeroes are exactly to the left of all the ones.

→ [0, 1, 1, 0, 0, 1, 0, 0, 1]

$N \le 10^7$

→ [0 0 0 0 0 1 1 1 1]

→ 5 occurrences for zero

4 " for one

This approach will take 2 passes on the array.

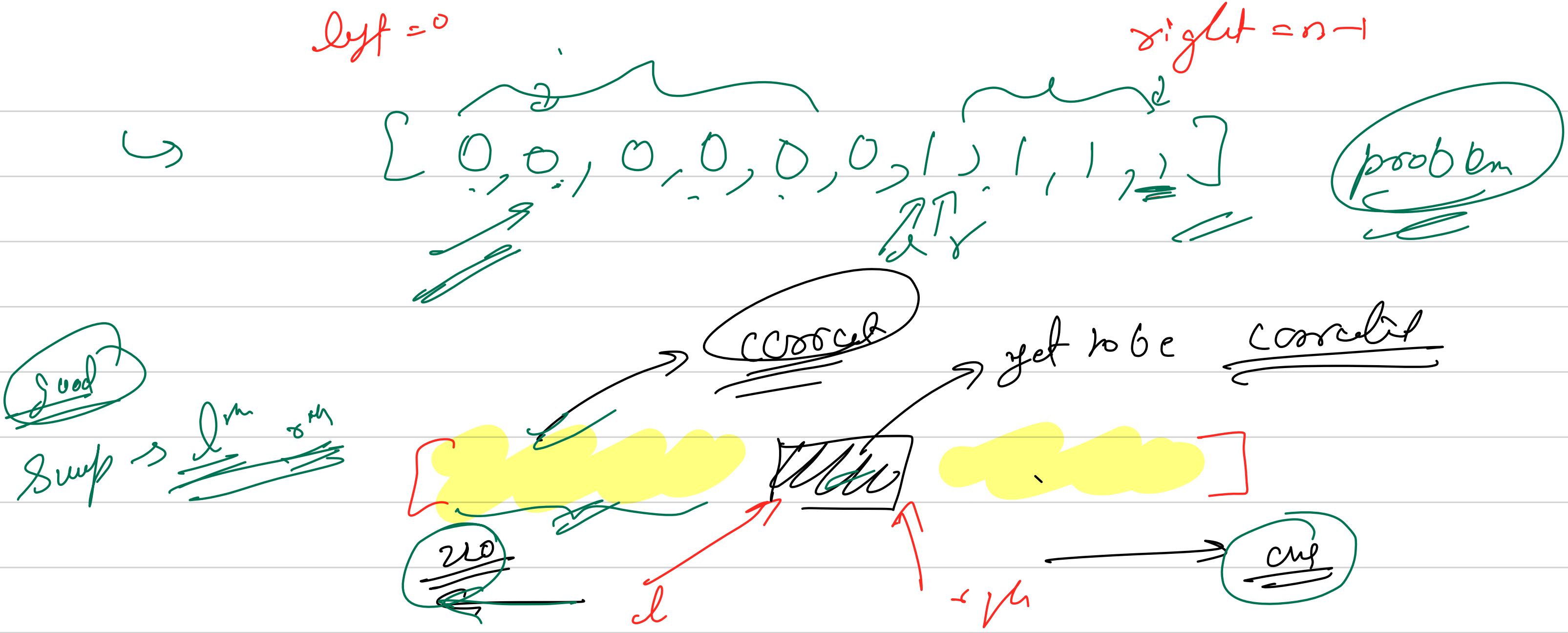
Can we do it in a single pass

→ location

→ Category of problems → Two pointers

2-3

Partition the array into 2 subsets such that one part is correct & 1 part is yet to be corrected


left = 0
right = n-1
r
l
[0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1]
problem
good
swap → lth rth
correct
yet to be correct
zero
one
l
r

Functions

→ algorithm

→ $y = f(x)$

input value → argument

output → return value

can be present or can be absent

→ This reduces code redundancy. ×

→ Cleanliness readability

for all the primitive data types (int, float, bool, etc)

If you pass them as arguments in the function then their original version is not shared instead they make a copy of themselves & pass by copy pass by reference → the original existence of the var is passed

By default arrays support it.

Write a function that takes an array as input where the array will have either zero, one or two & it sorts the array.

Counting X single pass, extra space X

$N \leq 10^7$

[2, 2, 1, 0] → [0, 1, 3 2]

Two poin

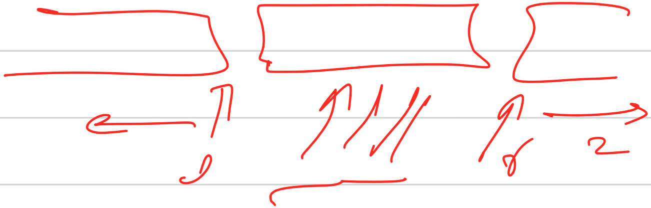
[0, 0, 0, 1, 1, 1, 2, 2 2, 2]

l r m

n operation

Single pass

no Conditi



```
{
if (m == 2) {
    swap(m, r)
}
```

```
if (arr[m] == 0)
    swap(arr[l], arr[m])
    l++, m++;
if (arr[m] == 1) m++
```

return 0;

K

$$a^{b} = a^{b/2} \times a^{b/2}$$

$$a^{b/2} = a^{b/4} \times a^{b/4}$$

$$a^{b/4} = a^{b/8} \times a^{b/8}$$

$$\vdots$$

$$a^{4} = a^{2} \times a^{2}$$

$$a^{2} = a^{1} \times a^{1}$$

$$a^{1}$$

$$b \to b/2 \to b/4 \to b/8 \to \cdots \to 1$$

$$\frac{b}{2^{k}} = 1$$

$$b = 2^{k}$$

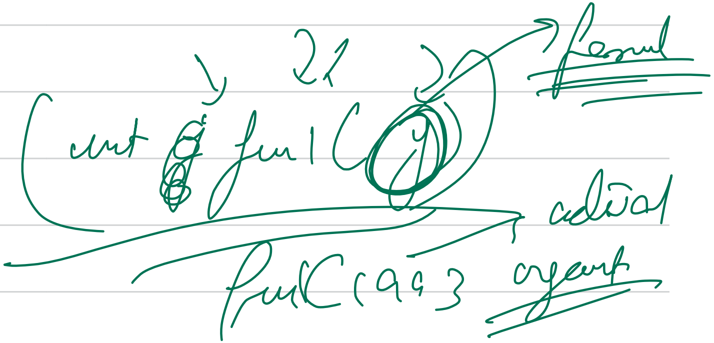
$$k = \log b$$

<return type> <nameof function> (<argument>) {

<code>

↳ <return value>

}



Q:- Function arguments can be of many types

↳ classification

actual   formal   default

① Default ⇒ the argument can have default values

argument's actual parameter

value

Q2 write a function to swap 2 number value a, b ⇒ a=10, b=20

temp = a

a = b

b = a

a, 10 b, 20

temp = a, 10

a = b

(a, 20)

(b, 10)

cout<<a << b;

→ count all the zeroes & all the ones

↳ start looping on the array and first fill it using the zeros till the til we put zeros equal to the count of zero, then put all ones