Tentative Plan

Friday, 10 September 2021 4:55 PM

Week-1

Remaining

- ✓ 1 Arrays and String
- ✓ 2 Stacks and Queue
- ✓ 3 Searching sorting
- ✓ 4 Heap and Hash Map
- ✓ 5 Tries 6 Bits

} L2

Schedule

- ~~1~~ Friday - 6:30 - 10:30
- ~~2~~ Saturday - 2:20 - 6:10
- ~~3~~ Sunday - 2:20 - 6:10

Introduction

1. Sync. → cooperate → Patience → cooperate

   Rajneesh ↔ Shreesh

   Gap

2. Hand Raise → Responsive
3. Doubt → $\boxed{\text{Honestly}}$

Day-1

1. ✓ ~~1~~ Long pressed name
2. ✓ ~~2~~ Container with most water
3. ✓ ~~3~~ Square of Sorted array
4. ✓ ~~4~~ Majority Element
5. ~~5~~ Majority Element - 2
6. ~~6~~ Majority Element General

H-W

Day 2

1. Next Greater - III
2. Max. product of 3 No.
3. Max chunk to make array sorted.
4. Max chunks - II
5. Longest Number atleast twice of others.
6. Sort array by parity

Day 3

1. No. of Subarray with bounded max
2. wiggle sort - 1
3. wiggle Sort - 2
4. Reverse vowels of string
5. Product of array Except itself
6. Maximise distance to closest one

Long Pressed Name

Friday, 10 September 2021 6:44 PM

Example 1

Name → "aabcc"

typed → "aaaabbbcc"

Is it possibly correct typing for name or not??

↳ True → It is possibly correct typing.

Example 2

Name → "abbccde"

Typed → "aabbcddee"

×

It is possibly not correct typing

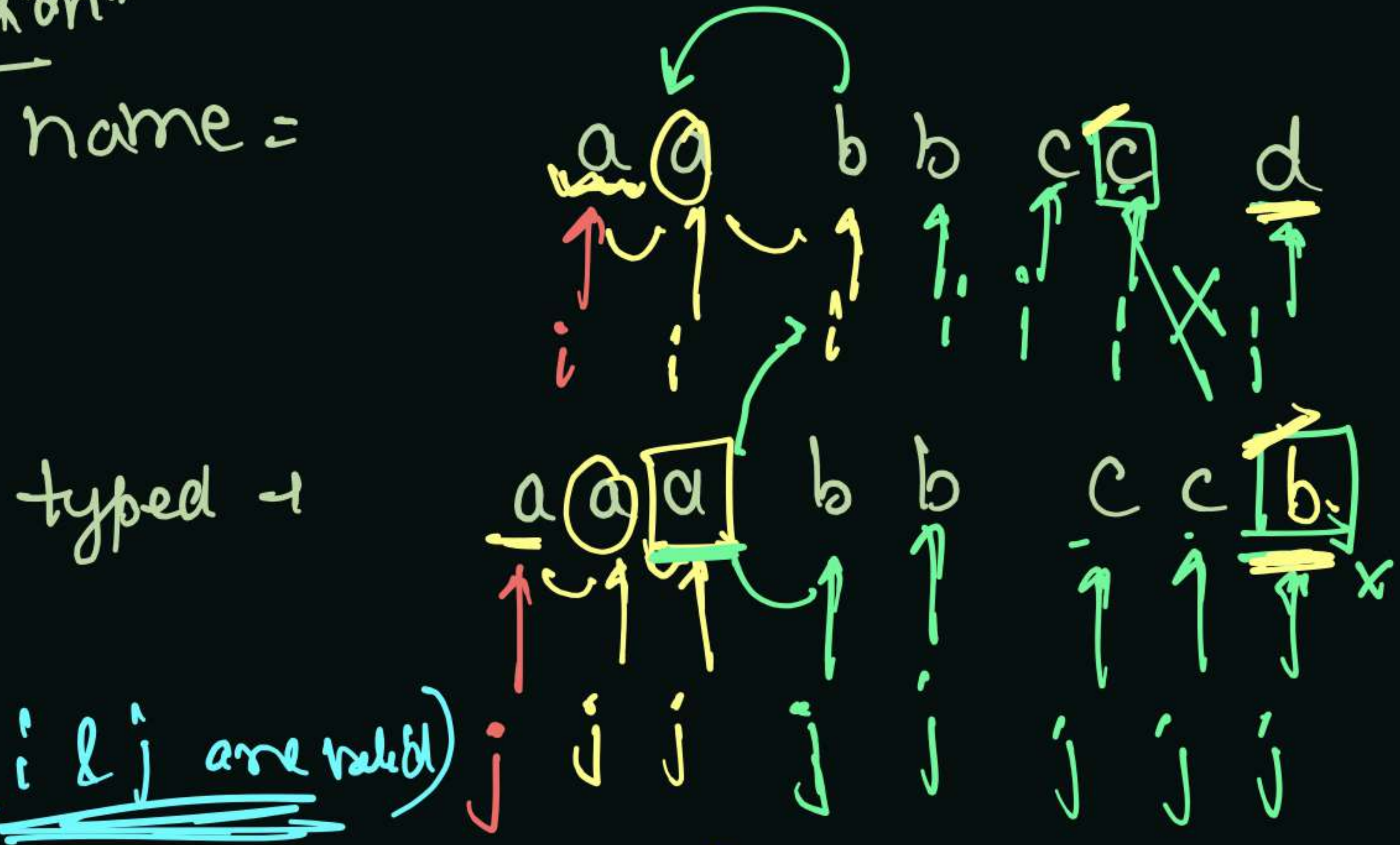
↳ False

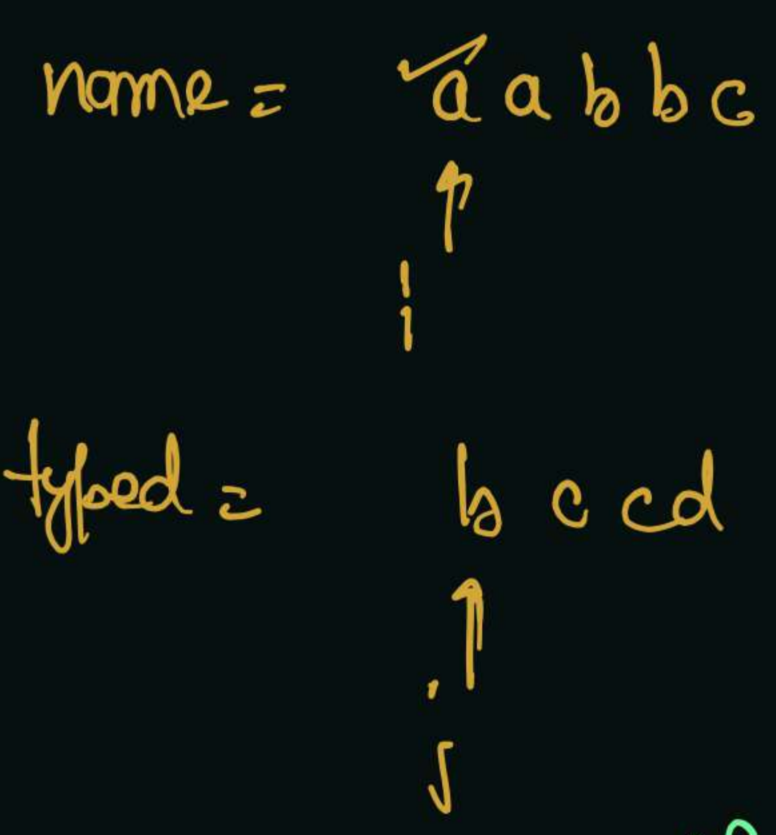
Edge cases

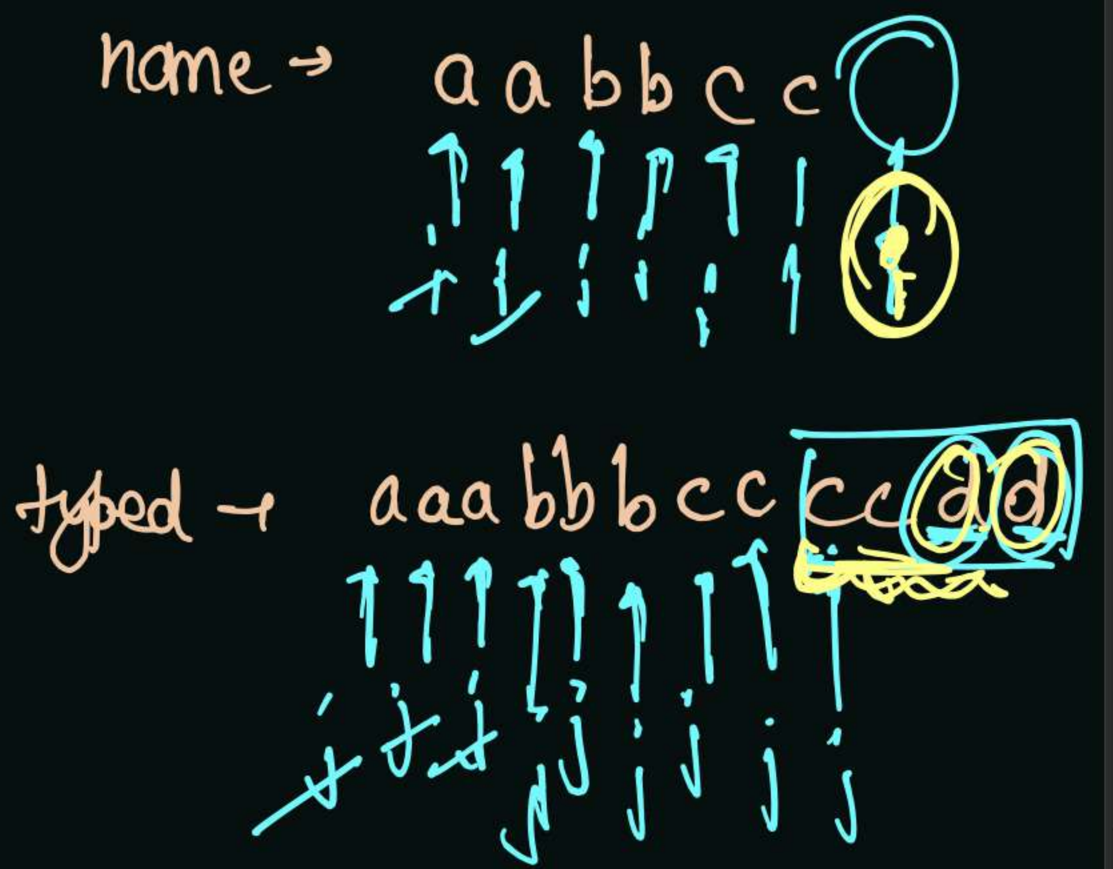
Hint → typed is correct in its Range.

Hint 2 → Name is longer than typed.

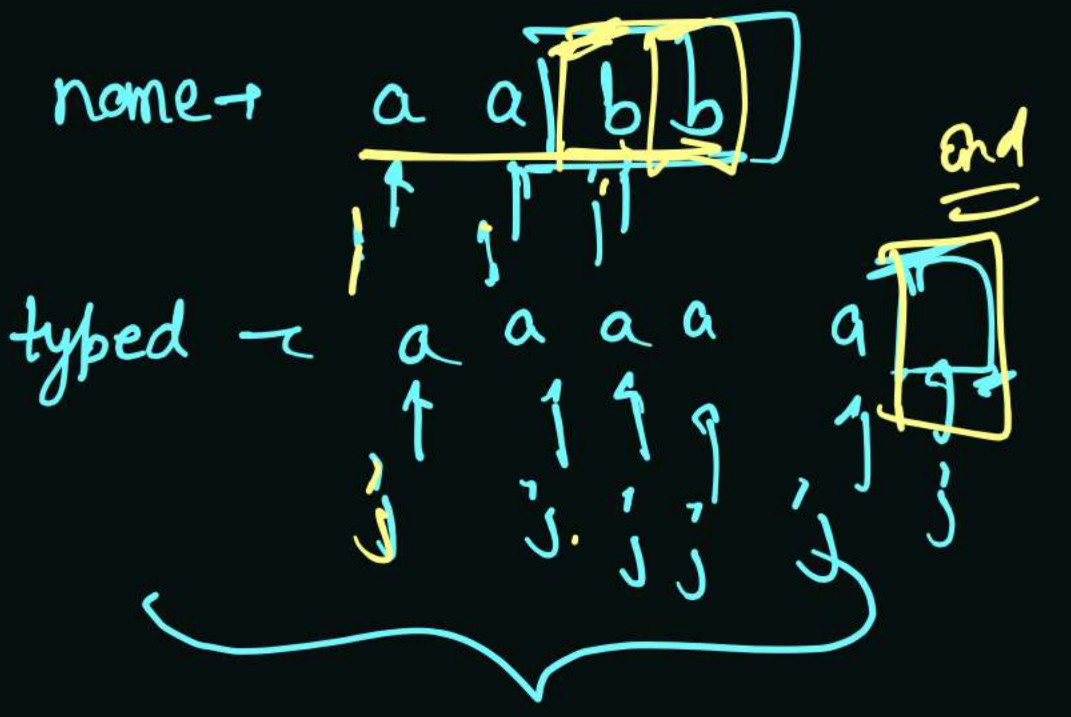
Implementation:







```
while (i & j are valid) {
    if (charAt(i) == charAt(j)) {
        i++;
        j++;
    } else {
        else => charAt(i) != charAt(j)
            validity of (i-1) &&
        if (charAt(j) == charAt(i-1)) {   → validity of (i-1)
            j++;
        } else {
            // j is wrong character
            return false;
        }
    }
}
```



Container With Most Water

Friday, 10 September 2021 7:41 PM

⇒ Assumption

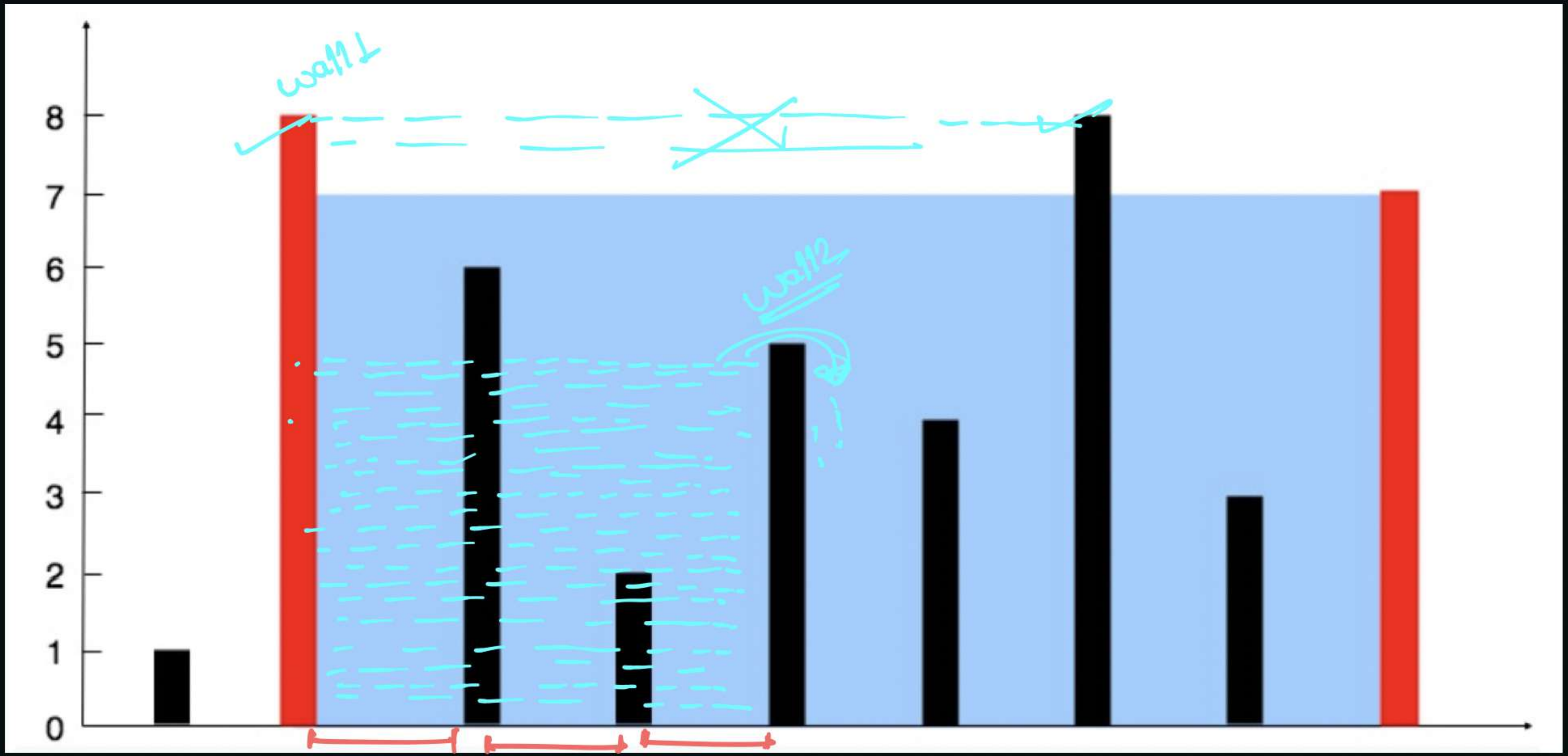
Container → length, breadth, height

breadth = 1 unit

water = l × b × h.

~~volume~~

Area ✓



hts → 1 8 6 2 5 4 8 3 7

indx 0 1 2 3 4 5 6 7 8

length = 4 − 1 = 3

Pair [wall 1 = 8 ht, wall 2 → 5 ht]

length = 4 − 1 = 3

height = min(wall1, wall2)

breadth = 1 unit ✗

height

water = l × b × h
= 3 × 5 × 1
= 15 unit

Container = length, breadth, height

volume = l × b × h.

maximise.

1	8	6	2	5	4	8	3	7
0	1	2	3	4	5	6	7	8

All pairs:→ complexity - $O(n^2)$

Brute force Approach

0,1	1,2	----	6,7	7,8
0,2	1,3	----	6,8	
0,3	1,4	--		
0,4	1,5			
0,5	1,6			
0,6	⋮			
0,7	1,7			
$\boxed{0,8}$				

max water

water = l × h

h = min(ht[i], ht[j])

l = j-i

length.

1 8 6 2 5 4 8 3 7

$\boxed{0}$ 1 2 3 4 5 6 $\boxed{7}$ - $\boxed{8}$

↑ i ↑ j

length = j - i - 1

h = m

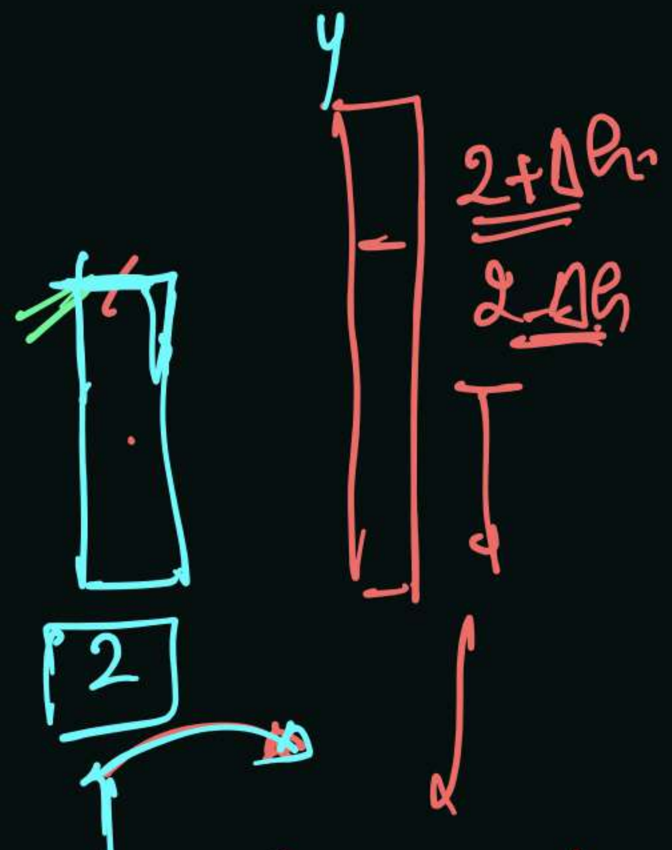
current volume = Math.min(ht[i] ht[j]) * (j - i)

water = max(water, current volume)

```
if( ht[i] < ht[j]) {
    i++;
} else {
    j--;
}
```

ht = 2

length = j - i



(2+Δh) i

height = min(2+Δh, 5)

length = j - i - 1

water ↑ may be increase

water ↓ may be water will decrease

length = 9

height = 4

water = $\boxed{36}$

2 × 10 = $\boxed{20}$

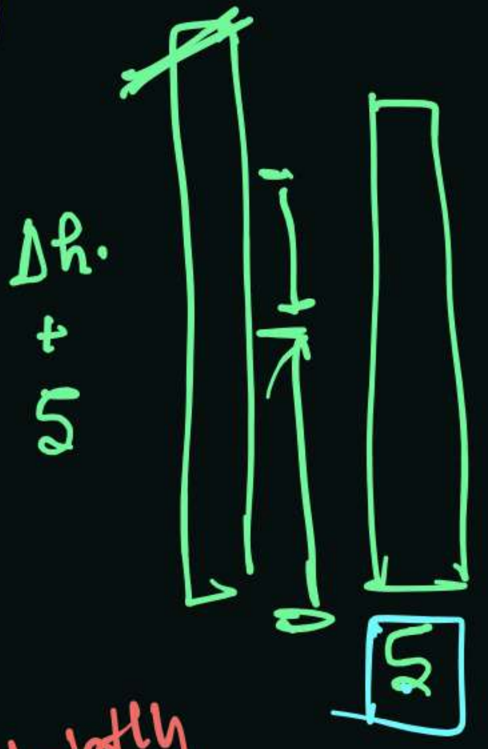
j - i = 10

j - i ↓

ht = min(2, 5-Δh)

length = j - i - 1

water ↓



definitely water decrease

j (j-1) ↑

② ht = min(2, 5+Δh)

length = j - i - 1

water = ↓

Square of Sorted Array

Friday, 10 September 2021 8:53 PM

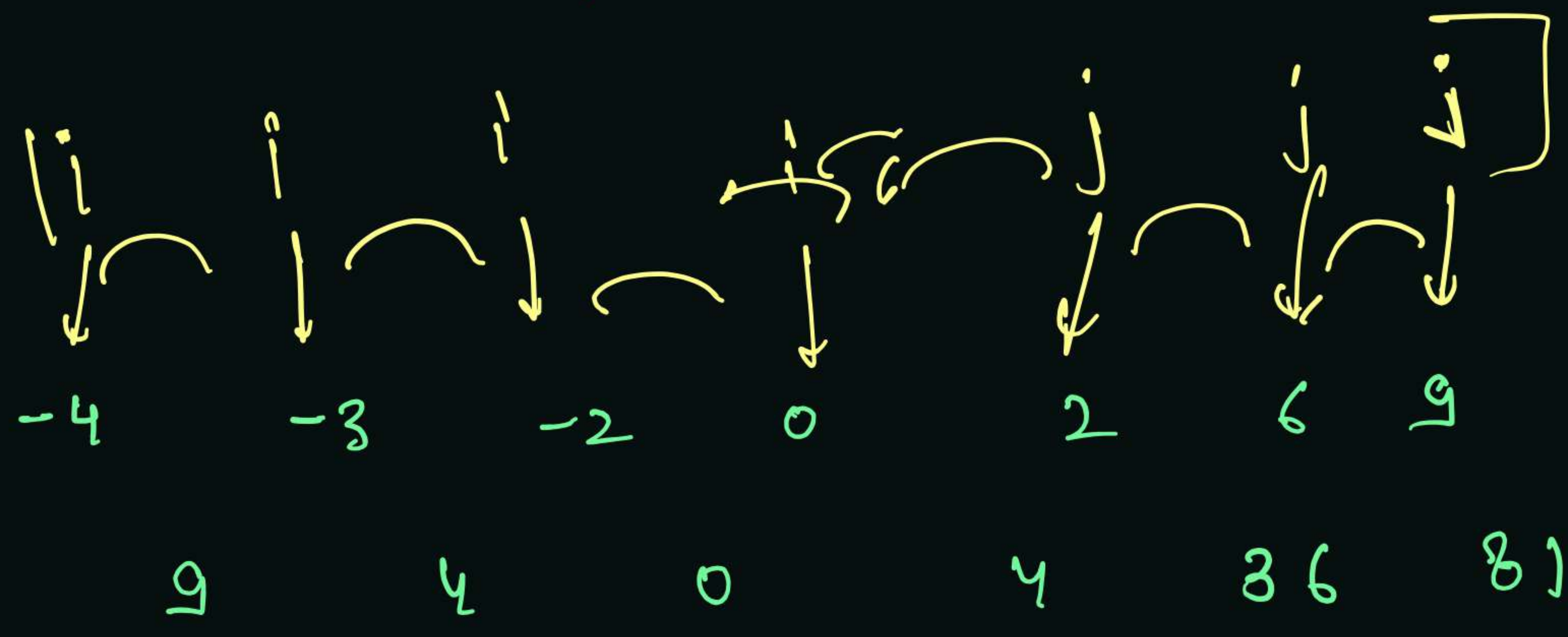
array → 0 2 3 6 9 10

0 1 2 3 4 5

Approach → Square → 0 4 9 36 81 100

X All no. are +ve Element

if -ve no. are there.

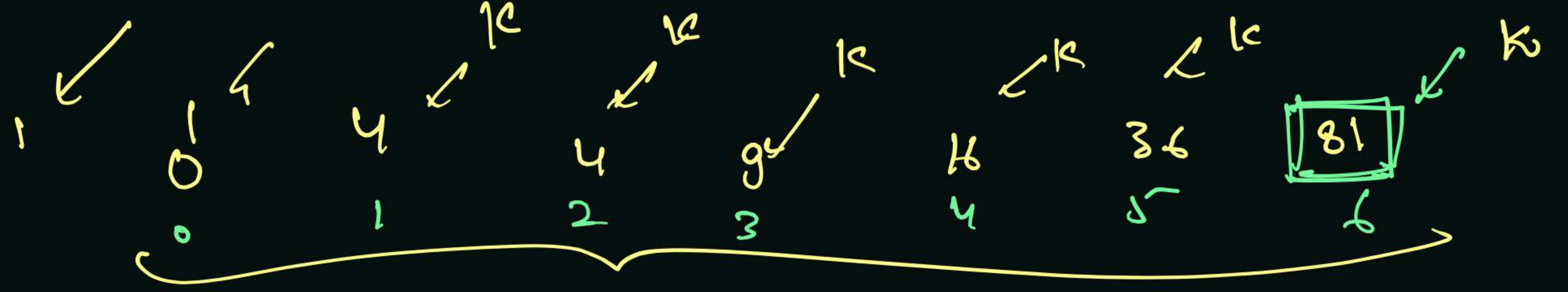


max square

least -ve

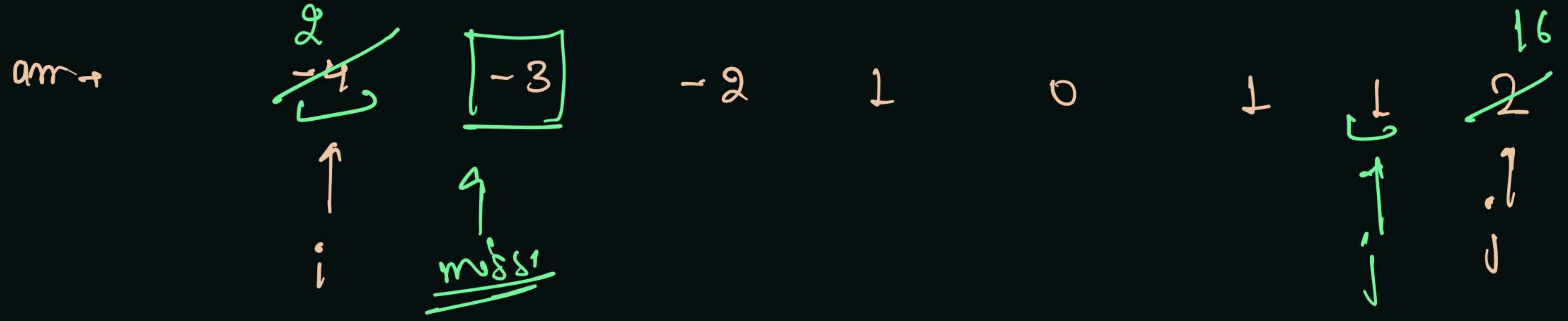
most +ve

Square → Sorted X



-ng →

Why swapping of element will not work?



```
if (nums[j] * num[j] < nums[i] * num[i]){
    swap( nums, i, j);
    j--;
}
```

16 > 4 → swap , j--;

comparing → 2, 1 → $\textcircled{1}$, $\textcircled{4}$

Sol

miss → $\textcircled{8}$

↳ complex Algorithm

Majority Element

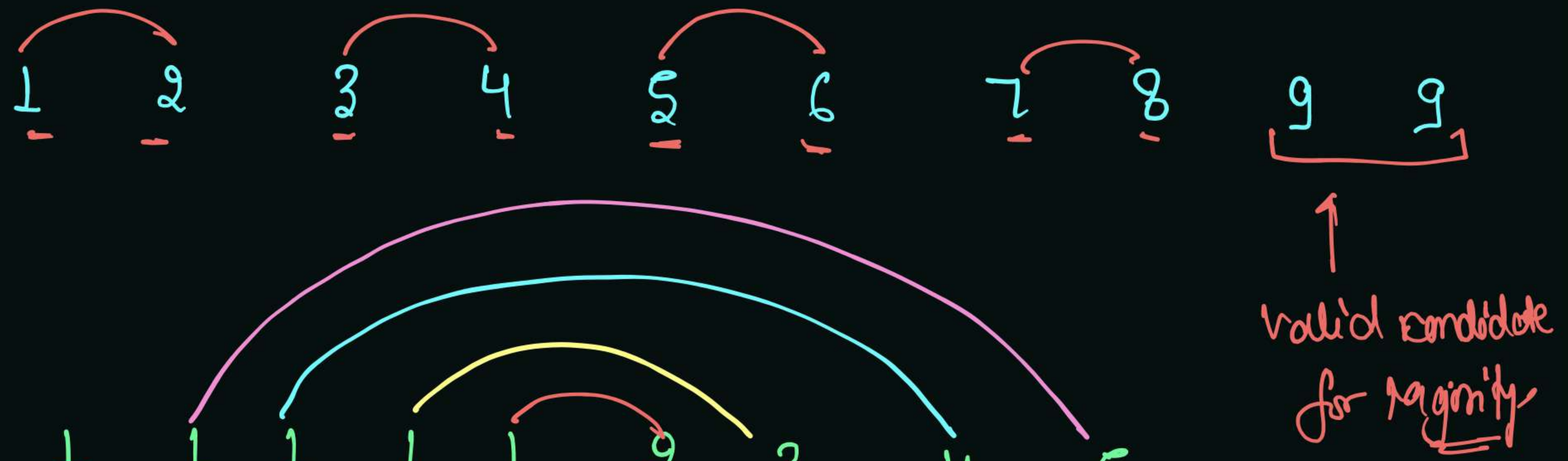
Friday, 10 September 2021 9:07 PM

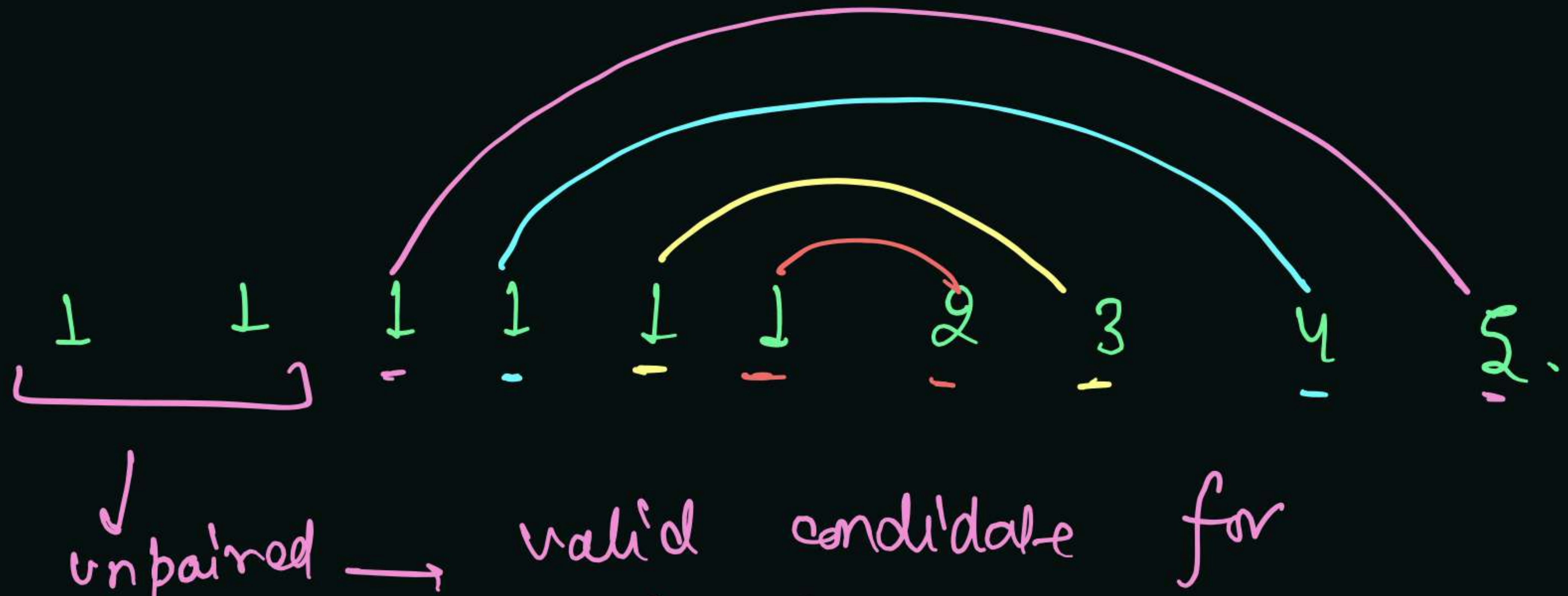
Majority → No. of occurrence > $n/2$

Valid candidate → solution.

→ pairing of distinct Element

→ unpaired Element → May be it is Majority Element.





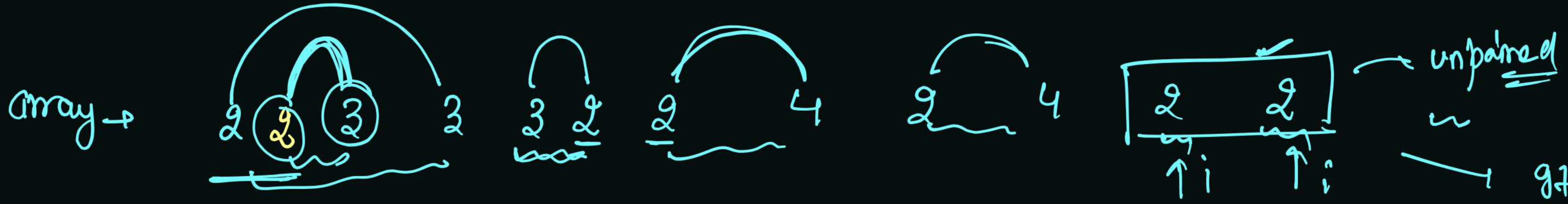
unpaired → valid candidate for Majority Elemt.

Steps for solution →

① Find valid candidate for Majority Element with pairing of distinct Element

② Check if candidate is present in Majority or not?

How to find valid candidate for majority using pairing of $\underbrace{\text{distinct element}}$??



→ It may be majority element

MOORE'S VOTING ALGO.

To find Majority.

val = arr[0]
count = 1

```
if( val1 == arr[i]){
        count1++;
}else{
    if(count > 0){
        count--;
    }else{
        val = arr[i];
        count = 1;
    }
}
```

↳ If majority exist then definitely val is majority

val = ~~2~~ ~~2~~ 2

count = ~~1~~ ~~2~~ 1 ← ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ 2

Majority Element 2

Friday, 10 September 2021 9:51 PM

Majority $> n/3$

How many Elements have more than $n/3$ freq?

$\underline{x} \times \frac{\cancel{n}}{3} = \cancel{n}$

21 $\frac{21}{3} = \textcircled{7}$

$\boxed{x = 3} \longrightarrow \underline{\text{Max}}$ for equal to $n/3$

$\underbrace{1\ 1\ 1\ 1\ 1\ 1\ 1}_{n/3}\ \underbrace{3\ 3\ 3\ 3\ 3\ 3\ 3}_{n/3}\ \underbrace{\cancel{5}\ \cancel{5}\ 5\ 5\ 5\ 5\ 5}_{n/3}$

1 2

for more than $n/3$ fre.

$\underline{x < 3} \longrightarrow \underline{\text{Max} = 2}$

arr → 1 2 $\boxed{3}$ 3 2 $\boxed{1}$ 3 1 $\boxed{2}$ $\boxed{2}$ $\boxed{1}$

$n/3$

↑ i

val1 = arr[0]

count1 = 1

val2 = arr[0] //any initial val.

count2 = 0

✓val1 = ~~1~~ ~~3~~ $\boxed{2}$

count1 = ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ 1

✓val2 = ~~1~~ ~~2~~ $\boxed{1}$

count2 = ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ 1

unmatched

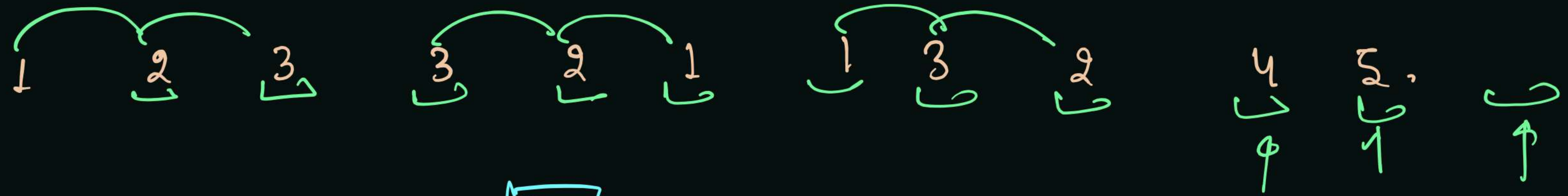
```
loop of i
if(arr[i] == val1){
        count1 ++;
} else if ( arr[i] == val2) {
        count2++;
} else {
        if( count1 == 0){
            val1 = arr[i];
            count1 = 1;
        } else if( count2 == 0){
            val2 = arr[i],
            count2 = 1;
        } else {
            count1 --;  count2 --;
        }
}
```

Check if val1 and val2 have more than $n/3$ freq

Edge case

$\boxed{val2 \; != val1}$

while adding in result



val1 = ~~1~~ ~~2~~ ~~1~~ $\boxed{4}$

count1 = ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ $\boxed{1}$

val2 = ~~1~~ ~~2~~ ~~3~~ $\boxed{5}$

count2 = ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ $\boxed{1}$

if majority Exist then it is in val1 and val2 → according to definition

is val1 and val2 present in majority ??

Majority Element General

Friday, 10 September 2021 10:20 PM

$\longrightarrow$ Discussion

Majority $\longrightarrow$ having freq more than $\boxed{\frac{n}{K}}$

① val → array size → k-1

count → array size → k-1

② $\boxed{\frac{n}{k}}$ $k>3$ → HashMap

if consider equal of $\boxed{\frac{n}{k}}$ elements, how many elements are possible.

freq map

check if freq is more than n/k add it in res.

$$x \times \frac{\cancel{n}}{k} = \cancel{n}$$

no. of set

$\boxed{x = k}$

so for more than $\frac{n}{k}$, it should be $\boxed{k-1}$

$\frac{n}{3} \longrightarrow$ 2 Ele

$\frac{n}{2} \rightarrow$ 1