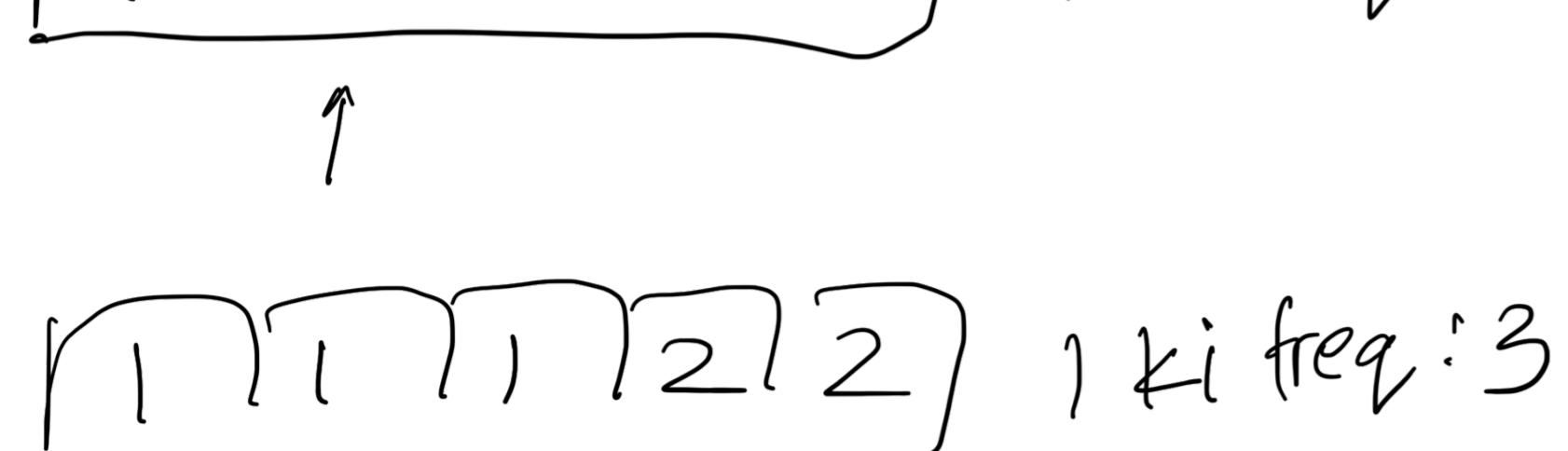
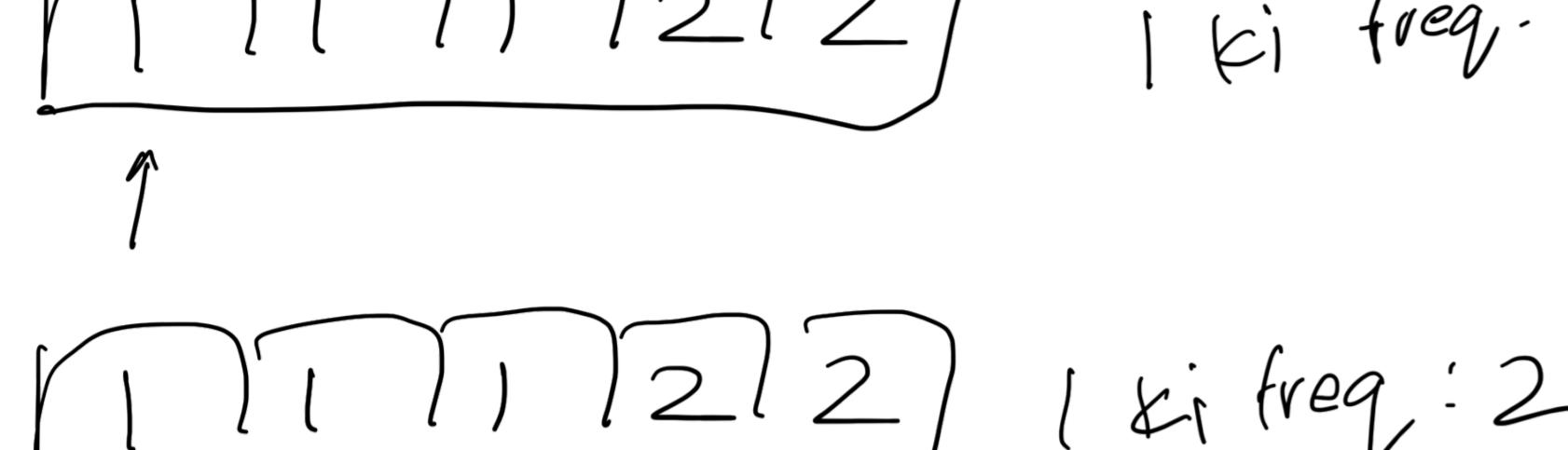


majority element
 $[1, 2, 2, 1, 1]$
 $\lfloor \frac{n}{2} \rfloor$
 majority element > $\lfloor \frac{n}{2} \rfloor$

① Brute force

```
for (int val : nums) {
    freq = 0
    for (int element : nums) {
        if (element == val) {
            freq++
    }
    if (freq > n/2) → majority element
}
```

② Optimized solution (Sorting based)



$3 > \frac{n}{2}$
 $3 > \frac{5}{2}$
 $3 > 2 \Rightarrow$ majority element

```
freq = 1; ans = nums[0]
for (i = 1; i < n; i++) {
    if (num[i] == num[i - 1]) {
        freq++
    } else {
        freq = 1; ans = num[i]
    }
    if (freq > n/2)
        return ans
}
```

Dry run

$| 1 | 2 | 2 | 2$
 \uparrow
 $freq = 1$ $ans = 1$

$| 1 | 2 | 2 | 2$
 \uparrow
 $freq = 2$ $ans = 1$

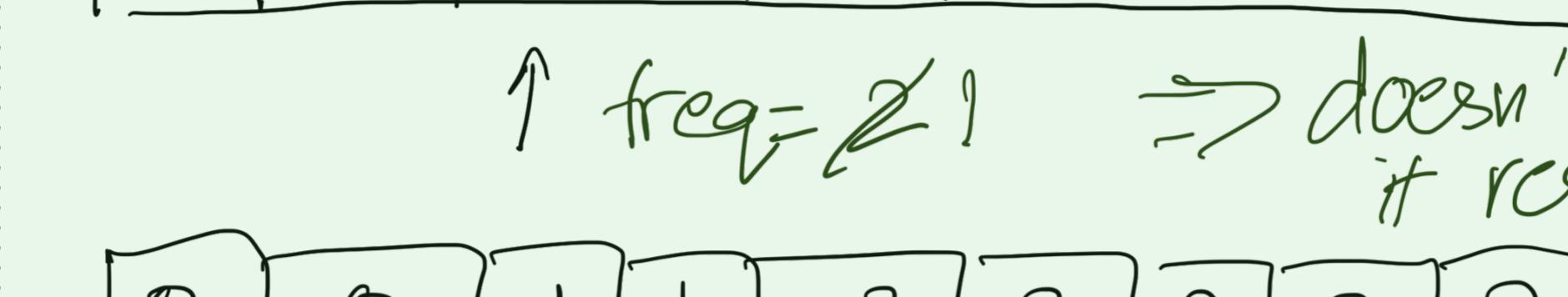
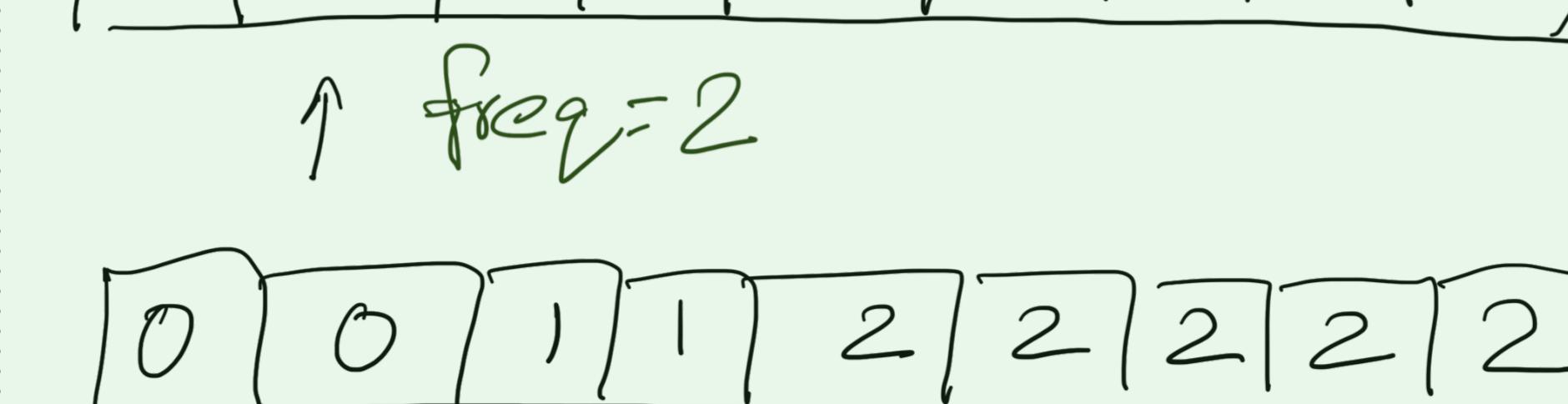
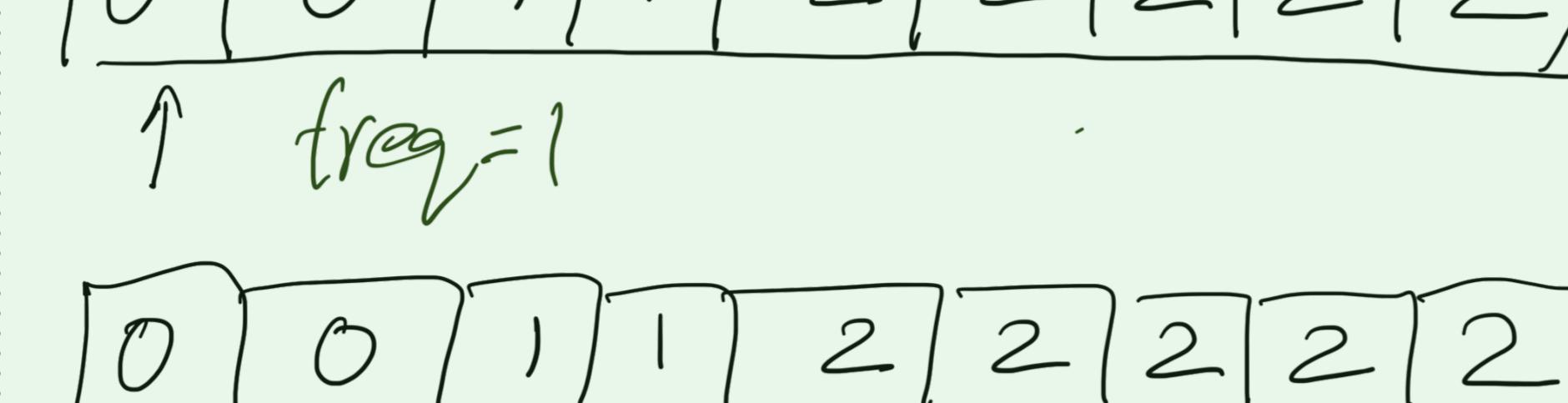
now element \neq prev \therefore freq resets to 1
 $| 1 | 2 | 2 | 2$

\uparrow
 $freq = 1$ $ans = 2$

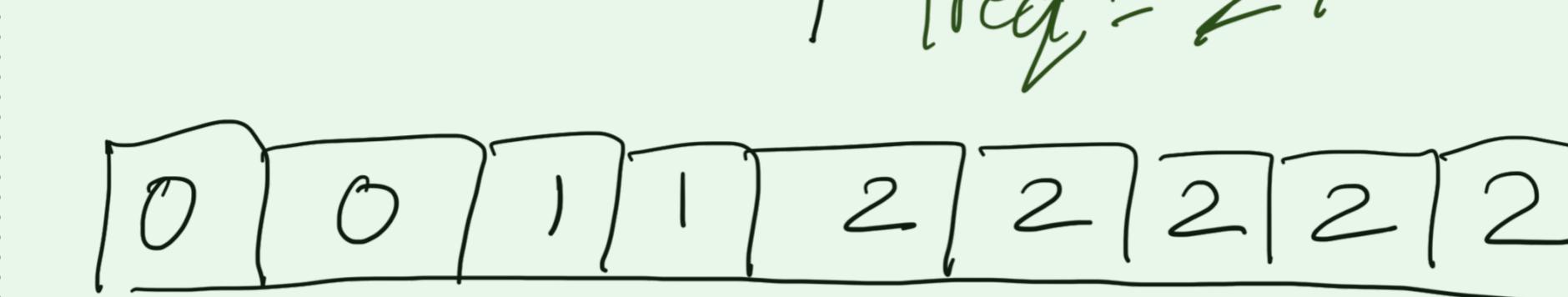
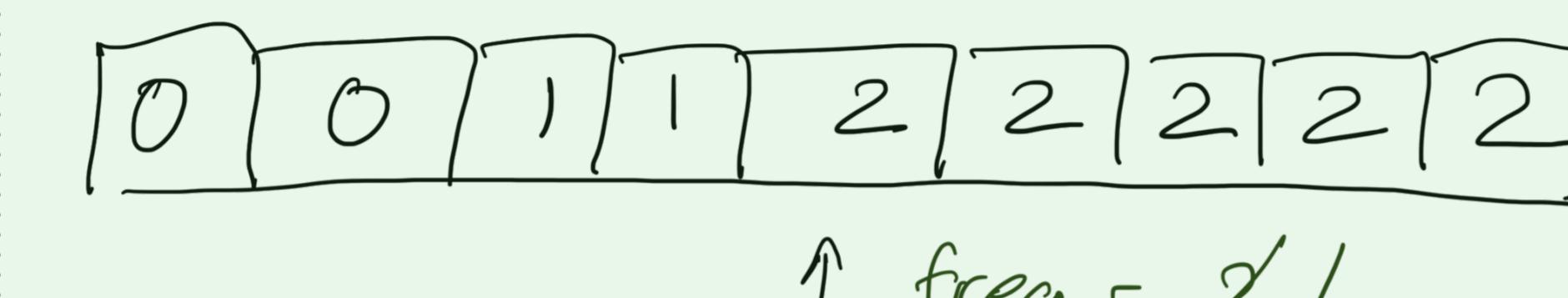
$| 1 | 2 | 2 | 2$
 \uparrow
 $freq = 2$ $ans = 2$

$| 1 | 2 | 2 | 2$
 \uparrow
 $freq = 3$ $ans = 3$

$freq > n/2 \Rightarrow$ return 3



$1 freq \geq 1 \Rightarrow$ doesn't matter that it reset because it's not the majority element anyways



$1 freq = 2$

Same like optimized but no sorting & we will not reset freq instead freq --

same element \Rightarrow freq ++
 diff element \Rightarrow freq --

```
freq = 0, ans = 0
for (int i = 0; i < n; i++) {
    if (freq == 0) {
        ans = nums[i]
    }
    if (ans == nums[i]) freq ++
    else freq --
}
return ans
```

time complexity: $O(n)$

③ Moore's algorithm (most optimized)

$| 1 | 2 | 2 | 1 | 1$

Same like optimized but no sorting & we will not reset freq instead freq --

same element \Rightarrow freq ++

diff element \Rightarrow freq --

```
freq = 0, ans = 0
for (int i = 0; i < n; i++)
```

if (freq == 0)

ans = nums[i]

if (ans == nums[i]) freq ++

else freq --

return ans

time complexity: $O(n)$

Dry run
 $freq = 0$ $ans = 0$

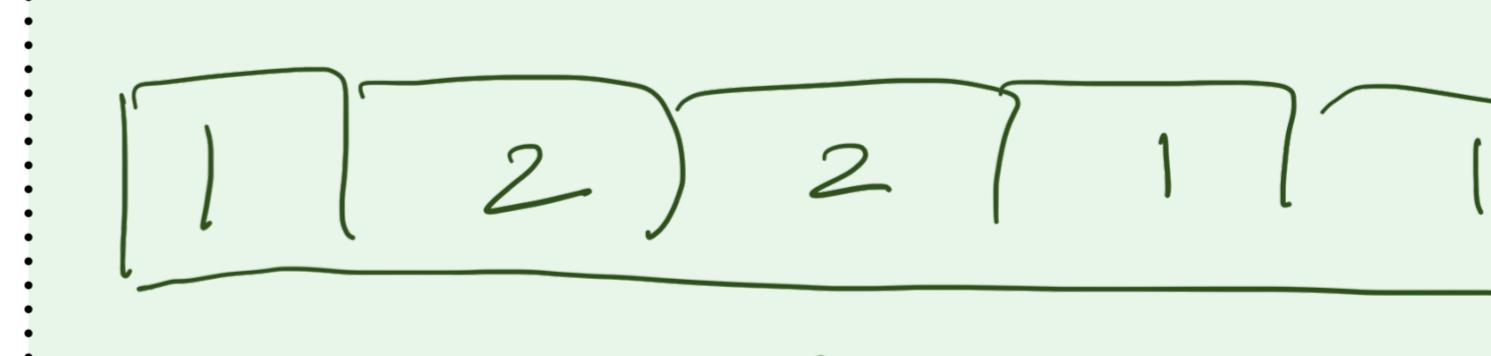


$freq = 1$ $ans = 1$



$num[i] = 2 \neq ans \Rightarrow freq --$

$freq = 0$ $ans = 1$



$freq = 0$ $\&$ freq ++ \Rightarrow ans gets updated to 2

$freq = 1$ $ans = 2$



$num[i] = 1 \neq ans \therefore freq = - -$

$freq = 0$ $ans = 2$



$freq = 0 \&$ ans gets updated to 1

$freq = 1$ $ans = 1$

freq is basically like a count
 It will always be positive & never zero if majority element exists