Interview Problems

Q Given a array of 0's & 1's, you are allowed to replace atmost one 0 with one 1. Return length of max possible continous 1's

MS Amazon LinkedIn

Eg A: 0 1 1 1 0 1 1 0 1 1 0
(x = 3, y = 2)

Contest 23 Nov
9 PM – 10:30 PM

Obs: Whenever we replace a 0 with a 1, the resulting length that we get is $x + y + 1$

Code

```
ans = 0
for ( i = 0 ; i < N ; i++ ) {
    if ( ar[i] == 0 ) {
        x = 0    // count 1's on the left side
        for ( j = i-1 ; j >= 0 ; j-- ) {
            if ( a[j] == 1 )
                x++
            else
                break
        }
        y = 0    // count 1's on the right side
        for ( j = i+1 ; j < n ; j++ ) {
            if ( a[j] == 1 )
                y++
            else
                break
        }
        ans = max ( ans, x+y+1 )
    }
}
return ans
```

- What is TC ?

Ex 0 1 1 1 1 0 1 1 1 0 1 1 0 1

Obs: No element is visited more than 3 times.

Max no of iterations = $1 \rightarrow 3$, $N \rightarrow 3N$

TC: $O(N)$

Edge Case Ex: 1 1 1 1 1 1 1

- If all 1's, $\Rightarrow$ ans = n

1 1 1 1 0 0

1 1 0 1 1 1

Q Follow-up instead of swap replace

1) Count total no of 1's

2) for every 0 in array.

calc 1's in the left → l
calc 1's in the right ⇒ r

if l+r < total
⇒ same ans (l+r+1)

if l+r = total
no 1 available to swap
l+r

0 0 0 1 0 0 0

$l = 0$ $r = 1$

$Res = total = 1$

Ola

Q1) Given N +ve elements in integer array
Find majority element.
↳ elem with freq > N/2 (N → size of array)
freq → no of occurences

Eg 1: A = { 1 2 1 6 1 1 }

1 — 4 > 6/2 — 4 > 3 — ✓

Eg 2: A = { 3 4 4 8 4 9 4 3 4 }

4 — 5 > 9/2 — ✓

Eg 3 A = { 4 6 5 3 4 5 6 4 4 4 }

4 — 5 > 10/2 — 5 > 5 ✗

no majority elem

Solutions :

1) Count freq for every element
TC: $O(N^2)$

2) Sort the array & then count.

A: [3 2 4 6 6 3 4 4 2 3]

sorted A: [2 2 3 3 3 4 4 4 6 6]

?

TC for sorting = $n \log n$

TC: $n \log n + n = O(n \log n)$

Say 2 majority elem are present

ele 1 ele 2

freq(ele1) > N/2

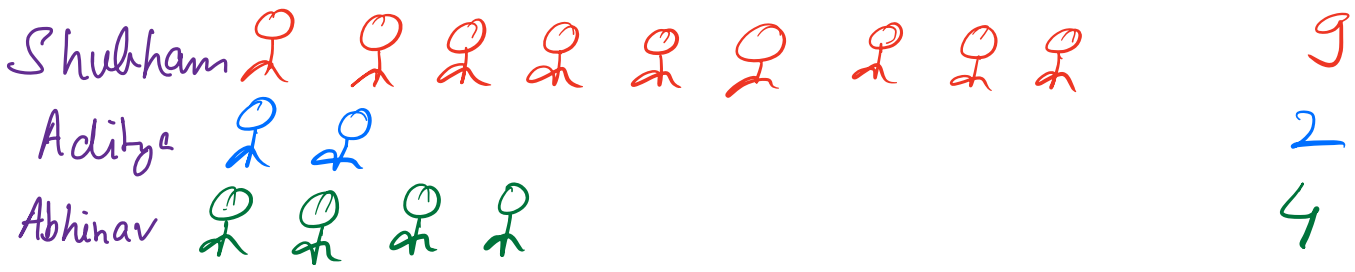
freq(ele2) > N/2

freq(elem1) + freq(elem2) > N

Max no of majority elem 1/0

College Election

Say 15 seats.



S	Ad	Ab	
9	2	4	→ Shubham
8	1	4	→ Shubham
7	1	3	→ Shubham
7	0	2	→ Shubham

Obs: If we remove 2 different elem, majority elem is still same.

	0	1	2	3	4	5	6	7	8
Eg:	4	4	3	8	8	4	9	4	4
count	1	2	1	0	1	0	1	0	1
elem	4	4	4	4	8	8	9	9	4

4 4 3 8 8 4 9 4 4

ans = 4

Algorithm of deleting 2 distinct elements is called Moore's Voting Algorithm

A: {1, 2, 3, 4, 5, 6}

A: {1, 2, 3, 4, 5, 6, 7}

⇒ ?

	0	1	2	3	4	5	6	7	8	9
Eg 1	4	6	5	3	4	5	6	4	4	4
count	1	0	1	0	1	0	1	0	1	2
elem	4	4	5	5	4	4	6	6	4	4

elem = 4 freq = 5

Now, calc freq(4) $5 \geq 10/2$

ans = No elem is majority

	0	1	2	3	4	5	6
Eg 2	4	3	3	3	6	3	7
count	1	0	1	2	1	2	1
elem	4	4	3	3	3	3	3

Check if 3 is majority? $4 > 7/2$

	0	1	2	3	4	5	6
Eg 3	3	3	3	3	4	4	4	1
count	1	2	3	2	1	0	1
elem	3	3	3	3	3	3	1

Check if 1 is majority? No

	0	1	2	3	4	5	6	7	8	9
Eg 4	4	2	3	3	3	4	2	4	1	4
count	1	0	1	2	3	2	1	0	1	0
elem	4	4	3	3	3	3	3	3	1	1

Is 1 majority elem? No

Code

```
int majority (int arr[], int N) {
    elem = a[0]
    freq = 1

    for(i=1 ; i<N; i++) {

        if (arr[i] == elem) {
            freq ++
        }
        else {
            if (freq == 0) {
                elem = a[i]
                freq = 1
            }
        }
```

```
        else {
            freq--
        }
    }
}

// check if elem is majority.
int c=0
for(i=0; i<N; i++) {
    if(arr[i] == elem)
        c++
}

if(c > n/2)
    return elem
else {
    No majority
    return -1
}
}
```

TC: O(n) SC: O(1)

03 Row to column zero

integer

2D matrix of positive. If any row has 0, make the row all 0. If any colum has 0, make col all 0

1	2	3	4
5	6	7	0
9	2	0	4

⇒

1	2	0	0
0	0	0	0
0	0	0	0

1	2	0	0
0	0	0	0
0	0	0	0

original 0

converted 0 → -1

idea : Convert to -1

-1	0	-1	0
5	-1	7	-1
9	-1	1	-1

0	0	0	0
5	0	7	0
9	0	1	0

Code

```
for ( i=0 ; i<n ; i++) {
    f = 0
    for ( j=0 ; j<m ; j++) {
        if ( ar[i][j] == 0 )
            f = -1
    }
    if ( f == -1 ) {
        for ( j=0 ; j<m ; j++) {
            if ( ar[i][j] != 0 )
                ar[i][j] = -1
        }
    }
}
```

```
for ( j=0 ; j<m ; j++) {
    f=0
    for ( i=0 ; i<n ; i++)
        if ( ar[i][j] == 0 )
            f = -1
    }
    if ( f == -1 ) {
        for ( i=0 ; i<n ; i++)
            if ( ar[i][j] != 0 )
                ar[i][j] = -1
        }
    }
}
```

Now convert all -1's to 0.

{done}