Interview Problems

Question 1

Given a binary array, we can atmost replace a single 0 with 1.

(arr is either 0 or 1)

(<= 1)

Find max consecutive 1's you can get in array.

eg 1 1 0 1 1 0 1 1

len = 5

len = 5

ans = 5

eg 1 1 0 1 1 0 1 1 1

len = 5

len = 6

eg 0 1 1 1 0 1 1 0 1 1 0

len = 6

eg 1 1 1 1 1

len = 5

logic :

for every 0,

1. count consecutive 1's in left = l
2. count consecutive 1's in right = r

len = l+r+1

```
int replace (a[], n){
    c=0
    for( i=0 to n-1)
        c+=a[i]   // count of 1's in array
    if (c==n)
        return n

    ans=0
    for(i=0 to n-1){
        if(a[i]==0){
            l=0, r=0
            for(j=i-1; j>=0; --j){
                if(a[j]==1)  ++l
                else  break
            }
```

if (a[i]==1)
c++

total 1's in the left

```
        for ( j=i+1 to n-1){
            if (a[j] ==1)  ++r
            else  break
        }
        ans = max( ans, l+r+1)
    }
}
return ans
}
```

total 1's in the right

$TC = O(\cancel{N^2})$

$O(N)$

$SC = O(1)$

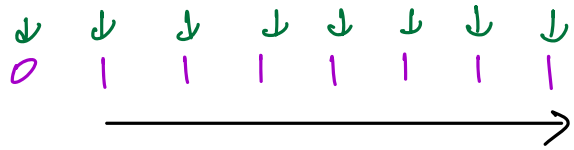
	0	1	1	0	1	1	0	1	1	0
l	0			2			2			2
r	2			2			2			0
$l+r+1$	3			5			5			3

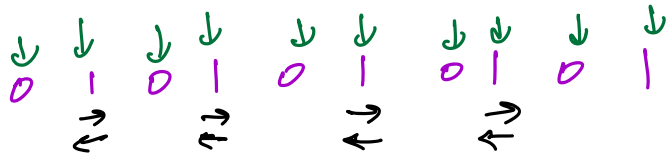
all elements are visited <=3 times

total iterations <= 3N

$TC = O(N)$

eg

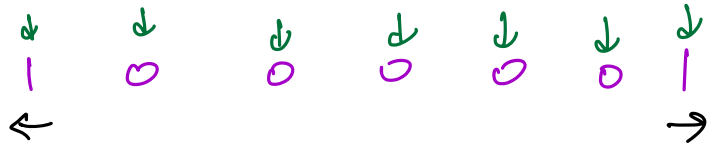




$l = 3$ $\quad$ $l = 0$

$r = 0$ $\quad$ $r = 2$

$l + r + 1 = 4$ $\quad$ $l + r + 1 = 3$



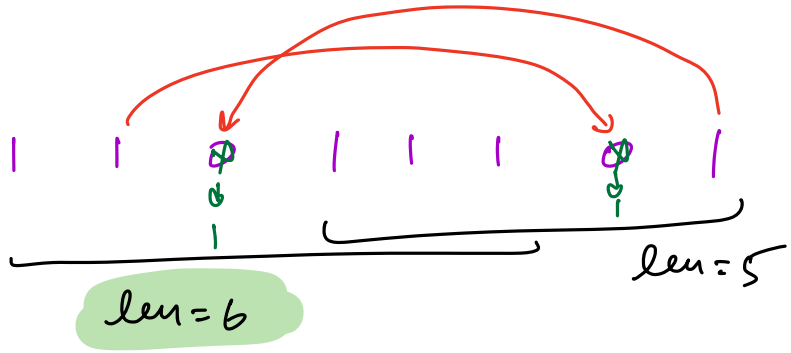
Lesson: If there is a break in any loop, calculate TC carefully.

Question 1 - Part 2

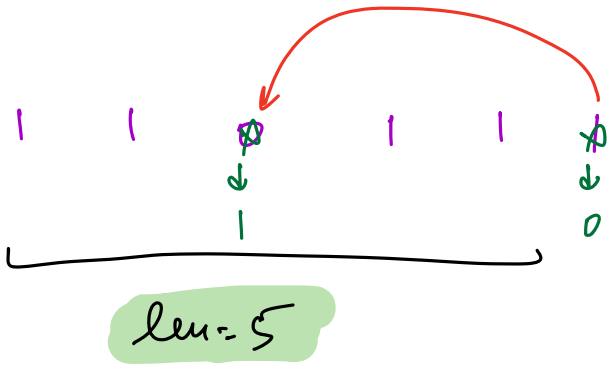
Given a binary array, we can atmost swap single 0 with 1.

Find max consecutive 1's.

eg



eg




```
int swap (a[], n){
    c=0
    for( i=0 to n-1)
        c+=a[i]   // count of 1's in array
    if (c==n)
        return n
```

if (a[i] == 1)
c++

```
ans = 0
for (i=0 to n-1) {
    if (a[i] == 0) {
        l=0, r=0
        for (j=i-1; j>=0; --j) {
            if (a[j] == 1)  ++l
            else  break
        }

        for (j=i+1 to n-1) {
            if (a[j] == 1)  ++r
            else  break
        }
        if (l+r == c) {
            ans = max(ans, l+r)
        else
            ans = max(ans, l+r+1)
    }
}
return ans
}
```

total 1's in the left

total 1's in the right

$TC = O(N)$

1 0 1 0 1 1 0 1 0

2 1

$c = 5$

$2+1+1 = 4$

Question 2 – Majority element

Given N elements, find majority element.

↳ ele. with frequency $> N/2$

eg 1 2 1 6 1 1 $N=6$

freq(1) = 4 $N/2 = 6/2 = 3$ $\Rightarrow 4 > 3$ YES

3 4 3 6 1 3 2 5 3 3 3 $N=11$

freq(3) = 6 $N/2 = 11/2 = 5$ $6 > 5$ YES

4 6 5 3 4 5 6 4 4 4 $N=10$

freq(4) = 5 $N/2 = 10/2 = 5$ $5 \not> 5$ NO

At max how many majority elements can be there in array?

assume that there are 2 majority elements x & y

$$freq(x) > N/2$$

$$freq(y) > N/2$$

$$freq(x) + freq(y) > N/2 + N/2$$

$> N$ invalid since we only have N elements

<u>Idea</u>: count freq of each element & compare with $N/2$

1. Using 2 nested loops $\rightarrow$ $TC = O(N^2)$ $SC = O(1)$
2. Hashmap $\rightarrow$ $TC = O(N)$ $SC = O(N)$
3. Sort the array $\rightarrow$ $TC = O(N \log N)$ $SC = O(1)$

$$a = [1 \quad 3 \quad 1 \quad 5 \quad 3 \quad 3 \quad 5]$$

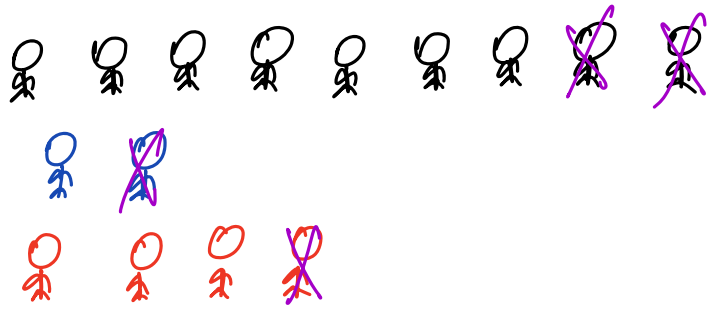
$$sorted(a) = [\underbrace{1 \quad 1}_{2} \quad \underbrace{3 \quad 3 \quad 3}_{3} \quad \underbrace{5 \quad 5}_{2}]$$

Election → 15 MLAs

Pranav →

Chandra →

Sunil →



Pranav = 9 $\quad 15/2 = 7 \quad 9 > 7$ ✓

↓ 2 disqualified

Pranav = 8 $\quad 13/2 = 6 \quad 8 > 6$ ✓

↓ 2 disqualified

Pranav = 7 $\quad 11/2 = 5 \quad 7 > 5$ ✓

> If we delete two distinct elements, majority won't change.

A[9] = 3 4 4 3 4 9 4 3 4

majority = ~~3~~ ~~4~~ ~~4~~ ~~4~~ 4

freq = ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ 1

A[11] = 3 3 4 6 4 3 2 5 3 3 3

majority = ~~3~~ ~~1~~ ~~2~~ 3

freq = ~~1~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~2~~ 3

A[5] = 1 2 3 4 5

majority = ~~1~~ ~~3~~ 5

NO MAJORITY

freq = ~~1~~ 0 ~~1~~ 0 1

3 3 5 5 5 5 5 7 9

majority = ~~3~~ 5

freq = ~~1~~ ~~2~~ ~~1~~ 0 ~~1~~ ~~2~~ ~~3~~ ~~2~~ 1

```
int majority( a[], n ) {
    am = a[0] , freq = 1
    for ( i = 1 to n-1 ) {
        if ( freq == 0 ) {
```

```
        ans = a[i] , freq=1
    }
    else if ( ans == a[i]) {
        ++freq
    }
    else {
        freq--
    }
}
//now either ans contains majority ele. OR
    there is no majority ele.

count=0
for (i=0 to n-1) {
    if (ans == a[i])
        ++count
}
if ( count > n/2)
    return ans
return NO_MAJORITY
}
```

Moore's Voting Algorithm

TC = O(N)

SC = O(1)

Question 3

Given 2D array, make all elements in a row & column zero if $a[i][j] = 0$ **O(1) space**

eg

$$\begin{matrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 0 \\ 4 & 2 & 0 & 4 \end{matrix} \Rightarrow \begin{matrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

Hint: All values are $>= 0$

Idea:

1. for each zero,

   replace all rows & col with -1

2. finally, convert all -1 to 0.

```
void convert( a[][], n, m){

    for( i=0 to n-1){
        flag=0
        for(j=0 to m-1){
            if(a[i][j] ==0){
                flag=1
                break
            }
        }
        if( flag ==1){
            for(j=0 to m-1){
                if(a[i][j] !=0)
                    a[i][j] = -1
            }
        }
    }

    for(j=0 to m-1){
        flag=0
        for(i=0 to n-1){
            if(a[i][j] ==0){
                flag=1
                break
            }
        }
```

$N \times (M+M)$

$\Rightarrow O(NM)$

$M \times (N+N)$

$\Rightarrow O(NM)$

```
if ( flag == 1 ) {
    for ( i=0 to n-1 ) {
        if (a[i][j] != 0)
            a[i][j] = -1
    }
}
}

for ( i=0 to n-1 ) {
    for (j=0 to m-1 ) {
        if (a[i][j] == -1)
            a[i][j] = 0
    }
}
}
```

TC = O(NM)

SC = O(1)

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

→

1	2	3	4
-1	-1	-1	0
-1	-1	0	-1

↓

1	2	-1	-1
-1	-1	-1	0
-1	-1	0	-1

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

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

$\rightarrow$

0	2	3	0
0	0	0	0
0	0	0	0