18/10/2023

Interview Questions

$10^3 \approx 2^{10}$

1 sec ⟶ $10^8 - 10^9$ iterations

$N <= 10^6$

$O(N)$ ✓

$O(N^2)$ X

$O(N \log_2 N)$ ✓ $\Rightarrow 10^6 \times 20 \approx 2 \times 10^7$

$10^6 \log 10^6$

$\Rightarrow 10^6 * \log 10^3 \times 10^3$

$N <= 20$

$O(N)$ ✓

$O(N^2)$ ✓

$O(N^3)$ ✓

$O(2^N)$ ✓

$O(N!)$ x

$N <= 10^{10}$

$O(N)$ X

$O(\log N)$ ✓

$O(\sqrt{N})$ ✓

$$\left.\begin{array}{l} 2^{10} \approx 1024 \\ 10^3 = 1000 \end{array}\right\}$$

1B → KB → MB → GB → TB . . . .

$\hookrightarrow 10^3 \hookrightarrow 10^3 \hookrightarrow 10^3 \hookrightarrow 10^3$

Amazon
Adobe
MSFT

Q Given an array of 1 & 0. We can replace one of the 0 with a 1. Return the count of max consecutive 1s in the array.

Ex → 1 1 0 1 1 0 1 1 1 , Ans = 6
0 1 2 3 4 5 6 7 8

2 +1+ 2 + 1 + 3
5   6

Ex → 0 1 1 1 0 1 1 0 1 1 0 , Ans = 6
0 1 2 3 4 5 6 7 8 9 10

1 1 1 0 1 1 0 1 1 1 1 0 0 1 1 0 1 1

For every 0 in the array:-

- Count no of consecutive 1s on left → l
- Count no of consecutive 1s on right → r
- if (l+r+1 > ans) { ans = l+r+1 }

Edge case :- If all are 1 (no 0s)

1 1 1 1 1 1 , ans = len(A).

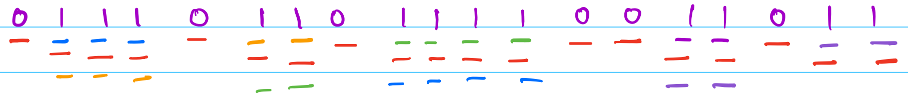
0 0 0 0 0 ✓

```
                                        for (j = i-1 ; j >= 0; j--)

for i → 0 to (n-1) {

    if (A[i] == 0) {
        for j = i-1 to 0
            if (A[j] == 0) {
                break;             } l
        }
        for j → i+1 to (n-1)
            if (A[j] == 0)
                break;             } r
        ans = max (ans , (r+l+1));
    }

}
```



every element is getting accessed at max 3 times.

$\therefore$ # iterations → 3N

TC: O(N)

Q Given an array of 1 & 0. We can swap one of 0 with a 1. Return count of max consecutive 1s in the array.

Ex → 1 1 0 1 1 0 1 1 1

0 1 2 3 4 5 6 7 8

1 1 0 1 1 1

Calculate total count of 1s.

For every 0 in the array:-

- Count no of consecutive 1s on left → l
- Count no of consecutive 1s on right → r
- $$count = \begin{cases} l+r & \text{if } (l+r) == \text{total count of 1s} \\ l+r+1 & \text{if } (l+r) < \text{total count of 1s.} \end{cases}$$

ans = max (ans, count).

Q $0\ \ \underline{1}\ \underline{1}\ \underline{1}\ \ 0\ \ \underline{1}\ \underline{1}\ \underline{1}$

Q Majority element.

Given an array of +ve nos. Return if there exists an element with frequency $> N/2$ ($N \to$ length of the array), SC: O(1), TC: O(N)

Eg 1 6 1 1 2 1 $\longrightarrow$ 1 , N = 6 , $|ME| > N/2 \Rightarrow 3$

A $\to$ 3 4 3 6 1 3 2 5 3 3 3 , N = 11 , $|ME| > 11/2$

$\longrightarrow$ 3

$|ME| > 5$

A $\to$ 4 6 5 3 4 5 6 4 4 4 , N = 10

$\longrightarrow$ X

$|ME| > 5$

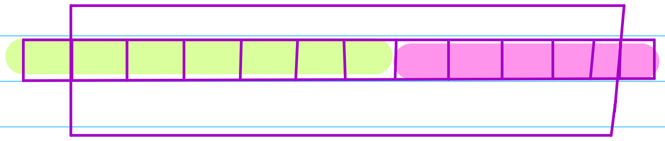
majority element, $> N/2$

$M1 + M2 \Rightarrow$

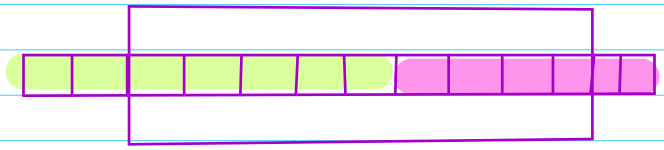
$> N/2 + > N/2 \Rightarrow > N$

There can only be 1 majority element in the array.

$N=13, ME > 6$



$N=11$, $|ME| > 11/_2$ (5)



$N=9$, $|ME| > 9/_2$ (4)

If we remove any 2 distinct elements, the majority element remains the same.

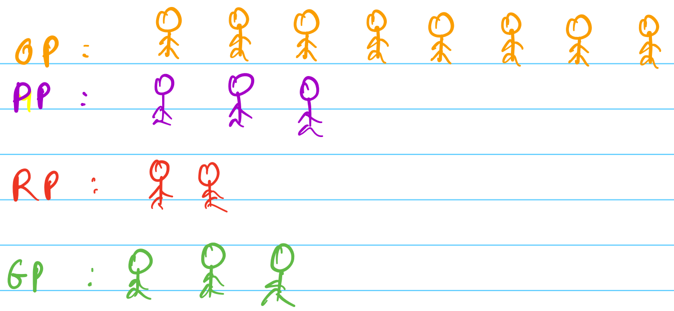
3 ~~4~~ ~~3~~ ~~6~~ ~~1~~ ~~3~~ ~~2~~ ~~5~~ ~~3~~ 3 3

1 ~~6~~ ~~1~~ 1 ~~2~~ ~~1~~

~~4~~ ~~6~~ ~~5~~ ~~8~~ ~~4~~ ~~5~~ ~~6~~ ~~4~~ 4 4

A	B

C = ~~0~~ ~~1~~ ~~2~~ ~~1~~ ~~2~~ 1



Winner : ~~PP~~ OP
~~1~~ ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~2~~ ~~1~~ ~~2~~ ~~1~~ 0

3 4 3 6 1 3 2 5 3 3 3

ME 3 1 2 3

count ~~1~~ ~~0~~ ~~1~~ 0 ~~1~~ 0 ~~1~~ 0 ~~1~~ ~~2~~ 3

Moores Voting algorithm.

1 2 1 4 1 5 1

ME 1

count ~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ ~~0~~ 1

4 6 5 3 4 5 6 4 4 4

ME 4 5 4 6 4

count ~~1~~ 0 ~~1~~ 0 ~~1~~ 0 ~~1~~ 0 ~~1~~ 2

~~1~~ ~~2~~ ~~1~~ ~~2~~ ~~1~~ ~~2~~ ③

1 3

~~1~~ ~~0~~ ~~1~~ ~~0~~ ~~1~~ 0 · ~~1~~

TC : O(N)

Meet at 10:55 pm.

Q You are given a 2D integer matrix A, make all elements in a row or column zero if A[i][j] = 0 (make i^{th} row & j^{th} col^n as 0) (+ve integers)

$$\begin{matrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 0 \\ 9 & 2 & 0 & 4 \end{matrix} \Rightarrow \begin{matrix} 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 1 & 2 & 3 & 4 \\ 5 & 0 & 7 & 0 \\ 9 & 2 & 0 & 4 \end{matrix} \Rightarrow \begin{matrix} 1 & 2 & 3 & 4 \\ -1 & 0 & -1 & 0 \\ -1 & -1 & 0 & -1 \end{matrix} \Rightarrow \begin{matrix} 1 & -1 & -1 & -1 \\ -1 & 0 & -1 & 0 \\ -1 & -1 & 0 & -1 \end{matrix}$$

$$\Downarrow$$

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$$

```
for i→ 0 to (N-1)
    for j→0 to (M-1)
        if (A[i][j] == 0) {

            for (K = 0 to (M-1))
                if (A[i][j] != 0)
                    A[i][j] -1
}
```

```
for  i → 0  to (n-1) {
      int  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

      }
}
```

N*M + N*M

O(N*M)

SC:O(1)

rows

```
for  j → 0  to  (M-1) {
      flag =0

      for  i → 0  to  (N-1)
           if ( A[i][j] ==0)
                  flg =1, break;

      if (flag = 1)
         for (j → 0 to (N-1))
              if (A[i][j] != 0)
                   A[i][j] = -1;
```