Q. MAJORITY ELEMENT

Google Amazon MS|GS| Directi, Flipkart ....

Given an array of size N, Return if there exists a majority element.

↳ freq > $\frac{N}{2}$

SC : O(1)

Ex :- A : 1, 6, 1, 1, 2, 1

ME = 1

N = 6

freq > $\frac{6}{2}$

Minfreq = 4

Quiz A : 3, 4, 3, 6, 1, 3, 2, 5, 3, 3, 3

ME = 3.

N = 11

Minfreq = 6

Quiz A : 4, 6, 5, 3, 4, 5, 6, 4, 4, 4

freq(4) = 5 > 5 ✗

N = 10

minfreq = 6

ND M.E

A: 4, 6, 5, 3, 4, 5, 6, 4, 4, 4

i

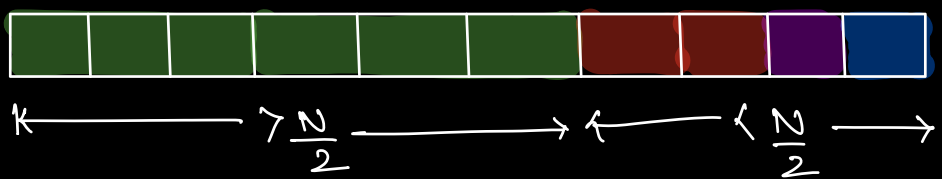
i

```
int majority (int A[], int N) {
    for(i = 0; i < N; i++) {
        freq = 0
        for(j = 0; j < N; j++) {
            if(a[i] == a[j])
                freq++;
        }
    }
    if (freq > N/2)
        return a[i];
}
return -1;
}
```

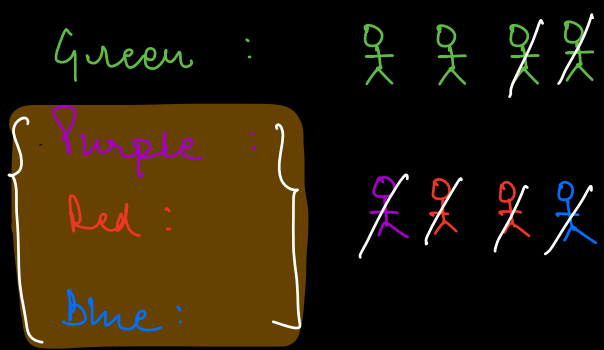
TC: $O(N^2)$
SC: $O(1)$

Quiz At max how many M.E can be there in an Array ?

min freq = 6



⇒ There can be a single M.E only.



N	MEI
10	6
8	5
6	4
4	3
2	2

Obs :- If 1 majority & 1 non majority elements are fighting then ME won't change.

Obs: If 2 non ME's are fighting then ME won't change.

A: 3, 4, 3, 6, 1, 3, 2, 5, 3, 3, 3 (indices 0 to 10)

N = 11

freq > $\frac{11}{2} = 5$

ME = 3

N	|ME|
11	6
↓	↓
9	5
↓	↓
7	4
↓	↓
5	4

A: 3, 4, 3, 6, 1, 3, 2, 5, 3, 3, 3 (indices 0 to 10) ↑

ME = ~~3~~ ~~3~~ ~~4~~ ~~2~~ 3

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

Ex :-

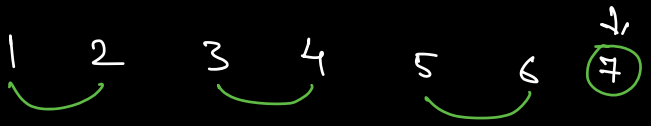
A : 3 3 4 2 4 4 2 4 4 ↓ N = 9

ME = ~~3~~ 4

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

freq(4) = 5 > $\frac{9}{2}$

Ex



ME = ~~1 3 5~~ 7

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

Ex

1 2 3 4 ↓

ME = ~~1 3~~

f = ~~1 0 1~~ 0

Ex 1 1 2 3 ↓

ME = ~~1~~

f = ~~1 2 1~~ 0

MOORE'S VOTING ALGORITHM

TC : O(N)

SC : O(1)

Code

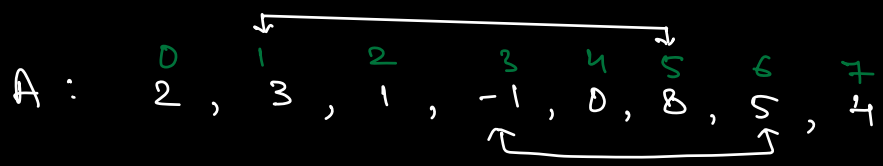
```
int me = a[0]
freq = 1
for( i = 1 ; i < N; i++) {
    if(a[i] == me){
        freq++
    }
    else {
        if( freq == 0){
            me = a[i];
            freq = 1;
        }
        else {
            freq--;
        }
    }
}
count = 0
for (i = 0; i < N; i++) {
    if(a[i] == me)
        count++
}
if (count > N/2)
    return me;
return -1;
```

Q: Given an array & Q queries

Direct-i

s, e, O ⇒ Sum of all Odd indexed elements from [s, e]

s, e, E ⇒ Sum of all even indexed elements from [s, e]



Q: 2

s	e	O/E	
3	6	O	⇒ -1 + 8 = 7
1	5	E	⇒ 1 + 0 = 1

Ex

	0	1	2	3	4	5
A :	2	3	1	6	4	5
PS :	2	5	6	12	16	21
PS_E :	2	2	3	3	7	7
PS_O :	0	3	3	9	9	14

$$PS_E[0] = A[0]$$

$$PS_E[i] = \begin{cases} PS_E[i-1] + A[i] & i\%2 == 0 \\ PS_E[i-1] & \text{else} \end{cases}$$

$$PS_0[0] = 0$$

$$PS_0[i] = \begin{cases} PS_0[i-1] + A[i] & i\%.2 \;!= 0 \\ PS[i-1] & \text{else} \end{cases}$$

Quiz

	0	1	2	3	4
A :	2	4	3	1	5
PS_0 :	0	4	4	5	5

Quiz

	0	1	2	3	4	5	6
A :	4	1	0	-2	3	2	5
PS_E :	4	4	4	4	7	7	12

Sum of **ODD** indexed elements from s to e ⇒

$PS_0[e] - PS_0[s-1]$

Sum of **even** indexed elements from s to e ⇒

$PS_E[e] - PS_E[s-1]$

TC : $O(N) + O(N) + O(Q) : O(N+Q)$

SC : $O(N)$

Q: Given an Array, count the no. of Special Indexes in the Array.

Amazon
Directi
CodeNation
Google

Special Index: An index after removing which

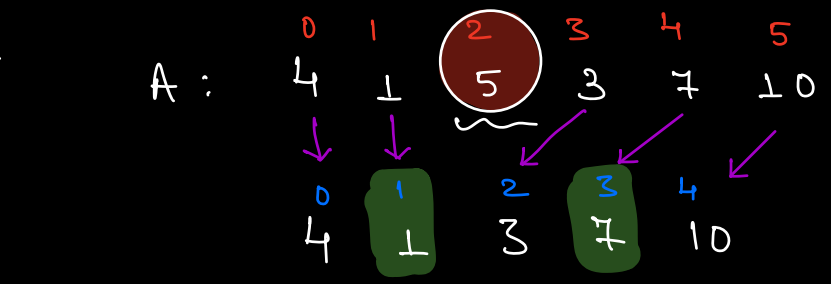
Sum of all ODD indexed elements = Sum of all even indexed elements

in the resultant array.

	0	1	2	3	4	5
A:	4	3	2	7	6	-2

i	[]	S_E	S_O	
0	3 2 7 6 -2	8	8	✓
1	4 2 7 6 -2	9	8	X
2	4 3 7 6 -2	9	9	✓
3	4 3 2 6 -2	4	9	X
⋮	⋮	⋮	⋮	

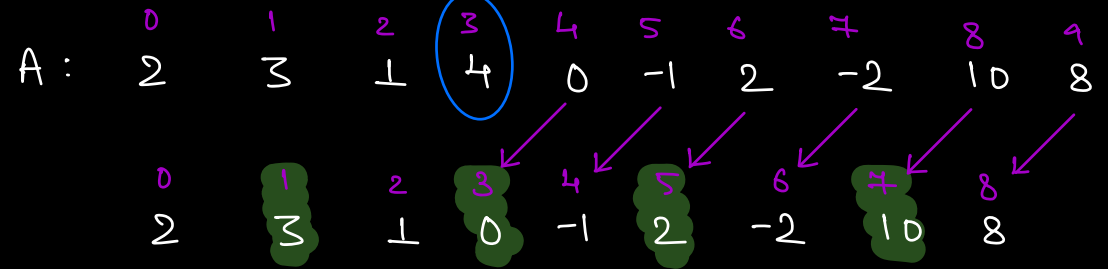
Quiz



$S_0 = 8 = S_0[0-1] + S_E[3-5]$

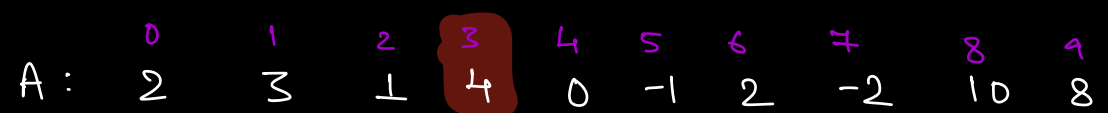
$= 1 + 7 = 8$

Quiz



$S_0 = S_0[0-2] + S_E[4-9]$

Quiz



$S_E = S_E[0-2] + S_0[4-9]$

$= 3 + 5 = 8$

Sum of ODD indexed elements after removing index = i

$$S_O = S_O[0, i-1] + S_E[i+1, N-1]$$
$$= PS_O[i-1] + (PS_E[N-1] - PS_E[i])$$

Sum of Even indexed elements after removing index = i

$$S_E = S_E[0, i-1] + S_O[i+1, N-1]$$
$$= PS_E[i-1] + PS_O[N-1] - PS_O[i]$$

$$\boxed{Sum[s,e] = PS[e] - PS[s-1]}$$

i = 0

$$S_O = S_E[1, N-1]$$
$$\Rightarrow S_O = PS_E[N-1] - PS_E[0]$$

$$\Rightarrow S_E = S_O[1, N-1]$$
$$S_E = PS_O[N-1] - PS_O[0]$$

i = N-1

$$S_O = S_O[0, N-2] = PS_O[N-2]$$
$$S_E = S_E[0, N-2] = PS_E[N-2]$$

1. Build PS_O
2. Build PS_E
3. for (i = 0; i < N; i++) {
   if ($S_E == S_O$)
   count++
   }
   return count;

TC: O(N)
SC: O(N)

———— * ————