Today's Quote :-

With my desire to improve everything, I destroy the moment.

↓

Superman Syndrome.

Ques) Given an array of size n, asked q queries :-

Each query can be of 2 types.

if type 1 → sum of all even indices b/w

if type 2 → sum of all odd indices b/w

0	1	2	3	4	5	6	7
2	3	1	-1	0	8	5	4

s	e	type	
3	6	1	→ 5
1	5	2	→ 10

Brute force :-

Nothing but traversal, considering the right indices.

T.C → (q*n)

(*) Optimize it

	0	1	2	3	4	5	6	7
	2	3	1	-1	0	8	5	4
pfeven →	2	2	3	3	3	3	8	8
pfodd →	0	3	3	2	2	10	10	14

s	e	type		
3	6	1	→ 5	pfeven[e] - pfeven[s-1]
1	5	2	→ 10	pfodd[e] - pfodd[s-1]

```
pfeven[0] = arr[0].
for (i=1; i<n; i++) {
        if (i is even) {
            pfeven[i] = pfeven[i-1] + A[i]
        } else {
            pfeven[i] = pfeven[i-1];
        }
}

pfodd[0] = 0;

for (i=1; i<n; i++) {
        if (i is odd ) {
            pfodd[i] = pfodd[i-1] + A[i]
        } else {
            pfodd[i] = pfodd[i-1];
        }
}
```

T.C → O(n)
S.C → O(n)

Ques) Given an array of size N, count no of special idx.

↓

if we delete that idx
sum of all odd idx = sum of all even idx

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

→ Ans = 2

Delete idx		Seven	Sodd
0	0 1 2 3 4 3 2 7 6 -2	8	8
1	0 1 2 3 4 4 2 7 6 -2	9	8
2	0 1 2 3 4 4 3 7 6 -2	9	9
3	0 1 2 3 4 4 3 2 6 -2	4	9
4	0 1 2 3 4 4 3 2 7 -2	4	10
5	0 1 2 3 4 4 3 2 7 6	12	10

0 i-1 i i+1 N-1

Before deletion

x = pfodd[i-1] → sum of all odd indices to the left of i

y = pfeven[i-1] → sum of all even idx to the left of i

z = pfodd[n-1] - pfodd[i], sum of all odd idx elements to right

of i

w = pfeven[n-1] - pfeven[i], sum of all even idx elements to right of i

After deletion :-

Sum of all odd elements = x + w

Sum of all even elements = y + z

for i to be special idx :-

x + w = y + z .

// pseudocode :-

// create pfeven & pfodd .

```
int c = 0;
for (i=0; i<n; i++){
    // i is special index or not
    int sumeven = pfodd[n-1] - pfodd[i]
    if (i!=0) {
        sumEven += pfeven[i-1]
    }
    int sum odd = pfodd[i-1] +
                  pfeven[n-1] - pfeven[i]
```

```
if (sumeven == sumodd) { C++; }
```

T.C → O(N)
S.C → O(N)

Break 9:52 - 10:02 pm.

(*) → Majority Element :-

Given an arr[], return if there exists a no., with freq > $\frac{N}{2}$.

Ex1) arr[6] = {1, 6, 1, 1, 2, 1 3}, freq > $\frac{6}{2}$

Ex2) arr[] = {3, 4, 3, 6, 1, 3, 2, 5, 3, 3, 3}
freq > $\frac{11}{2}$

Ex 3) arr[] = {4, 6, 5, 3, 4, 5, 6, 4, 4, 4} freq > $\frac{10}{2}$
↓
No majority element.

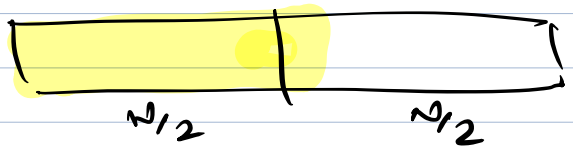
Brute force :-

① → for all elements calc freq.
T.C → O(N²)
S.C → O(1)

② HashMap :- T.C → O(N)
S.C → O(N)

③ sorting :- T.C → O(NlogN)

⊛ No. of Majority elements at most



① Only 1 Maj. element is possible.

Run over even elections :-

Aman :-
Bibi :-
Kumar :-
Jay :-

N = 11 $\frac{N}{2} \Rightarrow 5$ atleast 6 needy.

Total	Majority
11	6
9	5
7	4
5	3
3	2

Aman :-
Bibi :-
Kumar :-
Jay :-

Conclusion:- If two distinct elements are removed, the majority remains same.

0	1	2	3	4	5	6	7	8	9	10
3	4	3	6	1	3	2	5	3	3	3

may :- 3 x x 3
freq :- 1 0 1 0 1 0 1 0 1 2 3

4 4 5 6 7 4 4 4 4 7 8

may : 4 x 4
freq : 1 2 1 0 1 0 1 2 3 2 1

1 2 3 4 5

may = 1 x 5
freq : 1 0 1 0 1

2 2 2 2 4 4 4

May :- 2
freq: 1 2 3 4 3 2 1

1, 1, 1, 2, 2, 2, 3

May → x 3
freq → 3 0 1

Moore's Voting Algorithm

```
int maj= arr[0]
freq= 1

for (i=1; i<n; i++){
    if (arr[i]==maj){
        freq++;
    } else if (freq==0){
        maj= arr[i];
        freq= 1
    } else{
        freq--;
    }
}

return maj;
```

T.C → O(N)
S.C → O(1)

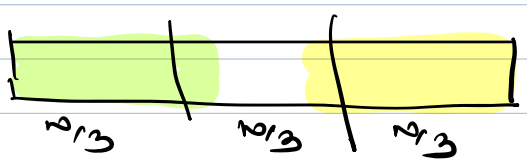
→ assuming we always have a majority element, otherwise check.

→ Majority element (n/3)

→ iterate it once.

to be Majority freq > n/3.

(10, 10, 20, 30, 10, 10) → 6/3 → 2



At max two majority elements can be there.

Majority	Majority elements	how many elements can delete so majority remains here
$n/2$	1	2
$n/3$	2	3

HINT

10, 10, 20, 30, 10, 10

may 1 → 10 may 2 → 20

freq1 → ✗ ✗ ✗ ✗ 3 freq2 → ✗ 0

→ 10, 10, 20, 30, 10, 10, 30

may1 → 10 may2 → 20 30

freq1 → ✗ ✗ ✗ ✗ 3 freq2 → ✗ 0 1

Puzzle :-

5 pirates and 100 gold coins :-

A, B, C, D, E → Proposal.

↳ if 50% people vote in your favour, proposal is accepted.

→ ~~A~~ ~~B~~ C, D, E

D → 100, E → 0

→ ~~A~~ ~~B~~ C D E

C → 99, D → 0, E → 1

→ ~~A~~ B C D E

B → 99, C → 0, D → 1, E → 0

→ A B C D E

A → 98, B → 0, C → 1, D → 0, E → 1

Poison & Rat

1000 wine Bottles.
1 bottle is poisoned.
A rat dies 1 hour after drinking poisoned wine.

you have 1 hour, how many (min) rats you need to figure out poisoned bottle.

Ans → 10.

1 10 11 100 101 110 111 1000

1 2 3 4 5 6 7 8

996 997 998 999
1000

10 rats.

0 0 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 8 9

→ 2, 4, 6 rat dies.

6 5 4 3 2 1 0
1 0 1 0 1 0 0 → 84.

10 bits $2^{10} \rightarrow 1024$.

Contest → Revise all classes
↓
Assignments & H.w