

Array: Important Problems - II

Question 1

Given an array of size N and Q queries of $[s, e]$, for every query return the sum of all even indexed element in the range from s to e .

eg

2	3	1	6	4	5
0	1	2	3	4	5

$Q=4$

s	e	sum
1	3	1
2	5	$1+4=5$
0	4	$2+1+4=7$
3	3	0 [No index]

Idea 1: for each query, [TODO coding part]
iterate from s to e and sum even-indexed elements.

TC: $O(N*Q)$

SC: $O(1)$

Idea 2 : Prefix Sum

$$pf[i] = \underbrace{a[0] + a[1] + \dots + a[i-1]}_{\downarrow} + a[i]$$

$$pf[i] = pf[i-1] + a[i]$$

$pfe \rightarrow$ prefix sum of even-indexed elements

$$pfe[i] = pf[i-1] + a[i] \quad (\text{if } i/2 == 0)$$

$$pfe[i] = pf[i-1] \quad (\text{if } i/2 == 1)$$

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

pf : 2 5 6 12 16 21

pfe : 2 2 3 3 7 7

Code

$pfe[n]$

$pfe[0] = a[0]$

for ($i=1; i < n; ++i$) $\}$ $\rightarrow N$ iterations

if ($i/2 == 0$)

$pfe[i] = pf[i-1] + a[i]$

else

$pfe[i] = pf[i-1]$

TC: $O(1 + N)$

SC: $O(N)$

for (i=0; i<8; ++i) { \rightarrow 8 iterations

// s, e index

if (s==20)

print (pfe [e])

else

print (pfe [e] - pfe [s-1])

}

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

pfo : 0 4 4 5 5

\hookrightarrow prefix sum of odd-indexed elements

pfo[0] = 0

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

if (i%2 == 1)

pfo[i] = pfo[i-1] + a[i]

else

pfo[i] = pfo[i-1]

}

Question 2 - Special Index

Given an array. Count no. of special index in the array.

An index is special, if after removing the index

$$\text{sum of all even-indexed elements} = \text{sum of all odd-indexed elements}$$

eg A : $\begin{matrix} 4 & 3 & 2 & 7 & 6 & -2 \\ 0 & 1 & 2 & 3 & 4 & 5 \end{matrix}$

i	0	1	2	3	4	Se	So	
0	3	2	7	6	-2	8	8	✓
1	4	2	7	6	-2	9	8	✗
2	4	3	7	6	-2	9	9	✓
3	4	3	2	6	-2	4	9	✗

Idea 1: for each index

create a new array after removing the index, calculate se & so and compare.

Code

```
int specialCount (arr) {
```

```
    ans = 0
```

```
    for (i = 0; i < n; ++i) {
```

```
        // delete ith index
```

```
        // arr : a0 a1 ... ai-1 ai ai+1 ... an-1
```

```
        // create new array w/o ai
```

```
        copy[m-1] = [ ]
```

TODO
TC: O(N)

```
        iterate on copy and calculate se & so } TODO  
                                           TC: O(N)
```

```
        if (se == so)
            ++ans
```

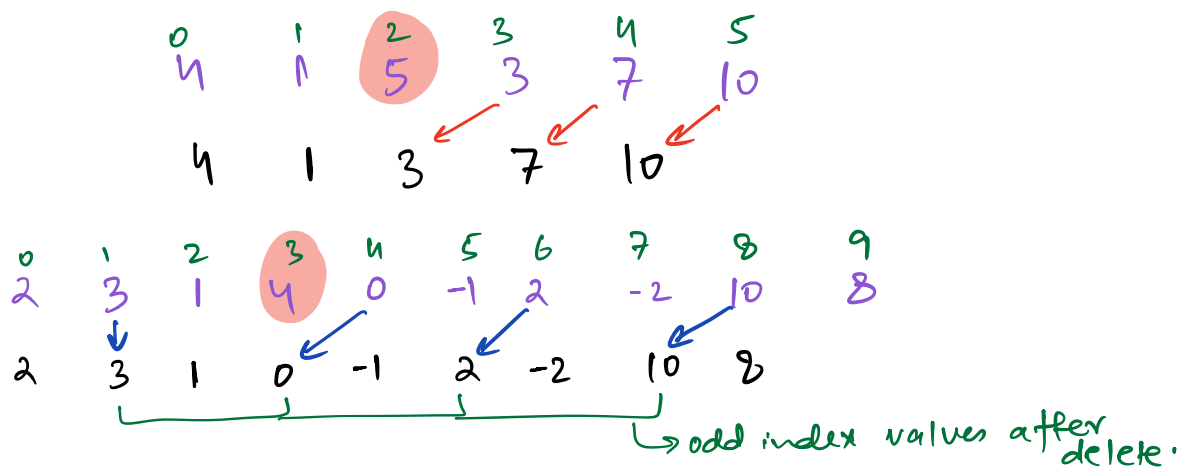
```
    }
    return ans
```

```
}
```

TC: O(N²)

SL: O(N)

Observation Time



sum of odd-indexed elements after removing
index 3 =

sum of odd-indexed elements from [0-2]

3

+

sum of even-indexed elements from [4-9]

$0 + 2 + 10 = 12$

in the original array

sum of even-indexed elements after removing
index 3 =

sum of even-indexed elements from [0-2]

$2 + 1 = 3$

+

sum of odd-indexed elements from [4-9]

$-1 + -2 + 8 = 5$

in the original array

After removal of index i

$$S_e = S_e[0, i-1] + S_o[i+1, n-1]$$

$$S_o = S_o[0, i-1] + S_e[i+1, n-1]$$

We have already solved prefix sum for even and odd indexed elements.

$pfe \rightarrow$ prefix sum of even-indexed elements

$pfo \rightarrow$ " " " " odd-indexed " "

$$se[0, i-1] = pfe[i-1]$$

$$so[0, i-1] = pfo[i-1]$$

$$se[i+1, n-1] = pfe[n-1] - pfe[i]$$

$$so[i+1, n-1] = pfo[n-1] - pfo[i]$$

Code

Create $pfe[n]$ and $pfo[n] \rightarrow O(N)$ TODO

ans = 0

for ($i=0; i < n; i++$) {

// $se = pfe[i-1] + (pfo[n-1] - pfo[i])$

// $so = pfo[i-1] + (pfe[n-1] - pfe[i])$

if ($i == 0$) {

$se = pfo[n-1] - pfo[i]$

$so = pfe[n-1] - pfe[i]$

}

else {

TC: $O(N)$

SC: $O(N)$

```

        sc = pfeli-1 + pfo[n-1] - pfo[i]
        so = pfo[i-1] + pfe[n-1] - pfe[i]
    }
    if (sc == so)
        + ans
    }
    return ans
}

```

Question 3 - Majority Element

Given an array, return if there exists an element with frequency $> N/2$.
(N = length of array)

eg $A[6] = 1 \ 2 \ 1 \ 6 \ 1 \ 1$
 $\text{freq}(1) = 4$ $N/2 = 6/2 = 3$ $\Rightarrow 4 > 3$ YES

eg $A[9] = 3 \ 4 \ 4 \ 8 \ 4 \ 9 \ 4 \ 3 \ 4$
 $\text{freq}(4) = 5$ $N/2 = 9/2 = 4$ $\Rightarrow 5 > 4$ YES

$A[10] = 4 \ 6 \ 5 \ 3 \ 4 \ 5 \ 4 \ 4 \ 4 \ 8$
 $\text{freq}(4) = 5$ $N/2 = 10/2 = 5$ $\Rightarrow 5 \not> 5$ NO

Ideas: Count freq. of each element & compare with $N/2$.

1. Using 2 nested loops $\rightarrow TC: O(N^2)$ $SC: O(1)$

2. Hashmap / Dictionary $\rightarrow TC: O(N)$ $SC: O(N)$

3. Sort the array

Ignore if you don't understand
 $a[7] = \{1, 3, 1, 5, 3, 3, 5\}$ $TC: O(N \log N)$
 $sort\ a[7] = \{1, 1, 3, 3, 3, 5, 5\}$ $SC: O(1)$

At max how many majority elements can be there in an array? \Rightarrow Ans = 1

assume 2 majority elements x & y

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

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

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

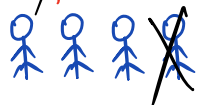
invalid because we have only N elements.

Election \rightarrow 15 MLAs

$$15/2 = 7$$

Amul \rightarrow  - 9

Sagar \rightarrow  - 2

Siddharth \rightarrow  = 4

Amul \rightarrow 9 $15/2 = 7$ $9 > 7$

\downarrow 2 disqualify


Amul \rightarrow 8 $13/2 = 6$ $8 > 6$

\downarrow 1 disqualify

Amul \rightarrow 7 $11/2 = 5$ $7 > 5$

Observation:

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

$A(9) =$ 
 $\text{freq}(4) = 5$ $9/2 = 4$ $5 > 4$

$A[11] =$ $\begin{array}{cccccccccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ \cancel{2} & \cancel{2} & \cancel{2} & \cancel{6} & \cancel{2} & \cancel{2} & \cancel{2} & \cancel{2} & 3 & 3 & \cancel{2} & \cancel{2} \end{array}$

$A[10] =$ $\begin{array}{cccccccc} \cancel{2} & \cancel{2} & \cancel{2} & \cancel{2} & \cancel{2} & \cancel{2} & \cancel{2} & \cancel{2} \end{array}$ NO MAJORITY

Implementation

1. Assume first element as majority element
2. If you get same element, increase freq.
3. If you get diff. element, decrease freq
4. If $\text{freq} = 0$, change majority element.

$A[11] =$ $\begin{array}{cccccccccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 3 & 4 & 6 & 1 & 3 & 2 & 5 & 3 & 3 & 3 \end{array}$

$\text{ele} = 3$ $\text{freq} = 1 \cancel{2} \cancel{1} 0$	$\text{ele} = 1$ $\text{freq} = \cancel{1} 0$	$\text{ele} = 2$ $\text{freq} = \cancel{1} 0$	$\text{ele} = 3$ $\text{freq} = \cancel{1} \cancel{2} 3$	$\text{freq}(3) = 6$ $11/2 = 5$
---	--	--	---	------------------------------------

$A[10] =$ $\begin{array}{cccccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 3 & 3 & 4 & 6 & 1 & 3 & 2 & 5 & 8 & 8 \end{array}$

$\text{ele} = 3$ $\text{freq} = 1 \cancel{2} \cancel{1} 0$	$\text{ele} = 1$ $\text{freq} = \cancel{1} 0$	$\text{ele} = 2$ $\text{freq} = \cancel{1} 0$	$\text{ele} = 8$ $\text{freq} = \cancel{1} 2$	$\text{freq}(8) = 2$ $10/2 = 5$
---	--	--	--	------------------------------------

If final element $\text{freq} \neq N/2$ then
NO MAJORITY

$A[10] =$

 \downarrow
 3
 \uparrow

 \downarrow
 3
 \uparrow

 \downarrow
 4
 \times

 \downarrow
 3
 \times

 \downarrow
 1
 \times

 \downarrow
 3
 \times

 \downarrow
 2
 \times

 \downarrow
 3
 \times

 \downarrow
 8
 \times

 \downarrow
 8
 \times

$ele = 3$
 $freq = 1 \times 1 \times 1 \times 1 \times 1 \times 1 \times 0$

$freq(3) = 5$
 $10/2 = 5$

NO MAJORITY

$A[9] =$

 \downarrow
 3
 \uparrow

 \downarrow
 3
 \uparrow

 \downarrow
 4
 \times

 \downarrow
 3
 \times

 \downarrow
 1
 \times

 \downarrow
 3
 \times

 \downarrow
 2
 \times

 \downarrow
 3
 \times

 \downarrow
 8
 \times

$ele = 3$
 $freq = 1 \times 1 \times 1 \times 1 \times 1 \times 1$

$freq(3) = 5$
 $9/2 = 4$

majority = 3

Code

```

int majority ( int a[] ) {
    n = a.length
    ele = a[0], freq = 1
    for ( i = 1; i < n; ++i ) {
        if ( freq == 0 ) {
            ele = a[i], freq = 1
        }
        else if ( ele == a[i] ) {
            freq++
        }
        else {
            freq--
        }
    }
}

```

TC : $O(N)$

SC : $O(1)$

```

c = 0
for (i = 0; i < n; ++i) {
    if (a[i] == ele)
        ++c
}
if (c > n/2) return ele
else return NO-MAJORITY
}

```

Moore's Voting Algo

Homework - majority element

Given array return if there exist an element with freq $> N/3$. SC: O(1)

Hint: Previously we deleted 2 different elements

Here, you have to delete 3 different elements

So, you have to keep 2 variables: ele_1 & ele_2 .