Q2. Given an array of N elements, check if there exists an ele with frequency > N/4

* Can't take extra space

Eg arr[6]

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

$\frac{N}{4} \mid \frac{6}{4} = 1.5$

present atleast twice

Ans → (5), (1)

arr[8]

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

$\frac{N}{4} \mid \frac{8}{4} = 2$ present atleast 3 times

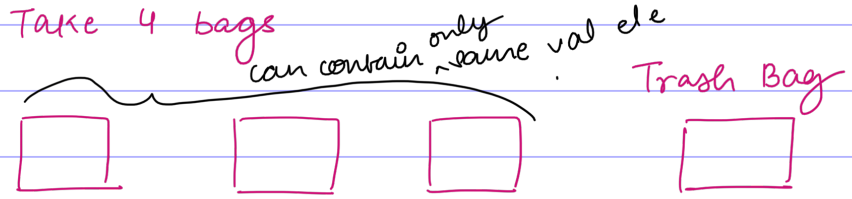
No such ele.

Other solutions

(1) Check every other ele — $O(n^2)$

(2) Sort — $O(n \log n)$

Algorithm:

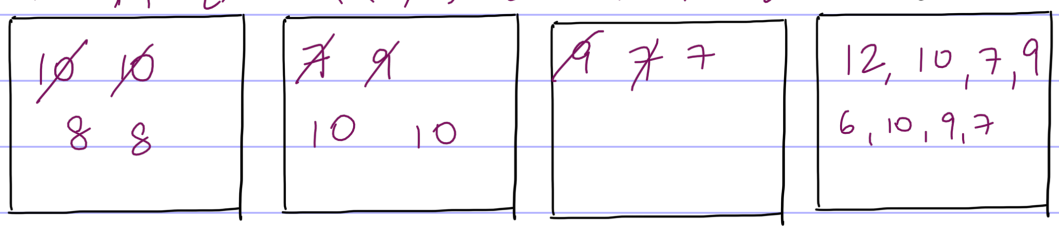


(1) When ele x comes, if a bag contains ele x, add x to that bag

(2) If no bag contains x but if empty bag is present, add x to that x

(3) If no bag contains x, if no empty bag present, throw x is dustbin & take one occurrence from each & throw away.

	0	1	2	3	4	5	6	7	8	9	10	11	12
ar[13]	10	10	7	9	12	9	7	7	6	8	10	8	10



$> \frac{13}{4}$

> 3

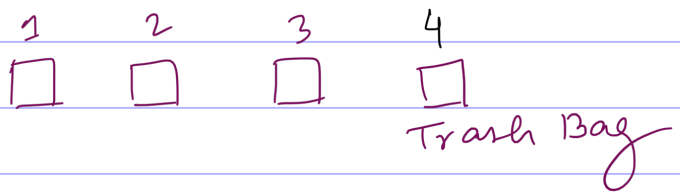
freq(10) >= 4 ✓

freq(8) >= 4 X

* Algo claims one of the bag ele will be your answer

Proof of Concept [Boyer Moore Counting Algo]

Given N ele, check if there exists an ele with freq $> N/4$



Say 'x' is in none of the bags, it'll be in the trash bag.

freq(x) → m

when a unit of x goes down in trash, it takes 3 other elements with itself

At every pt.,

$4m <= N$

$m <= N/4$

max $m = N$

K+1

conclusion: Any ele in a trash bag can't have freq $> \frac{N}{4}$

Now, if one answer exists, where will it be? In other bags.

Check for every bag, if it's answer or not.

Proved.

Pseudocode

```
pair<int, int> a, b, c;
a.second = b.second = c.second = 0
for(int i=0; i<N; i++){
    if(a.second != 0 && A[i] == a.first)
        a.second += 1
    else if(b.second != 0 && A[i] == b.first)
        b.second += 1
    else if(b.second != 0 && A[i] == b.first)
        b.second += 1
    else if(a.second == 0) a.first = A[i]; a.second = 1
    else if(b.second == 0) b.first = A[i]; b.second = 1
    else if(c.second == 0) c.first = A[i]; c.second = 1
    else {
        a.second--; b.second--; c.second--
    }
```

Que:

construct array

Given AP, first term a & common difference d
what will be the first 10 elements of it?

$a, a+d, a+2d, a+3d, \ldots a+9d$

If n ele? →

$a, a+d, a+2d, \ldots a+(n-1)d$

Given 2nd & 5th ele of AP
find d & a?

2nd → 10 → $a+d = 10$

5th → 25 → $a+4d = 25$

$(a+4d) - (a+d) = 25-10$

$3d = 15$

$\| d = 5 \|$

Given 3rd (→ a+2d) & 10th (→ a+9d) of AP
a & d?

3rd → 24 → $a+2d = 24$

10th → 45 → $a+9d = 45$

$-7d = -21$

$\| d = 3 \|$

Given 5 elements in AP & 2 ele X & Y in AP
nonce can belong to any position

conditions
1) $X < Y$
2) All AP nos. are vet integers
3) Common difference > 0

$N = 5$

1st	2nd	3rd	4th	5th
X	B	B	B	B

$x < y$ $\quad n = 5$

20 $\quad$ 50

a

x	y	y − x	d	a	a+4d
1	2(a+d)	a+d−a = 50−20	30	20	140
1	3 (a+2d)	a+2d−a = 50−20	15	20	95
1	4 (a+3d)	a+3d−a = 50−20	10	20	90
1	5 (a+4d)	a+4d − a = 30	7.5 ✗	—	—

a+d

x	y	y − x	d	a	a+4d
2	3	a+2d−a−d=30	30	−10	—
2	4	a+3d−a−d=30	15	5	5+60=65
2	5	a+4d−a−d=30	10	10	10+40=50

a+2d

x	y	y − x	d	a	a+4d
3	4	a+3d−a−2d=30	30	−40	—
3	5	a+4d−a−2d=30	15	−10	—

a+3d

4	5

Pseudocode

values → X & Y

```
for( i = 1  to  N ) {
    for( j = i+1  to  N ) {

        Y - X = d
        = a + (j-1)d - [a + (i-1)d]
        = a + jd - d - a - id + d
        = (j - i) d

        || d = (Y - X) / (j - i) ||

        // check if d is valid
        if( (y-x) % (j-i) == 0 ) {
            // a = x - (i-1) d      // check a > 0
            // an = a + (n-1) d

            // last should be min
            // if last & an are same, then first
                should be min
        }
    }
    return first & last
}
```

$$\left[\begin{array}{l} X = a + (i-1)d \\ Y = a + (j-1)d \end{array}\right]$$

first	last

Time $O(n^2)$

Space $O(1)$

Que: Minimize Difference

Given an array of integers of size N.
Minimize absolute difference b/w max & min ele of the array.
You can perform 2 types of operations almost B times

1) Increment A[i] → A[i] + 1
2) Decrement A[i] → A[i] - 1

Return min possible difference

Eg A = [2, 6, 3, 9, 8]
B = 3

Ans → 2 → 2+1 = 3

3, 6, 3, 9, 8

9 → 9-1 = 8

3, 6, 3, 8, 8

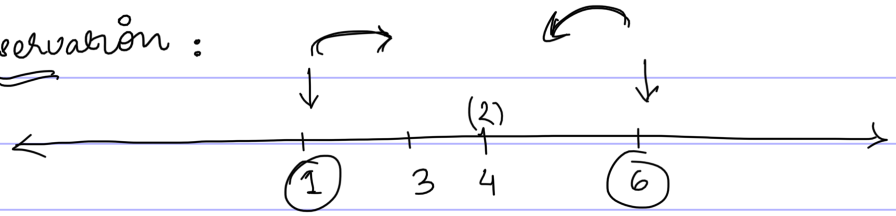
Ans → 8 - 3 = 5

Eg → [4, 6, 3, 1, 4]
B → 5

Ans → 1.

[4, 6, 3, 3, 4]

Observation :



I can't blindly pick any (either min / max) & do relevant operation

why is that?

Say K = 3

1 → 1 + 1 = 2

2 → 2 + 1 = 3

3 → 3 + 1 = 4

Given [4, 6, 3, 1, 4]

Final array → [4, 6, 3, 4, 4]

Ans → 3

K = 1

Could have I done better?

1 → 1 + 1 = 2

2 → 2 + 1 = 3

6 → 6 - 1 = 5

[4, 5, 3, 3, 4]

Ans → 2

Observation : Frequency of min & max de matters

If say there are two elements "3" & if 3 is min & K = 1

Now it won't matter if I increase just a single 3.

Therefore, we should deal with the freq of min or max ele as a whole.

$K=3$

$A = [3 \quad 5 \quad 3 \quad 9 \quad 7 \quad 2 \quad 5]$

frequency array !

- map
- array

we pick the max ele of the array

create array of size max + 1

max → 9.

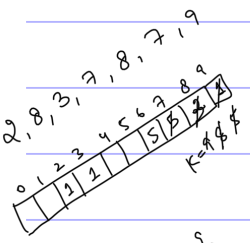
int freq[10]

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

i ↑ (index 2) j ↑ (index 9)

// find max — O(n)

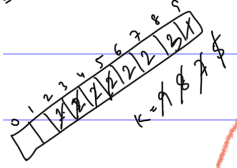
// find min — O(n)



// create freq array of size max + 1

// fill freq array — O(n)

i = min j = max



```
while(i<j)
    if ( freq[i] > freq[j]) {
        if ( freq[j] <= K) {
            freq[j-1] = freq[j-1] + freq[j]
            K = K - freq[j]
            j--
        } else {
            break
        }
```

O(max - min)

```
    }
    else {
        if ( freq[i] <= K) {
            freq[i+1] = freq[i+1] + freq[i]
            K = K - freq[i]
            i++
        }
        else break
    }
}
return j-i
```

Time complexity → O(max-min) + O(n)

Space Complexity → O(max)