Majority Element ($> \lfloor \frac{n}{2} \rfloor$ times)

$n = 9 \Rightarrow \frac{n}{2} = \frac{9}{2} = 4$

arr

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

- **Brute Force :-**

In this approach we ~~compare~~ start from index 0 and every element with every other element and in this process we keep on increasing count by 1 whenever that element is repeated. As soon as we get count $> \frac{n}{2}$, we return the element.

$T = O(n^2)$

$S = O(1)$

- **Better Approach :-**

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

HashMap<Integer, Integer>

(4, 1)	
~~(6,1)~~	(6,2)
~~(7,1)~~	~~(7,2)~~ ~~(7,3)~~ ~~(7,4)~~ (7,5)
(8, 1)	

Now, we, can iterate over the hashmap an check which element has count $> \frac{n}{2}$.

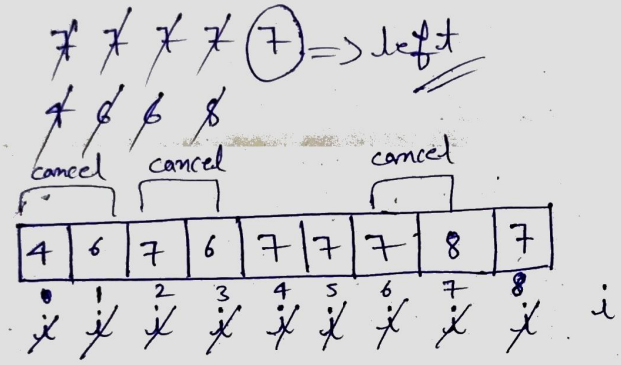
So, $T = O(n + n) = O(2n) \approx O(n)$

We can optimize this approach. When we will iterate the array and increase the count of elements. After increasing the count, there itself we can check if count > n/2.

So, with this approach, T = O(n)

S = O(n)

- **Best Approach :-** (Moore's Voting Algorithm) or Cancel Out Algorithm



element = -∞ // Anything which is not in the array

count = 0 1 0 1 0 1 2 1 2 1 3

So, here, element = 7 is the answer

Note: count can be anything depending on order of elements.

T = O(n)

S = O(1)