Finding Majority Element using Linear Search

element should occur (n/2)+1 times. As the array is sorted we can check at index i and (i+(n/2)). If they are same then that is majority element. if n = 7. majority element should occur 4 times . 1 is at index 0 and 3 . therefore majority element = 1 [1,1,1,1,2,3,4] Complexity: Time: O(n/2) = O(n) Space: O(1)

```
In [13]: def findMajorityele(arr):
             arr_len = len(arr)
             for i in range(arr_len - (arr_len//2)):
                  if arr[i] == arr[i+(arr_len//2)]:
                      return arr[i]
             return -1
In [14]:
         arr = [1,1,1,1,2,3,4]
         findMajorityele(arr)
         # ans 1
Out[14]: 1
In [15]: arr = [1,2,3,4,4,4,4]
         findMajorityele(arr)
         # ans 4
Out[15]: 4
In [16]:
         arr = [4,5,8,8,8,8]
         findMajorityele(arr)
         # ans 8
Out[16]: 8
In [17]:
         arr = [3,4,5,8,8,8]
         findMajorityele(arr)
         # ans -1
Out[17]: -1
In [18]: arr = [1,2,2]
         findMajorityele(arr)
         # ans 2
Out[18]: 2
```

MOORE VOTING ALGORITHM

If the array is not sorted just count the votes and decide Complexity: Time: O(n) --> scanning entire array while counting votes Space: O(1)

```
voter = i
                                votes = 1
                   else:
                        voter = i
                        votes +=1
               return voter
           def checkMajorityElement(arr,ele):
               count = 0
               for i in arr:
                   if i == ele:
                       count+=1
               if count > (len(arr)//2):
                   return True
           def findMajorityelement(arr):
               potential_candidate = getMajorityElement(arr)
               if potential_candidate:
                    if checkMajorityElement(arr,potential_candidate):
                        print("found {0}".format(potential_candidate))
                        print("Not found")
               else:
                   print("Not found")
           arr = [1,1,1,1,2,3,4]
 In [20]:
           findMajorityelement(arr)
           # ans found 1
          found 1
 In [21]: arr = [1,2,3,4,4,4,4]
           findMajorityelement(arr)
           # ans found 4
          found 4
 In [22]: arr = [4,5,8,8,8,8]
           findMajorityelement(arr)
           # ans found 8
          found 8
 In [23]: arr = [3,4,5,8,8,8]
           findMajorityelement(arr)
           # ans Not found
          Not found
 In [24]: arr = [1,2,2]
           findMajorityelement(arr)
           # ans found 2
          found 2
Another simple way of same logic
 In [25]: # int getMajorityElement(int arr[], int size)
           # int majorityIndex=0, count=1;
           # for(int index = 1; index < size; index++)</pre>
           # {
           # if(arr[majorityIndex] == arr[index])
```

count++; # else # count--;

{

if(count == 0)

```
# }
# return arr[majorityIndex];
# }

In []:
In []:
```

majorityIndex = index;
count = 1;
}