MOORES VOTING ALGORITHM (For finding the majority element in the array ).

There are number of ways with which we can find the majority element in an array.

**A majority element is a element which appears more than n/2 times in the array.**

Solution 1:

The first solution is to run a outer loop (i.e loop i) and then the inner loop (i.e loop j) and make a count variable to keep track of element having maximum count.

And then print the element which has count greater than n/2.

The cons of this approach is that it takes time complexity of O(n\*n), which is not recommended as a optimized code.

**Solution 2:- Using Moore Voting Algorithm**

**Time complexity :-** O(n)

Steps:-

1: First find the majority element by running the outer loop with the two candidates (variables first and second) use and maintain their counts. There may be a chance that we have more than one majority element(greater than n/2) present in the array.

2:- Now make the two count again zero.

3:- Now iterate again through the array to find the one element which is occurring maximum times.

Code :-

int majority\_element(int arr[],int n)

{

int first=INT\_MAX,second=INT\_MAX;

int count\_1=0,count2=0;

for(int i=0;i<n;i++)

{

if(first==arr[i])

count1++;

else if(second==arr[i])

count2++;

else if(count1==0)

first=arr[i];

else if(count2==0)

second=arr[i];

else {

count1--;

count2--;

}

}

count1=0;

count2=0;

for(int i=0;i<n;i++)

{

if(first==arr[i])

count1++;

else if(second==arr[i])

count2++;

}

if(count1>n/2)

return first;

if(count2>n/2)

return second;

return -1;

}

Related Questions:-

* Program to find maximum when N/3 repeated is taken.