Find the Number Occurring Odd Number of Times

Given an array of positive integers. All numbers occur even number of times except one number which occurs odd number of times. Find the number in O(n) time & constant space.

Example:
I/P = [1, 2, 3, 2, 3, 1, 3]
O/P = 3

Algorithm:
Do bitwise XOR of all the elements. Finally we get the number which has odd occurrences.

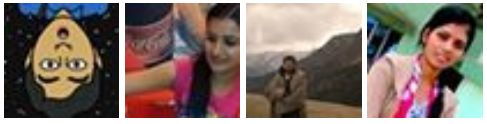
Program:

```
#include <stdio.h>

int getOddOccurrence(int ar[], int ar_size)
{
     int i;
     int res = 0;
     for (i=0; i < ar_size; i++)
        res = res ^ ar[i];

     return res;
}

/* Diver function to test above function */
int main()
{
     int ar[] = {2, 3, 5, 4, 5, 2, 4, 3, 5, 2, 4, 4, 2};
     int n = sizeof(ar)/sizeof(ar[0]);
     printf("%d", getOddOccurrence(ar, n));
```

```
    return 0;
}
```

Time Complexity: O(n)

Related Tpoics:

- Remove minimum elements from either side such that 2*min becomes more than max
- Divide and Conquer | Set 6 (Search in a Row-wise and Column-wise Sorted 2D Array)
- Bucket Sort
- Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
- Find the number of zeroes
- Find if there is a subarray with 0 sum
- Divide and Conquer | Set 5 (Strassen's Matrix Multiplication)
- Count all possible groups of size 2 or 3 that have sum as multiple of 3

Writing code in comment? Please use **ideone.com** and share the link here.

Popular Posts

Sort by Newest

Join the discussion…

groomnestle • 5 months ago

There is a similar post here (find the element that appeared once, method 2):

```
int getOddOccurrence(int ar[], int ar_size)

{

int result = 0;



int x, sum;



// Iterate through every bit

for (int i = 0; i < ar_size; i++)
```

see more

• Reply • Share ›

Amit Baghel • 6 months ago

visited :D

Reply • Share ›

guest ➚ Amit Baghel • 6 months ago

so u are marking all the posts as visited:p

15 | Reply • Share ›

Pratik Sahoo • 7 months ago

We can also do it by Hash Map where the hash array would keep the no of occ
Complexity

1 | Reply • Share ›

Vinod ➚ Pratik Sahoo • 6 months ago

@Pratik...In case of Hash map you are considering only the time comp

1 | Reply • Share ›

Pratik Sahoo ➚ Vinod • 6 months ago

Since there is no restriction on the Space... we can use it..

Reply • Share ›

hookah ➚ Pratik Sahoo • 5 months ago

The question mentions Find the number in O(n) time &

3 | Reply • Share ›

Avinash Nigam • 7 months ago

```
public static int getOddOccuringElement(int[] arr)
```

705

Subscribe

Recent Comments

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 18 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 22 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 47 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 48 minutes ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

newCoder3006 Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

```
    {

            int oddOccuringElement = 0;

            for (int i = 0; i < arr.length - 1; i = i + 2)

            {

                    oddOccuringElement = oddOccuringElement + arr

            }

            oddOccuringElement = oddOccuringElement + arr[arr.len
```

see more

• Reply • Share ›

Marsha Donna • 8 months ago

the prblm can be solved by subtractng elemnts of the array in sequence,
in the end only the ele ocring odd no of times wil be left....pls corrct
me if anything is wrong..see the follwing code

```
#include <stdio.h>
#include<math.h>
int getOddOccurrence(int ar[], int ar_size)
{
int i;
int res = 0;
ar[ar_size]=0;
for (i=0; i <= ar_size; i=i+2)
```

```
res = ar[n-1] ar[i];

return abs(res);
}

/* Diver function to test above function */
int main()
{
int ar[] = {55,77,44,55,77,44,44};
int n = sizeof(ar)/sizeof(ar[0]);
printf("%d", getOddOccurrence(ar, n));
return 0;
}
```

Sourin Sutradhar Marsha Donna • 7 months ago

This wont work as you are totally neglecting the initial calculated values example, if ar[] = {1,2,1,2,0,2,1,2,0}, this gives output as '0' , which is w

Marsha Donna Sourin Sutradhar • 7 months ago

thanks i dint notice that case

Sourin Sutradhar Marsha Donna • 7 months ago

you are welcome

Guest • 8 months ago

the prblm can be solved by subtractng elemnts of the array in sequence in the times wil be left....pls corrct me if anything is wrong

```
#include <stdio.h>
```

```
int getOddOccurrence(int ar[], int ar_size)
{
int i;
int res = 0;
ar[ar_size]=0;
for (i=0; i <= ar_size; i=i+2)
res = ar[i+1]-ar[i];

return abs(res);
}

/* Diver function to test above function */
int main()
{
int ar[] = {55,77,44,55,77,44,44};
int n = sizeof(ar)/sizeof(ar[0]);
printf("%d", getOddOccurrence(ar, n));
return 0;
}
```

  • Reply • Share ›

Chinmaya • a year ago

Nice Question. Thanks..

  • Reply • Share ›

Ujjwal ➚ Chinmaya • a year ago

yeah gud 1 but i guess its a standard problem so chances are that eve

  • Reply • Share ›

dazer • a year ago

sorry i didn read the question properly :(i missed "other elements occuring ev

any suggestion for finding out the odd occurence of element in o(n) time with r

occuring count ??

Reply • Share ›

Ankit Sablok ➔ dazer • a year ago

There we will have to use auxiliary memory in the form of a hashmap I time performance.

```
/* Paste your code here (You may delete these lines if not wri
```

Reply • Share ›

dazer • a year ago

i guess your algo doesnt work for input : {1,2,1,1}
answer should be 1 but its giving 3 .
so this algorithm works wrong :P

```
/* Paste your code here (You may delete these lines if not writing c
```

Reply • Share ›

Ankit Sablok ➔ dazer • a year ago

I think you didn't read the question carefully it says all other elements o case both 1 and 2 occur odd number of times.

1 Reply • Share ›

Nishant Kumar • a year ago

Please suggest me an algo in Time Complexity: O(n) and space Complexity: (large i.e instead of only one odd time repeating no if we have suppose 15 odd repeating). if possible

e.g - {2, 3, 5, 4, 5, 2, 3, 5, 2};

Reply • Share ›

nick ➚ Nishant Kumar • a year ago

What is that you exactly want ? Do you want to print all the odd numbe
have 15 odd time repeating no so you want to store them and print it ?

Reply • Share ›

Nishant ➚ nick • a year ago

@nick yes, i want to print all numbers occurring odd numbers c

Reply • Share ›

geekyboy ➚ Nishant • a year ago

even i have the same question.

How do we go about finding all the odd no of times occu

in that case xor won't work
and what if we want O(n) solution

Reply • Share ›

Nishant Kumar ➚ geekyboy • a year ago

@geekyboy if you don't bother abt space Complexity: O
using HashMap

```
/* Paste your code here (You may delete these li
```

1 Reply • Share ›

Saurabh • 2 years ago

@geeksForGeeks: I think your algo would n't work for the sequence:
int ar[] = {7, 3, 5, 4, 5, 2,2, 4, 3, 5, 2, 6, 6, 2,7};
please tell me if I am wrong.

Reply • Share ›

abcd ➚ Saurabh • a year ago

@saurabh:
Its not working as 2,5 i.e. 2 nos. r there odd no. of time but as per the q

```
/* Paste your code here (You may delete these lines if not wri
```

▲ | ▼ • Reply • Share ›

Sumeet ➚ Saurabh • a year ago

@Saurabh: It is working for this case also. Answer is 5. Take a closer

```
/* Paste your code here (You may delete these lines if not wri
```

▲ | ▼ • Reply • Share ›

GeeksforGeeks ➚ Saurabh • 2 years ago

Take a closer look at the algorithm. It's a simple and standard algorithm

▲ | ▼ • Reply • Share ›

rohith ➚ GeeksforGeeks • 10 months ago

Could you please explain the logic behind this algorithm.

How does doing XOR of all the elements get me this result, not
Thank you.

2 ▲ | ▼ • Reply • Share ›

Shubham Lakhiwal • 2 years ago

```
int main()
{
   int a[] ={2, 3, 5, 4, 4, 2, 4, 3, 5, 2, 4, 4, 2};
   int maj_index = 0, count = 0,i=0;
```

```
    while(count%2==0){
        count = 0;
        for(i = 0; i<13; i++){
          if (a[maj_index] == a[i])
            count++;
        }
        maj_index++;
    }
    printf("%d",a[maj_index-1]);

     scanf("%d",&i);
     return 0;
}
```

1 ▲ | ▼ • Reply • Share ›

Mo Pu ➦ Shubham Lakhiwal • a year ago

Correct algo, but time complexity is O(n^2)

```
/* Paste your code here (You may delete these lines if not wri
```

▲ | ▼ • Reply • Share ›

Saurabh ➦ Shubham Lakhiwal • 2 years ago

@Shubham Lakhiwal: I think your algo would n't work for the sequence
int ar[] = {7, 3, 5, 4, 5, 2,2, 4, 3, 5, 2, 6, 6, 2,7};
please tell me if I am wrong.

▲ | ▼ • Reply • Share ›

Anuj Bansal • 2 years ago

Here is another solution to this problem.

```
 #include <stdio.h>
#include<math.h>
int getOddOccurrence(int ar[], int ar_size)
{
     int i,j,res;
     i = 0; j = ar_size-1; res = 0;
     while(i < ar_size/2)
        res = res - ar[i++] + ar[j--];
     res = res - ar[i];
     return abs(res);
}

int main()
{
     int ar[] = {2, 3, 5, 4, 5, 2, 4, 3, 5, 2, 4, 4, 2};
     printf("%d\n", getOddOccurrence(ar, 13));
     getchar();
}
```

• Reply • Share ›

Lakshminarayana Anuj Bansal • 5 months ago

above solution is not working for { 7,3,5,4,5,2,2,4,3,5,2,6,6,2,7 }. Please

• Reply • Share ›

Shashank Anuj Bansal • 2 years ago

The above solution mentioned in first comment is incorrect as for an a
however it would work if we change the distribution to 2,4,3,3,3,4,2. So
only if elements before middle element and elements after middle elem
numbers.correct me if im wrong.

• Reply • Share ›

Ankita ➦ Shashank • 10 months ago

```
// Just need a little bit correction in above code. But
    #include <stdio.h>
    #include<math.h>
    int getOddOccurrence(int ar[], int ar_size)
    {
         int i=0,res = 0;
         while(i < ar_size)
         {
            (i%2==0)?(res = res + ar[i++]):(res = res -
            }
          return res;
    }
    int main()
    {
         int a[] = {2,2,2,2,3,3,3,3,3,4,4,4,4};
         int size=sizeof(a)/sizeof(a[0]);
         printf("%d\n", getOddOccurrence(a, size));
         getchar();
    }
```

1 ^ | v • Reply • Share ›

donbosio • 4 years ago

thanks but if the ques would ave been to find the even occuring element where

^ | v • Reply • Share ›