Find even occurring elements in an array of limited range

Given an array that contains odd number of occurrences for all numbers except for a few elements which are present even number of times. Find the elements which have even occurrences in the array in O(n) time complexity and O(1) extra space.

Assume array contain elements in the range 0 to 63.

Examples:

Input: [9, 12, 23, 10, 12, 12, 15, 23, 14, 12, 15] Output: 12, 23 and 15 Input: [1, 4, 7, 5, 9, 7, 3, 4, 6, 8, 3, 0, 3] Output: 4 and 7 Input: [4, 4, 10, 10, 4, 4, 4, 4, 10, 10] Output: 4 and 10

on to the solution. A simple method would be to traverse the array and store frequencies of its elements in a map. Later, print

elements that have even count in the map. The solution takes O(n) time but requires extra space for storing

We strongly recommend that you click here and practice it, before moving

frequencies. Below is an interesting method to solve this problem using bitwise operators. This method assumes that long long integers are stored using 64 bits. The idea is to use XOR operator. We know that

1 XOR 1 = 0 1 XOR 0 = 1

```
0 XOR 1 = 1
 0 XOR 0 = 0
Consider below input -
  1, 4, 7, 5, 9, 7, 3, 4, 6, 8, 3, 0, 3
```

If we right shift 1 by value of each element of the array and take XOR of all the items, we will get below binary integer -

Each 1 in the i'th index from the right represents odd occurrence of element i. And each 0 in the i'th index from

the right represents even or non-occurrence of element i in the array.

1101101011

0 is present in 2nd, 4th and 7th position in above binary number. But 2 is not present in our array. So our answer is 4 and 7.

// C++ Program to find the even occurring elements // in given array

#include <iostream> using namespace std;

Below is C++ implementation of above idea

// Function to find the even occurring elements // in given array void printRepeatingEven(int arr[], int n) long long _xor = 0L; long long pos; // do for each element of array for(int i = 0; i < n; ++i) // right pos 1 by value of current element pos = 1 << arr[i]; // Toggle the bit everytime element gets repeated // Traverse array again and use _xor to find even // occuring elements for (int i = 0; i < n; ++i) // right shift 1 by value of current element pos = 1 << arr[i]; // Each 0 in _xor represents an even occurrence if (!(pos & _xor)) // print the even occurring numbers cout << arr[i] << " "; // set bit as 1 to avoid printing duplicates _xor ^= pos;

int n = sizeof(arr) / sizeof(arr[0]); printRepeatingEven(arr, n); return 0;

int arr[] = { 9, 12, 23, 10, 12, 12, 15, 23,

14, 12, 15 };

Output: 12 23 15

// Driver code

int main()

This article is contributed by Aditya Goel. If you like GeeksforGeeks and would like to contribute, you can also

write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

Schneider



Find Two Missing Numbers | Set 2 (XOR based solution) Find profession in a special family

Print first n numbers with exactly two set bits

16 Comments Category: Bit Magic

Generate 0 and 1 with 25% and 75% probability

16 Comments

Recommend 2

Related Posts:

- · Cyclic Redundancy Check and Modulo-2 Division · Copy set bits in a range Check if a number is Bleak
- Count strings with consecutive 1's Previous post in category

Logged in as Ruturaj Mohanty 1 (Logout) Average Difficulty: 2.5/5.0 Based on 13 vote(s) Add to TODO List Mark as DONE

Join the discussion...

GeeksforGeeks

Share

Writing code in comment? Please use code.geeksforgeeks.org, generate link and share the link here.

Kesavan P - 2 months ago here we doing right shift or left shift..?(my confusion is << mentioned as right shift)

bhavik gujarati → Kesavan P • 2 months ago They have done left shift but written "right shift". It should be "left shift".

∧ V - Reply - Share >

Reply - Share > Kesavan P → bhavik gujarati - 2 months ago Hmm.. bhavik

they are doing right shift 1 on array element . u can read like that arr[i]>>1;

Aditya Gupta A bhavik gujarati - a month ago

∧ V - Reply - Share >

discussion on bit manipulation at http://notes4geeks.com/main/bi... ∧ V - Reply - Share > Rajat Kumar Jha - 3 months ago

Your solution is failed on this test case:

testacc - 2 months ago

10 43 5 27 29 26 4 39 61. ∧ V - Reply - Share >

Rajat Kumar Jha • 3 months ago

hmm.. 1(int) should be replaced by 1L(long) http://ideone.com/Pc1RPS 2 A V - Reply - Share

AllergicToBitches → Rajat Kumar Jha - 3 months ago

468 335 1 170 225 479 359 463 465 206 146 282 328 462 492 496 443 328 437 392 105 403

154 293 383 422 217 219 396 448 227 272 39 370 413 168 300 36 395 204 312 323. ∧ V - Reply - Share >

in optimal way

Your solution is failed on this test case.

AllergicToBitches → Rajat Kumar Jha - 3 months ago Problem statement states that array should contain elements in the range 0 to 63. ∧ V - Reply - Share >

Oh Sorry, Thank You. But in case, range will more than 63 then how will find

We can do in using O(n) space by storing count of repeating elements in a map. If the count is even, we print the element.

Rajat Kumar Jha - AllergicToBitches - 3 months ago

9 35 41 18 38 13 1 62 21 36 48 35 7 8 51 35 36 40 19 10 52 60 28 41 35 34 6 63 51 30 55 30

Reply - Share > AllergicToBitches → Rajat Kumar Jha - 3 months ago

Refer - http://ideone.com/sV3yqY ∧ V - Reply - Share > Rajat Kumar Jha → AllergicToBitches • 3 months ago Thank You @AllergicToBitches

i could see the output is proper but i didnt get the logic of XOR here for multiple numbers

∧ V - Reply - Share >

∧ V - Reply - Share > anon -> Vinodh - 3 months ago

occuring even times. anyone pls explain..

Vinodh - 3 months ago

The first for-loop flips the i-th bit of _xor whenever number i is in the array. When the loop completes, the 0-bits mark which values (between 0 and 63) were encountered an even number of times. However, there is a slight complication: number 0 is even,

times? The second for-loop tries to resolve this issue, but IMHO that code may be improved. To see the problem consider the input array of 1,000,000,000 zeroes. The

output should be "0" but the second loop will require 10^9 iterations rather than at most 63. To improve the code, I would suggest to add another mask variable, _present, of the same size as _xor. Set the i-th bit of _present whenever we see number i in the array in the first loop: present &= 1 << pos . Then iterate over the bits of xor by

too. :-) Should we also print the values that are absent from the array, i.e. seen 0

right- shifting it one bit at a time in a while loop until the value becomes zero (at most 64 -- and maybe fewer -- times). If the bit is zero and the same bit of _present is not zero, print the index. ∧ V - Reply - Share >

anon → anon - 3 months ago Sorry: should be _present |= pos ∧ V - Reply - Share >

anon - 3 months ago

Add Disgus to your site

Subscribe

The implementation is here: http://ideone.com/PnJsI0 ∧ V - Reply - Share >

Privacy

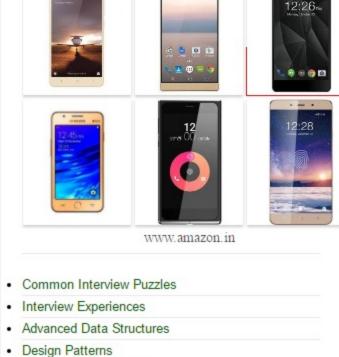
@geeksforgeeks, Some rights reserved

Start Coding Today



Competitive Programming

- Top 10 algorithms in Interview Questions How to begin with Competitive Programming?
- Reflection in Java Memory Layout of C Programs
- Push Relabel Algorithm
- Heavy Light Decomposition Sorted Linked List to Balanced BST
- Generics in Java Aho-Corasick Algorithm for Pattern Searching
- · Binary Search , QuickSort , MergeSort , HeapSort



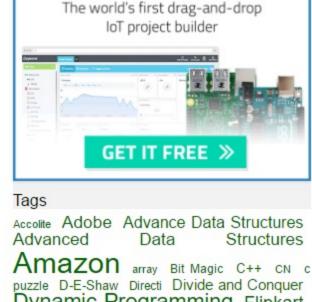
- Dynamic Programming
- Greedy Algorithms
- Backtracking Pattern Searching
- Divide & Conquer Geometric Algorithms
- Searching Sorting
- Hashing Analysis of Algorithms Mathematical Algorithms
- Randomized Algorithms Recursion

Run on IDE

Next post in category

Login -

Sort by Newest -



Beta Test **Cayenne** with

Arduino & Raspberry Pi

Hashing Interview Experience Java MakeMyTrip Software MathematicalAlgo Microsoft Morgan Stanley Operating systems Oracle Pattern Searching puzzle Python Recursion samsung SAP Labs SnapDeal stack Stack-Queue STL Zoho Subscribe and Never Miss an Article Email Address Subscribe

Like us on Facebook

Arnab Das The company didn't told us anything.....so...

Follow us on Twitter

Amdocs Interview Experience | Set 4 (On-Campus Software Engineering Associate) · 15 minutes ago

site,...

Subsequence) : 18 minutes ago Tapan Avasthi The Java solution posted will give

sonu was there section cut-off

Lowest Common Ancestor in a Binary Tree | Set 1 · 21 minutes

Rakesh Kejriwal nop, check

neethu t can we also use threaded binary tree

Select Category

GeeksQuiz

1233577

All Categories

Inorder Successor in Binary Search Tree · 50 minutes ago

 GeeksforGeeks Practice Data Structures Algorithms

Books

GATE CS Forum

Android App

Dynamic Programming Flipkart GATE GATE-CS-DS-&-Algo GATE-CS-GFacts Goldman Sachs Graph Greedy Algorithm

Recent Comments

dmr According to one comment on career cup Dynamic Programming | Set 12 (Longest Palindromic

Amdocs Interview Experience | Set 4 (On-Campus Software Engineering Associate) · 49 minutes ago

Check if array elements are consecutive | Added Method 3 · 50 minutes ago

 C Programming C++ Programming Java Programming Interview Experiences GATE CS

Contact Us!