GeeksforGeeks
A computer science portal for geeks
Practice IDE Q&A GeeksQuiz

Find duplicates in O(n) time and O(1) extra space

Given an array of n elements which contains elements from 0 to n-1, with any of these numbers appearing any number of times. Find these repeating numbers in O(n) and using only constant memory space.

For example, let n be 7 and array be {1, 2, 3, 1, 3, 6, 6}, the answer should be 1, 3 and 6.

This problem is an extended version of following problem.

Find the two repeating elements in a given array

Method 1 and Method 2 of the above link are not applicable as the question says O(n) time complexity and O(1) constant space. Also, Method 3 and Method 4 cannot be applied here because there can be more than 2 repeating elements in this problem. Method 5 can be extended to work for this problem. Below is the solution that is similar to the Method 5.

Algorithm:

```
traverse the list for i= 0 to n-1 elements
{
  check for sign of A[abs(A[i])] ;
  if positive then
     make it negative by   A[abs(A[i])]=-A[abs(A[i])];
  else  // i.e., A[abs(A[i])] is negative
     this   element (ith element of list) is a repetition
}
```

Implementation:

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

void printRepeating(int arr[], int size)
{
  int i;
  printf("The repeating elements are: \n");
  for (i = 0; i < size; i++)
  {
    if (arr[abs(arr[i])] >= 0)
      arr[abs(arr[i])] = -arr[abs(arr[i])];
    else
```

```
      printf(" %d ", abs(arr[i]));
  }
}

int main()
{
  int arr[] = {1, 2, 3, 1, 3, 6, 6};
  int arr_size = sizeof(arr)/sizeof(arr[0]);
  printRepeating(arr, arr_size);
  getchar();
  return 0;
}
```

Run on IDE

Note: The above program doesn't handle 0 case (If 0 is present in array). The program can be easily modified to handle that also. It is not handled to keep the code simple.

Output:

The repeating elements are:

1 3 6

Time Complexity: O(n)

Auxiliary Space: O(1)

Please write comments if you find the above codes/algorithms incorrect, or find better ways to solve the same problem.

221 Comments Category: Arrays

Related Posts:

- Longest Span with same Sum in two Binary arrays
- Count Inversions of size three in a give array
- Find the subarray with least average

- Count triplets with sum smaller than a given value
- Find zeroes to be flipped so that number of consecutive 1’s is maximized
- Reorder an array according to given indexes
- Find maximum value of Sum(i*arr[i]) with only rotations on given array allowed
- Find maximum average subarray of k length

(Login to Rate and Mark)

2.6 Average Difficulty : **2.6/5.0**
Based on **6** vote(s)

□ Add to TODO List

□ Mark as DONE

Like Share 28 people like this. Be the first of your friends.

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

Contact Us! About Us! Advertise with us!