

GeeksforGeeks

A computer science portal for geeks

Practice

IDE

Q&A

GeeksQuiz

Find the number of zeroes

Given an array of 1s and 0s which has all 1s first followed by all 0s. Find the number of 0s. Count the number of zeroes in the given array.

Examples:

Input: arr[] = {1, 1, 1, 1, 0, 0}

Output: 2

Input: arr[] = {1, 0, 0, 0, 0}

Output: 4

Input: arr[] = {0, 0, 0}

Output: 3

Input: arr[] = {1, 1, 1, 1}

Output: 0

We strongly recommend to minimize the browser and try this yourself in time complexity better than $O(n)$.

A **simple solution** is to traverse the input array. As soon as we find a 0, we return $n - \text{index of first 0}$. Here n is number of elements in input array. Time complexity of this solution would be $O(n)$.

Since the input array is sorted, we can use **Binary Search to find the first occurrence** of 0. Once we have index of first element, we can return count as $n - \text{index of first zero}$.

```
// A divide and conquer solution to find count of zeroes in an array
// where all 1s are present before all 0s
#include <stdio.h>

/* if 0 is present in arr[] then returns the index of FIRST occurrence
   of 0 in arr[low..high], otherwise returns -1 */
int firstZero(int arr[], int low, int high)
{
    if (high >= low)
    {
        // Check if mid element is first 0
        int mid = low + (high - low)/2;
```

```
    if (( mid == 0 || arr[mid-1] == 1) && arr[mid] == 0)
        return mid;

    if (arr[mid] == 1) // If mid element is not 0
        return firstZero(arr, (mid + 1), high);
    else // If mid element is 0, but not first 0
        return firstZero(arr, low, (mid - 1));
}
return -1;
}

// A wrapper over recursive function firstZero()
int countZeroes(int arr[], int n)
{
    // Find index of first zero in given array
    int first = firstZero(arr, 0, n-1);

    // If 0 is not present at all, return 0
    if (first == -1)
        return 0;

    return (n - first);
}

/* Driver program to check above functions */
int main()
{
    int arr[] = {1, 1, 1, 0, 0, 0, 0, 0};
    int n = sizeof(arr)/sizeof(arr[0]);
    printf("Count of zeroes is %d", countZeroes(arr, n));
    return 0;
}
```

Output:

```
Count of zeroes is 5
```

Time Complexity: $O(\log n)$ where n is number of elements in `arr[]`.

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



Querying with Transact-SQL

[Enroll now](#)

Self-Paced

45 Comments Category: [Arrays](#) [Divide and Conquer](#) Tags: [Divide and Conquer](#)

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

Average Difficulty : **2/5.0**
Based on **4** vote(s)

☐[Add to TODO List](#)☐[Mark as DONE](#)

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

[@geeksforgeeks](#), [Some rights reserved](#)

[Contact Us!](#)[About Us!](#)[Advertise with us!](#)