GeeksforGeeks
A computer science portal for geeks

GeeksQuiz

Login

| Home | Algorithms | DS | GATE | Interview Corner | Q&A | C | C++ | Java | Books | Contribute | Ask a Q | About |

| Array | Bit Magic | C/C++ | Articles | GFacts | Linked List | MCQ | Misc | Output | String | Tree | Graph |

Segregate 0s and 1s in an array

Asked by kapil.

You are given an array of 0s and 1s in random order. Segregate 0s on left side and 1s on right side of the array. Traverse array only once.

```
Input array   = [0, 1, 0, 1, 0, 0, 1, 1, 1, 0]
Output array = [0, 0, 0, 0, 0, 1, 1, 1, 1, 1]
```

Method 1 (Count 0s or 1s)
Thanks to Naveen for suggesting this method.
1) Count the number of 0s. Let count be C.
2) Once we have count, we can put C 0s at the beginning and 1s at the remaining n – C positions in array.

Time Complexity: O(n)

The method 1 traverses the array two times. Method 2 does the same in a single pass.

Method 2 (Use two indexes to traverse)
Maintain two indexes. Initialize first index *left* as 0 and second index *right* as n-1.

Do following while *left* < *right*
a) Keep incrementing index *left* while there are 0s at it
b) Keep decrementing index *right* while there are 1s at it
c) If left < right then exchange arr[left] and arr[right]

Implementation:

```c
#include<stdio.h>

/*Function to put all 0s on left and all 1s on right*/
void segregate0and1(int arr[], int size)
{
  /* Initialize left and right indexes */
  int left = 0, right = size-1;

  while(left < right)
  {
     /* Increment left index while we see 0 at left */
     while(arr[left] == 0 && left < right)
        left++;

     /* Decrement right index while we see 1 at right */
     while(arr[right] == 1 && left < right)
        right-;

    /* If left is smaller than right then there is a 1 at left
      and a 0 at right.  Exchange arr[left] and arr[right]*/
     if(left < right)
     {
       arr[left] = 0;
       arr[right] = 1;
       left++;
       right-;
     }
  }
}

/* driver program to test */
int main()
{
  int arr[] = {0, 1, 0, 1, 1, 1};
  int arr_size = 6, i = 0;

  segregate0and1(arr, arr_size);

  printf("array after segregation ");
  for(i = 0; i < 6; i++)
    printf("%d ", arr[i]);

  getchar();
  return 0;
}
```

Popular Posts

All permutations of a given string

Memory Layout of C Programs

Understanding "extern" keyword in C

Median of two sorted arrays

Tree traversal without recursion and without

stack!

Structure Member Alignment, Padding and

Data Packing

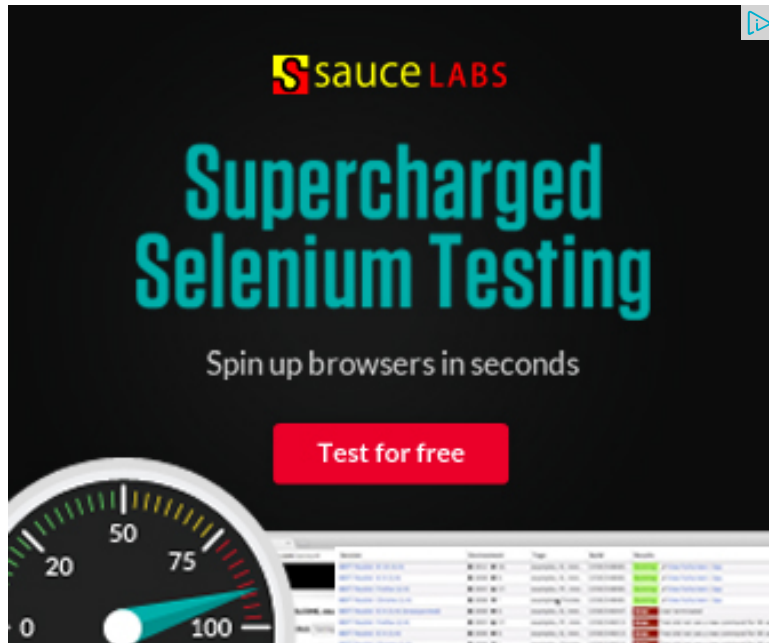Intersection point of two Linked Lists

Lowest Common Ancestor in a BST.

Check if a binary tree is BST or not

Sorted Linked List to Balanced BST

Time Complexity: O(n)

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

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

13   **Tweet** 1   0

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

Recent Comments

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

Write a C program to Delete a Tree. · 24 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 28 minutes ago

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

Root to leaf path sum equal to a given number · 53 minutes ago

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

Count trailing zeroes in factorial of a number · 54 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

► Memory Array

► An Array

► Programming C++

► Int Byte Array

► Array Pointer

► Array Function