**6.Segregate 0s and 1s in an array**

[**https://www.geeksforgeeks.org/segregate-0s-and-1s-in-an-array-by-traversing-array-once/**](https://www.geeksforgeeks.org/segregate-0s-and-1s-in-an-array-by-traversing-array-once/)

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)

SOLUTION BY ME:

**package** com.infogain.hackerrank;

**public** **class** Segreate0and1 {

**public** **static** **void** main(String[] args) {

**int** ar1[]={0, 1, 0, 1, 0, 0, 1, 1, 1, 0};

**int** ar2[]=**new** **int**[ar1.length];

**int** count1=0,count0=0;

**for**(**int** i=0;i<ar1.length;i++){

**if**(ar1[i]==0){

count0++;

}

**if**(ar1[i]==1){

count1++;

}

}

**for**(**int** i=0;i<count0;i++){

ar2[i]=0;

}

**for**(**int** j=count0;j<ar2.length;j++){

ar2[j]=1;

}

**for**(**int** x:ar2){

System.***out***.print(x);

}

/\* System.out.println(count0);

System.out.println(count1);\*/

}

}

**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:

\*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 above functions \*/

    public static void main(String[] args)

    {

        Segregate seg = new Segregate();

        int arr[] = new int[]{0, 1, 0, 1, 1, 1};

        int i, arr\_size = arr.length;

        seg.segregate0and1(arr, arr\_size);

        System.out.print("Array after segregation is ");

        for (i = 0; i < 6; i++)

            System.out.print(arr[i] + " ");

    }

}