Reversal algorithm for array rotation

Write a function rotate(arr[], d, n) that rotates arr[] of size n by d elements.

Example:

```
Input:  arr[] = [1, 2, 3, 4, 5, 6, 7]
            d = 2
Output: arr[] = [3, 4, 5, 6, 7, 1, 2]
```

Rotation of the above array by 2 will make array

Method 4(The Reversal Algorithm)

Please read this for first three methods of array rotation.

Algorithm:

```
rotate(arr[], d, n)
  reverse(arr[], 1, d) ;
  reverse(arr[], d + 1, n);
  reverse(arr[], 1, n);
```

Let AB are the two parts of the input array where A = arr[0..d-1] and B = arr[d..n-1]. The idea of the algorithm is:

Reverse A to get ArB. /* Ar is reverse of A */

Reverse B to get ArBr. /* Br is reverse of B */

Reverse all to get (ArBr) r = BA.

For arr[] = [1, 2, 3, 4, 5, 6, 7], d =2 and n = 7

A = [1, 2] and B = [3, 4, 5, 6, 7]

Reverse A, we get ArB = [2, 1, 3, 4, 5, 6, 7]

Reverse B, we get ArBr = [2, 1, 7, 6, 5, 4, 3]

Reverse all, we get (ArBr)r = [3, 4, 5, 6, 7, 1, 2]

Implementation:

C/C++

```
// C/C++ program for reversal algorithm of array rotation
#include<stdio.h>

/*Utility function to print an array */
void printArray(int arr[], int size);

/* Utility function to reverse arr[] from start to end */
void rvereseArray(int arr[], int start, int end);

/* Function to left rotate arr[] of size n by d */
void leftRotate(int arr[], int d, int n)
{
    rvereseArray(arr, 0, d-1);
    rvereseArray(arr, d, n-1);
    rvereseArray(arr, 0, n-1);
}

/*UTILITY FUNCTIONS*/
/* function to print an array */
void printArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", arr[i]);
}

/*Function to reverse arr[] from index start to end*/
void rvereseArray(int arr[], int start, int end)
{
    int temp;
    while (start < end)
    {
        temp = arr[start];
        arr[start] = arr[end];
        arr[end] = temp;
        start++;
        end--;
    }
}
```

```
/* Driver program to test above functions */
int main()
{
    int arr[] = {1, 2, 3, 4, 5, 6, 7};
    int n = sizeof(arr)/sizeof(arr[0]);
    int d = 2;
    leftRotate(arr, d, n);
    printArray(arr, n);
    return 0;
}
```

Java

```
// Java program for reversal algorithm of array rotation
import java.io.*;

class LeftRotate
{
    /* Function to left rotate arr[] of size n by d */
    static void leftRotate(int arr[], int d)
    {
        int n = arr.length;
        rvereseArray(arr, 0, d-1);
        rvereseArray(arr, d, n-1);
        rvereseArray(arr, 0, n-1);
    }

    /*Function to reverse arr[] from index start to end*/
    static void rvereseArray(int arr[], int start, int end)
    {
        int temp;
        while (start < end)
        {
            temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            start++;
            end--;
        }
    }

    /*UTILITY FUNCTIONS*/
    /* function to print an array */
    static void printArray(int arr[])
    {
        for (int i = 0; i < arr.length; i++)
            System.out.print(arr[i] + " ");
    }
```

```
    /* Driver program to test above functions */
    public static void main (String[] args)
    {
        int arr[] = {1, 2, 3, 4, 5, 6, 7};
        leftRotate(arr, 2); // Rotate array by 2
        printArray(arr);
    }
}
/*This code is contributed by Devesh Agrawal*/
```

Python

```
# Python program for reversal algorithm of array rotation

# Function to reverse arr[] from index start to end
def rverseArray(arr, start, end):
    while (start < end):
        temp = arr[start]
        arr[start] = arr[end]
        arr[end] = temp
        start += 1
        end = end-1

# Function to left rotate arr[] of size n by d
def leftRotate(arr, d):
    n = len(arr)
    rverseArray(arr, 0, d-1)
    rverseArray(arr, d, n-1)
    rverseArray(arr, 0, n-1)

# Function to print an array
def printArray(arr):
    for i in range(0, len(arr)):
        print arr[i],

# Driver function to test above functions
arr = [1, 2, 3, 4, 5, 6, 7]
leftRotate(arr, 2) # Rotate array by 2
printArray(arr)

# This code is contributed by Devesh Agrawal
```

Output:

```
3 4 5 6 7 1 2
```

Time Complexity: O(n)

References:

http://www.cs.bell-labs.com/cm/cs/pearls/s02b.pdf

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

49 Comments Category: Arrays Tags: array

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)

2.5 Average Difficulty : **2.5/5.0**
Based on **11** 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.