Program for array rotation

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

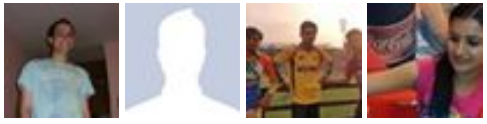
Rotation of the above array by 2 will make array

METHOD 1 (Use temp array)

```
Input arr[] = [1, 2, 3, 4, 5, 6, 7], d = 2, n =7
1) Store d elements in a temp array
   temp[] = [1, 2]
2) Shift rest of the arr[]
   arr[] = [3, 4, 5, 6, 7, 6, 7]
3) Store back the d elements
   arr[] = [3, 4, 5, 6, 7, 1, 2]
```

Time complexity O(n)
Auxiliary Space: O(d)

METHOD 2 (Rotate one by one)



```
leftRotate(arr[], d, n)
start
  For i = 0 to i < d
    Left rotate all elements of arr[] by one
end
```

To rotate by one, store arr[0] in a temporary variable temp, move arr[1] to arr[0], arr[2] to arr[1] … and finally temp to arr[n-1]

Let us take the same example arr[] = [1, 2, 3, 4, 5, 6, 7], d = 2
Rotate arr[] by one 2 times
We get [2, 3, 4, 5, 6, 7, 1] after first rotation and [3, 4, 5, 6, 7, 1, 2] after second rotation.

```
/*Function to left Rotate arr[] of size n by 1*/
void leftRotatebyOne(int arr[], int n);

/*Function to left rotate arr[] of size n by d*/
void leftRotate(int arr[], int d, int n)
{
  int i;
  for (i = 0; i < d; i++)
    leftRotatebyOne(arr, n);
}

void leftRotatebyOne(int arr[], int n)
{
  int i, temp;
  temp = arr[0];
  for (i = 0; i < n-1; i++)
     arr[i] = arr[i+1];
  arr[i] = temp;
}

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

/* Driver program to test above functions */
int main()
```


Popular Posts

```
{
    int arr[] = {1, 2, 3, 4, 5, 6, 7};
    leftRotate(arr, 2, 7);
    printArray(arr, 7);
    getchar();
    return 0;
}
```

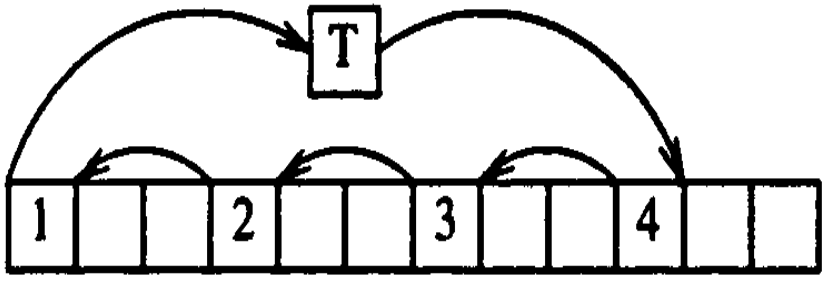
Time complexity: O(n*d)
Auxiliary Space: O(1)

METHOD 3 (A Juggling Algorithm)
This is an extension of method 2. Instead of moving one by one, divide the array in different sets where number of sets is equal to GCD of n and d and move the elements within sets.
If GCD is 1 as is for the above example array (n = 7 and d =2), then elements will be moved within one set only, we just start with temp = arr[0] and keep moving arr[I+d] to arr[I] and finally store temp at the right place.

Here is an example for n =12 and d = 3. GCD is 3 and

```
Let arr[] be {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}

a)      Elements are first moved in first set – (See below diagram for this movement
```




```
         arr[] after this step --> {4 2 3 7 5 6 10 8 9 1 11 12}

b)      Then in second set
```

```
b)      Then in second set.
          arr[] after this step --> {4 5 3 7 8 6 10 11 9 1 2 12}

c)      Finally in third set.
          arr[] after this step --> {4 5 6 7 8 9 10 11 12 1 2 3}
```

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

/*Fuction to get gcd of a and b*/
int gcd(int a,int b);

/*Function to left rotate arr[] of siz n by d*/
void leftRotate(int arr[], int d, int n)
{
  int i, j, k, temp;
  for (i = 0; i < gcd(d, n); i++)
  {
    /* move i-th values of blocks */
    temp = arr[i];
    j = i;
    while(1)
    {
      k = j + d;
      if (k >= n)
        k = k - n;
      if (k == i)
        break;
      arr[j] = arr[k];
      j = k;
    }
    arr[j] = temp;
  }
}

/*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]);
}

/*Fuction to get gcd of a and b*/
int gcd(int a,int b)
```

```
{
   if(b==0)
     return a;
   else
     return gcd(b, a%b);
}

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

Time complexity: O(n)
Auxiliary Space: O(1)

Please see following posts for other methods of array rotation:
Block swap algorithm for array rotation
Reversal algorithm for array rotation

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

Please write comments if you find any bug in above programs/algorithms.

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

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

63 Comments GeeksforGeeks

Sort by Newest

Mukul Kumar • 4 days ago

Anyone please explain concept of GCD in third Method

 • Reply • Share ›

stiwari • 23 days ago

1. Iterate to n-k

2. if i >= k, swap i with i -k

3. else swap i with n-k+i

Code can be found at: http://ideone.com/fZGnpw

 • Reply • Share ›

Tony • 25 days ago

why are u using gcd(d,n) in
for (i = 0; i < gcd(d, n); i++)
because every time when condition is checked gcd is calculated every time. y

 • Reply • Share ›

xiveman • 3 months ago

We can first reverse the part a[0 .. k-1] then a[k .. n-1] and finally the whole arr
k positions.

```
public static void help(int[] a, int l , int r){
while(l < r){
    int t = a[l];
    a[l] = a[r];
    a[r] = t;
    l++; r--;
```

```
    }

    public static void rotate(int[] a, int d){

    if(d % a.length == 0) return;    // no need to rotate

    boolean rotateToLeft = true;   // positive d: rotate to left else
    if(d < 0){ rotateToLeft = false; d = -d; }
```

see more

• Reply • Share ›

Amit Kumar • 4 months ago

Best option is by using modulo :

consider array of size 5 , and i want to shift each element n times then:

(5+n+ i) %5 where i is the actual position of a value in the array. This equation

It can be used along with recursion to shift values. A sample code would be lik

shiftValuesby2 (int array[], int i){

int value;

if(n==5){
return;
}

shiftValueby2(array,i+1);

array[(i+5+2)%5] = value; // shift each value location by 2

}

6 ^ | v • Reply • Share ›

renu • 6 months ago

can anyone please explain why the loop counter has gcd of d and n has the lin
concept behind using gcd?

4 ^ | v • Reply • Share ›

Maggi Lover • 7 months ago

Maggi 2 minute solution-
1) Reverse arr[0, n-d]
2) Reverse arr[n-d+1, n-1]
3) Reverse arr[0,n-1]
Complexity-o(n)

11 ^ | v • Reply • Share ›

Rohit Sharma ➚ Maggi Lover • 2 months ago

reverse_array(a,0,d-1);
reverse_array(a,d,n-1);
reverse_array(a,0,n);
This is correct please check if anything wrong.

^ | v • Reply • Share ›

Neha Garg • 8 months ago

please somebody explain how juggling theorm will work for first example?? or
plz correct where i m wrong

^ | v • Reply • Share ›

Rohit Sharma ➚ Neha Garg • 2 months ago

yes,it is also working if gcd is less than d..!!

^ | v • Reply • Share ›

for(int i=0;i<n;i++) cin="">>a[i];

cin>>d;

for(int i=0;i<n;i=i+2*d){ if(i+2*d="">= n) break;

int right=i+d,k=right;

for(int j=i;j<right;j++) swap(a,j,k++);="" }="">

• Reply • Share ›

Guest • 8 months ago

here is implementation of method 1..pls correct me if there are any mistakes

#include<stdio.h>
#include<stdlib.h>
void arrrot(int arr[],int n,int d)
{
int *temp=(int *)malloc(sizeof(int)*d);
int i,j=0;
for(i=0;i<d;i++) {="" temp[i]="arr[i];" }="" for(i="d;i<n;i++)" {="" arr[i-d]="arr[i];"
arr[i]="temp[j++];" }="" }="" void="" printarray(int="" arr[],="" int="" size)="" {="
i++)="" printf("%d="" ",="" arr[i]);="" }="" int="" main()="" {="" int="" arr[]="{1," 2,
arrrot(arr,="" 7,6);="" printarray(arr,="" 7);="" getchar();="" return="" 0;="" }="">

1 • Reply • Share ›

Kushagra Singhal • 8 months ago

We can use array reversal algorithm to solve this problem. The algo is:
1. Reverse first 'd' elements.
2. Reverse the remaining 'n-d' elements
3. Now reverse the complete array
Time complexity: O(n)

12 ^ | v • Reply • Share ›

Rohan • 8 months ago

Can we make a circular linked list out of array and then perform roation. This v
only.

Please correct me if i am wrong.

^ | v • Reply • Share ›

Sumit Monga • 9 months ago

sorry that I should be I in rotate_array() and main().

^ | v • Reply • Share ›

Sumit Monga • 9 months ago

This solution is based on the same approach used for finding the transpose of
container.

```
#include<cstdio>
#include<bitset>
#define HASH_SIZE 100.
using namespace std;.
void swap(int * a, int * b).
{
int x = *a;.
*a = *b;.
*b = x;.
}

void rotate_array(int a[], int d, int n).
{
int add_factor = n - d; // this is used to find new index for an element.
int I = 0 ; // this is the index of element.
```

int next; // contains next location of element

see more

Reply • Share ›

softvar • 9 months ago

```
/* Paste your code here (You may delete these lines if not writing c
```

Reply • Share ›

softvar • 9 months ago

```
/*
METHOD-1 (using temp array)
===========================

Complexity of this program - O(n)
---------------------------------

Auxillary Space - O(d)
----------------------
*/

#include<stdio.h>

int main()
{
        int arr[]={1,2,3,4,5,6,7};
        const int len = sizeof(arr)/sizeof(arr[0]);
        int d = 2,i,j,k;
```

see more

Reply • Share ›

shek8034 • 11 months ago

One more tricky algo.
1. Reverse first d elements.
2. Reverse next n-d elements.
3. Reverse the whole array.

Ex : {1,2,3,4,5,6} d=2;
After Step 1: {2,1,3,4,5,6}
After Step 2: {2,1,6,5,4,3}
After Step 1: {3,4,5,6,1,2} [done]

Reverse of each subarray in each step can be done by
swapping values at first and last index, as we simply do to reverse an array.
Works for all the cases.
:)

 • Reply • Share ›

shek8034 ➚ shek8034 • 11 months ago

This is also called Reversal Algorithm
http://www.geeksforgeeks.org/p...

 • Reply • Share ›

anonymous • 11 months ago

can anyone plz tell me wht is the significance of taking GCD(n,d)... here :p

2  • Reply • Share ›

Abhinav Aggarwal ➚ anonymous • 11 months ago

Suppose n=10 and d=53545

```
/* Paste your code here (You may delete these lines if not wri
```

 • Reply • Share ›

CodingSkills4u • a year ago

Check this out..
http://www.codingskills4u.com/...

Reply • Share ›

abhishek08aug • a year ago

```
 void leftRotate(int arr[], int d, int n)
{
  int i, j, temp;
  for (i = 0; i < d; i++)
  {
    temp = arr[i];
    arr[i]=arr[d];
    for(j=d; j<=n-2; j++) {
      arr[j]=arr[j+1];
    }
    arr[n-1]=temp;
  }
}
```

Reply • Share ›

abhishek08aug • a year ago

Intelligent :)

Reply • Share ›

Nirdesh • a year ago

How come this algorithm works with the data,
Input arr[] = [1, 2, 3, 4, 5, 6, 7], d = 2, n =7
as here GCD=1,so after 1st rotation ,result will be,
2,3,4,5,6,7,1 but the expected is 3,4,5,6,7,1,2

which is wrong....correct me if its wrong.

```
/* Paste your code here (You may delete these lines if not writing c
```

1 ▲ | ▼ • Reply • Share ›

Purushotham • a year ago

how does the above juggling algo behave for the case n=8 and d =6. In this ca
position will be shifted by just 2 places but not 6 places. Please clarify, if I am r

```
/* Paste your code here (You may delete these lines if not writing c
```

▲ | ▼ • Reply • Share ›

naresh b • a year ago

Here is the block-swap implementation in java (block swap can be replaced w
etc.. method not sure about it)

```
 package com.nbethi;

import java.util.Arrays;

public class BlockSwapArray {

        public static void main(String[] args) {
                int[] array;
                int size = 17;
                array = new int[size];
                for (int i = 0; i < size; i++) {
                        array[i] = i;
                }
                System.out.println(Arrays.toString(array));
```

see more

• Reply • Share ›

Hinidu • a year ago

Simple solution. Time O(n). Memory O(1).

```
 void rotate(int *a, int d, int n)
{
        for (int i = d; i < n; ++i)
                swap(a[i], a[i - d]);
        if (n % d)
                rotate(a + n - d, d - n % d, d);
}
```

1 • Reply • Share ›

poorvank • a year ago

#include
void main()
{
int arr[50],i,j,k,count=0,n,m;
printf("Enter no of elements in array:\n");
scanf("%d",&n);
printf("Enter elements in array:\n");
for(i=0;i<n;i++)
{
scanf("%d",&arr[i]);
}
printf("Enter the amount of rotation:\n");
scanf("%d",&m);

```
{
k=arr[0];
for(i=0;i<n;i++)
{
```

see more

Reply • Share ›

Mohit BHnadari • a year ago

You are given a sorted array [containing no repeated elements] of size N, on w
applied k number of times.
For example, left circular rotation of 1, 2, 3, 4, 5, 6 will give 2, 3, 4, 5,6 , 1
One more left circular rotation will give 3, 4, 5, 6, 1, 2 and so on.
The algorithm for finding the value of k should use O(1) space and should take

Reply • Share ›

Ankit Malhotra • a year ago

or right jump d can be negative and also added check for d > n.

```
 #include <iostream>
using namespace::std;
typedef unsigned long counter;
typedef long element;

void printArray(element term[], counter count)
{
  for (int i = 0; i != count; cout << term[i++] << ' ');
  cout << endl;
}

counter gcd (counter a, counter b)
{
```

```
  if (!b) return a;
  else return gcd (b, a%b);
}
```

see more

• Reply • Share ›

Ankit Malhotra ➦ Ankit Malhotra • a year ago

Handles -ve values of d for right jump and also checks for |d| > n

replace
typedef unsigned long counter;
with
typedef long counter;

• Reply • Share ›

Palash • 2 years ago

Algorithm below works too.
It places the first element in its correct position and then places the element th
we reach position 0 again, then it breaks.

```
#include
void rotate(int *arr, int n, int d)
{
int x=n-d,y=x;
int temp,prev=arr[0];
do
{
temp=arr[y];
arr[y]=prev;
prev=temp;
y=(y+x)%n;
```

}while(y!=x);

return;

see more

• Reply • Share ›

ying ➚ Palash • 2 years ago

when i test this code using
8
1 2 3 4 5 6 7 8
2

the output is wrong: 3 2 5 4 7 6 1 8
correct me if i am wrong.

```
/* Paste your code here (You may delete these lines if not wri
```

• Reply • Share ›

Palash ➚ ying • a year ago

I've rectified the code. Here it goes -

```
void rotate(int *arr, int n, int d)
{
int x=n-d,y;
int i,temp,prev=arr[0],count=0;
for(i=0;i<n;i++)
{
prev=arr[i];
y=(i+x)%n;
do
{
```

```
count++;
temp=arr[y];
arr[y]=prev;
prev=temp;
y=(y+x)%n;

}while(y!=(i+x)%n);
if(count==n)break;
}

return;
}
```

Reply • Share ›

Palash ying • a year ago

Yeah, sorry, my bad.
A small modification makes it work, though. It's still O(n) in time
I'll post the code as soon as I can.

Reply • Share ›

Nishant • 2 years ago

Juggling problem is having O(nd) complexity in worse case scenario i.e. when
please correct if i am wrong.

Reply • Share ›

Nishant Nishant • 2 years ago

Sorry for above comment. I got it now :)

Reply • Share ›

harsh jain • 2 years ago

#include
#include

```
using namespace std;

void rotation( int arr[] , int , int , int , int , int );
int main()
{
int n , k;
cin >> n;
int arr[n];
for ( int i = 0; i > k;
arr[i] = k;
}

int d;
cin >> d;

rotation( arr , n , d , 0 , arr[0] , 0);
```

see more

• Reply • Share ›

Ricky13 • 2 years ago

A simple O(n) solution without any extra space.

```
 /* Paste your code here (You may delete these lines if not writing c
#include <stdio.h>

void arrayrotate(int a[],int n, int d) {
        int temp = a[0];
        int count = 0;
        int i = 0;
        while ( count < n ) {
                a[i] = a[(i+d)%n];
```

```
                i = (i+d)%n;
        }

        a[n-d] = temp;
}
```

see more

Reply • Share ›

ying ➚ Ricky13 • 2 years ago

same testing as the above
when i test this code using
8
1 2 3 4 5 6 7 8
2

the output is wrong: 5 2 7 4 3 6 1 8
correct me if i am wrong.

```
/* Paste your code here (You may delete these lines if not wri
```

Reply • Share ›

pankaj • 2 years ago

Y so complex algorithm. It can be easily done by.
reverse 0 to d, reverse d to n , then again reverse 0 to n.

```
/* Paste your code here (You may delete these lines if not writing c
```

Reply • Share ›

GeeksforGeeks ➚ pankaj • 2 years ago

The reversal algorithm has been published as a separate post (See http://www.geeksforgeeks.org/archives/2838) . This post is to for one r

• Reply • Share ›

mAc • 2 years ago

There is one more method which i read somewhere.
pseudo code...

Step1:reverse entire array.
Step2:reverse the first 'd' elements.
Step3:reverse the last 'n-d' elements.

```
/* Paste your code here (You may delete these lines if not writing c
```

• Reply • Share ›

mAc ➚ mAc • 2 years ago

sorry...it was already posted...

• Reply • Share ›

geekva • 2 years ago

I guess this should work.

```
i=0,count=0; // a,b,temp are variables (for intermediate
             // book-keeping)
    while(count!=n)
    {
         if(d>i)
         {
              b=n+i-d;
              if(count==0) a=arr[i];
```

```
            a=arr[b];
            arr[b]=temp;
            i=b;

        }
        else
        {
```

see more

Reply • Share ›

how to rotate right by jugglin • 3 years ago

how to rotate right by juggling

Reply • Share ›

bala how to rotate right by jugglin • 3 years ago

```
          k = j + d;
        if (k >= n)
          k = k - n;
```

This particular section of code will become

```
          k = j - d;
        if (k < 0)
          k = n + k; //Remember k is negative
```

Reply • Share ›

Deepak • 3 years ago

Well, the better option is use of linked list

Reply • Share ›

Load more comments

Subscribe Add Disqus to your site