Design and analysis of Rotation Algorithm

Assignment no.1

Submitted by: Group-15

Guided by: Md. Meraz Sir
Professor: Dr. Mohammed Javed

TABLE OF CONTENTS

PROBLEM STATEMENT

Write a C program to left rotate and right rotate an array.

HOW WE HAVE SOLVED THE PROBLEM!

To rotate an array ar[ ] of size ‘x’, either leftwards or rightwards for ‘n’ times, we have used the concept of loops and functions. To begin with we take the input from the user. Our algorithm constitutes a nested loop. First loop ensures the rotation is carried out as many no. of times as the user wishes to and the second ‘nested loop’ carries out the process of array rotation by assigning the values to consecutive elements, depending upon the type of rotation. Further, we have applied the algorithm using two methods, namely, the standard method and the functions method.

Introduction

Here is an explanation to what exactly right and left array rotation means!

ORIGINAL ARRAY

58	59	60	61	62

LEFT ROTATED ARRAY (1 TIME)

59	60	61	62	58

RIGHT ROTATED ARRAY (1 TIME)

62	58	59	60	61

OUR ALGORITHM

RIGHT ROTATION

- Firstly, we execute ‘for’ loop for the desired number of rotations of the array.
- ar =[1, 2, 3, 4, 5, 6] (Original array)
- Then, we store the last element of the array in a different variable, say, t = ar[5] =6.
- After that we execute ‘for’ loop in such a way that every element of the array shifts towards right, overwriting the previous value.

  ar =[1, 1, 2, 3, 4, 5] (ar[x-1]=ar[x-2], starting from x=6)
- Now, first two elements of the array are same.
- In order to solve this problem, we assign the value of ‘t’ to the first position of the array.

  t = ar[0] =6.
- Hence, Right Rotation of the given array is completed

  ar =[6, 1, 2, 3, 4, 5].

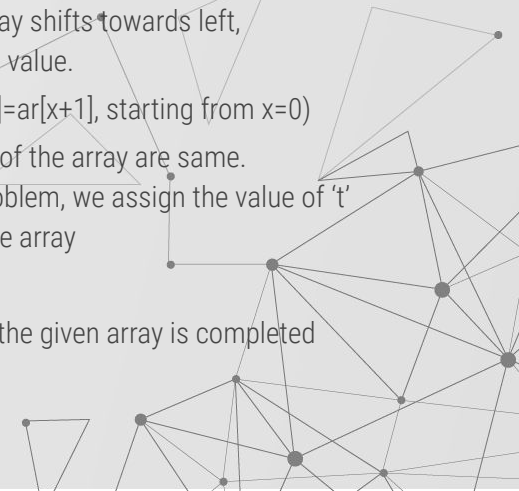
LEFT ROTATION

- Firstly, we execute ‘for’ loop for the desired number of rotations of the array.
- ar =[1, 2, 3, 4, 5, 6] (Original array)
- Then, we store the first element of the array in a different variable, say, t = ar[0] =1.
- After that we execute ‘for’ loop in such a way that every element of the array shifts towards left, overwriting the previous value.

  ar =[2, 3, 4, 5, 6, 6] (ar[x]=ar[x+1], starting from x=0)
- Now, last two elements of the array are same.
- In order to solve this problem, we assign the value of ‘t’ to the last position of the array

  t = ar[5] =1 .
- Hence, Left Rotation of the given array is completed

  ar =[2, 3, 4, 5, 6, 1].



PSEUDOCODE

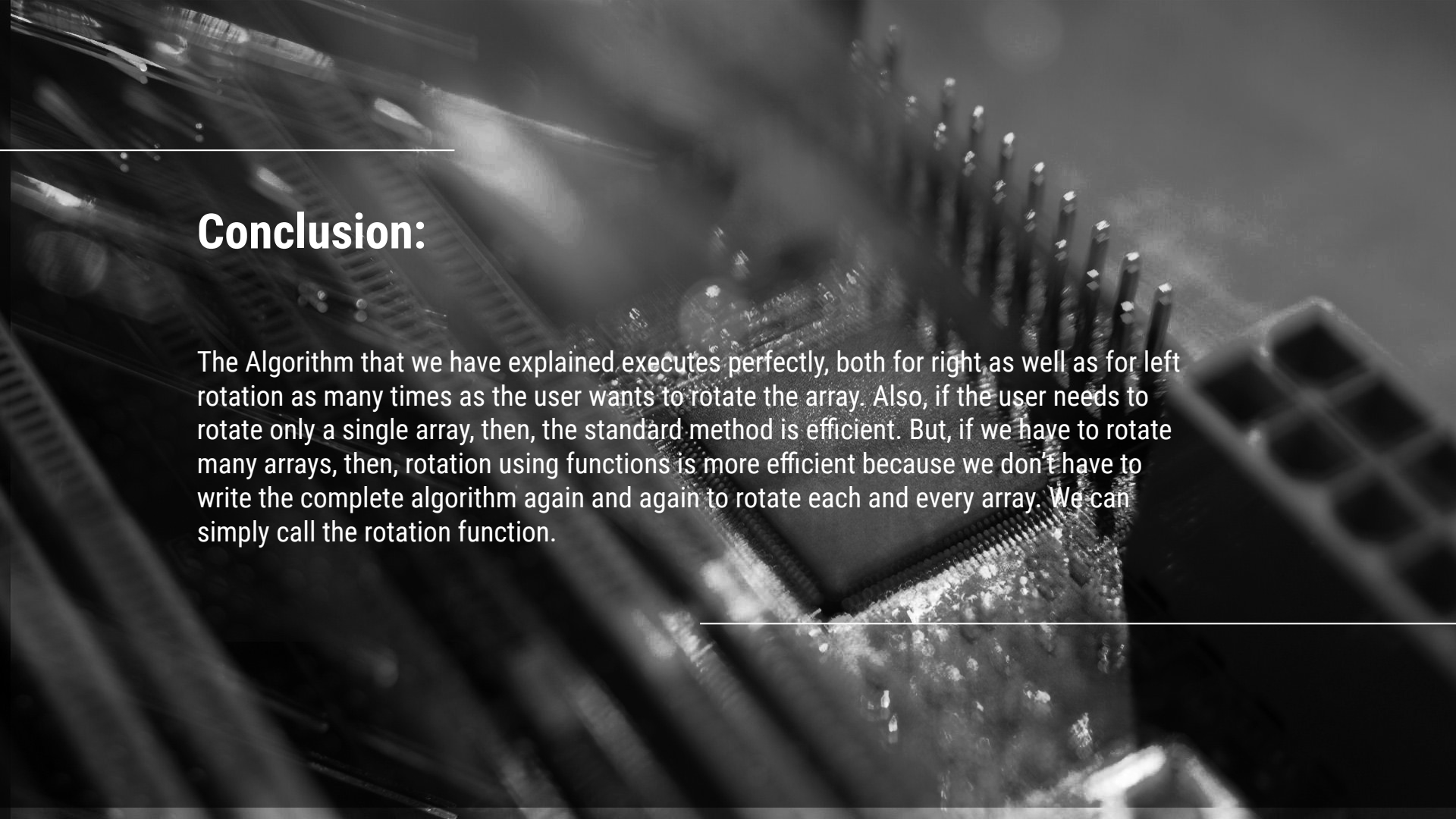
The standard method:

```
BEGIN
NUMBER x, ar[1000], i, j, t
CHARACTER c
OUTPUT "Enter the size of the array"
    INPUT x
OUTPUT "Enter elements of the array"
    FOR i=0 to i<x STEP 1
    INPUT ar[i]
    END FOR
OUTPUT "Enter number of times array needs to be rotated"
    INPUT n
OUTPUT "Enter 'r/R' to right rotate the array
    Enter 'l/L' to left rotate the array"
    INPUT c
//ALGORITHM TO ROTATE
   //RIGHT ROTATION
IF c ='r/R' THEN
    FOR i=1 to i<=n STEP 1 THEN
        STORE ar[0] in t
        FOR j =1 to J<x-1 STEP -1
        ar[j] = ar[j-1]
        END FOR
    END FOR
   //LEFT ROTATION
ELSE IF c ='l/L' THEN
    FOR i=1 to i<=n STEP 1 THEN
        STORE ar[0] in t
        FOR j=0 to j<x-1 STEP 1
        ar[j] = ar[j+1]
        ar[x-1] = t
        END FOR
    END FOR
ELSE OUTPUT "Wrong Output"
//OUTPUT PROCESS
OUTPUT "Rotated array-"
    FOR i=0 to i<x STEP 1
    OUTPUT ar[i]
END
```

RUNNING TIME OF THE ALGORITHM

NO. OF ROTATIONS

SIZE OF ARRAY	100	10000	1000000	100000000
7	0.003033	0.005014	0.015202	0.393884
9	0.005468	0.005567	0.012255	0.637720
11	0.006170	0.006818	0.0234402	0.653512



Conclusion:

The Algorithm that we have explained executes perfectly, both for right as well as for left rotation as many times as the user wants to rotate the array. Also, if the user needs to rotate only a single array, then, the standard method is efficient. But, if we have to rotate many arrays, then, rotation using functions is more efficient because we don't have to write the complete algorithm again and again to rotate each and every array. We can simply call the rotation function.

THANK YOU

GROUP - 15
IEC2021058, IEC2021059, IEC2021061, IEC2021062