Student Name:- Prameet Upadhyay

Student Roll No.:- 1905692

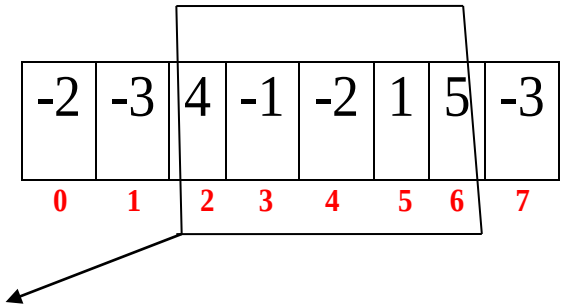
Algorithm Lab. Class Assignment-5

CSE Group 1

Date: - 6th August 2021

1. **Write a C program to find the sum of contiguous subarray within a one dimensional (1-D) array of numbers which has the largest sum. Find the time complexty of your program.**

Example



4 + (-1) + (-2) + 1 + 5 = 7

So the maximum contiguous subarray sum is 7

Program

```c
#include <stdio.h>
#include<time.h>
#include<limits.h>

int algo(int arr[], int n)
{
    int max_sum = INT_MIN;
    int curr_sum = 0;
    for (int i = 0; i < n; i++)
    {
        curr_sum += arr[i];
        if (max_sum < curr_sum)
            max_sum = curr_sum;
```

```
        if (curr_sum < 0)
            curr_sum = 0;
    }
    return max_sum;
}
int main()
{
    time_t strt, end;
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    strt = clock();
    int ans = algo(arr, n);
    end = clock();
    double t = end - strt;
    printf("The maximum contigous sum is %d", ans);
    printf("\nTime : %f",(t/CLOCKS_PER_SEC));
}
```

Output

```
8
-2 -3 4 -1 -2 1 5 -3
The maximum contigous sum is 7
Time : 0.000003
```

2. **Write a program to find out the largest difference between two elements A[i] and A[j] (A[j]-A[i]) of the array of integers A in O(n) time such that j > i. For example: Let A is an array of integers:**

int[] a = { 10, 3, 6, 8, 9, 4, 3 };

if i=1, j=3, then diff = a[j] – a[i] = 8 – 3 = 5

if i=4, j=6, then diff = a[j] – a[i] = 3 – 9 = -6

.....

.....

if i=1, j=4, then diff = a[j] – a[i] = 9 – 3 = 6

.....

.....

6 is the largest number between all the differences, that is the answer.

Find the time complexty of your program.

Program

```c
#include <stdio.h>
#include<time.h>

int algo(int arr[], int n)
{
    int max_diff = arr[1] - arr[0];
    int min = arr[0];
    for (int i = 1; i < n; i++)
    {
        int curr_diff = arr[i] - min;
        if (max_diff < curr_diff)
            max_diff = curr_diff;
        if (arr[i] < min)
            min = arr[i];
    }
```

```
    return max_diff;
}
```

```
int main()
{
    time_t strt, end;
    int n;
    scanf("%d", &n);
    int arr[n];
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
    strt = clock();
    int ans = algo(arr, n);
    end = clock();
    double t = end - strt;
    printf("The maximum differnece  is %d", ans);
    printf("\nTime : %f",(t/CLOCKS_PER_SEC));
    return 0;
}
```

Output

```
7
10 3 6 8 9 4 3
The maximum differnece  is 6
Time : 0.000003
```

3. **Find the GCD and LCM of n numbers where (n>=2).**

Program

```c
#include <stdio.h>

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

```c
int findGCD(int A[], int n)
{
    int result = A[0];
    for (int i = 1; i < n; i++)
    {
        result = gcd(A[i], result);
        if (result == 1)
        {
            return 1;
        }
    }
    return result;
}
int findlcm(int A[], int n)
{
    int ans = A[0];
    for (int i = 1; i < n; i++)
        ans = (((A[i] * ans)) /(gcd(A[i], ans)));
    return ans;
}
```

```c
int main()
{
    int n;
    scanf("%d",&n);
```

```
    int A[n];
    for(int i =0;i<n;i++){
        scanf("%d",&A[i]);
    }
    printf("GCD: %d\n",findGCD(A,n));
    printf("LCM: %d",findlcm(A,n));
    return 0;
}
```

Output

```
PS C:\Users\Prameet Upaddhyay\Desktop\CODES\DAA_lab\6aug> gcc 3.c
PS C:\Users\Prameet Upaddhyay\Desktop\CODES\DAA_lab\6aug> ./a.exe
5
3 4 7 2 12
GCD: 1
LCM: 84
```

4. Consider an n × n matrix A = (a_{ij}), each of whose elements a_{ij} is a nonnegative real number, and suppose that each row and column of A sums to an integer value. We wish to replace each element a_{ij} with either ⌉ a_{ij} ⌉ or ⌋ a_{ij} ⌋ without disturbing the row and column sums. Here is an example:

$$\begin{pmatrix} 10.9 & 2.5 & 1.3 & 9.3 \\ 3.8 & 9.2 & 2.2 & 11.8 \\ 7.9 & 5.2 & 7.3 & 0.6 \\ 3.4 & 13.1 & 1.2 & 6.3 \end{pmatrix} \rightarrow \begin{pmatrix} 11 & 3 & 1 & 9 \\ 4 & 9 & 2 & 12 \\ 7 & 5 & 8 & 1 \\ 4 & 13 & 2 & 6 \end{pmatrix}$$

Write a program by defining an user defined function that is used to produce the rounded matrix as described in the above example. Find out the time complexity of your algorithm/function.

Program

```c
#include <stdio.h>

int roundNo(float num)
{
    return num < 0 ? num - 0.5 : num + 0.5;
}
```

```c
void algo(float A[3][4])
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            A[i][j] = roundNo(A[i][j]);
        }
    }
```

```c
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            printf("%0.2f ",A[i][j]);
        }
        printf("\n");
    }
}
```

```c
int main()
{
    float A[3][4];
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 4; j++)
        {
```

```
            scanf("%f", &A[i][j]);
        }
    }
    algo(A);
    return 0;
}
```

Output

```
PS C:\Users\Prameet Upaddhyay\Desktop\CODES\DAA_lab\6aug> ./a.exe
3.2
3.1
2.1
4.5
2.9
2.1
5.7
4.3
9.8
11.2
2.6
6.7
3.00 3.00 2.00 5.00
3.00 2.00 6.00 4.00
10.00 11.00 3.00 7.00
```