



INSTITUTO POLITÉCNICO  
NACIONAL

ESCUELA SUPERIOR DE  
CÓMPUTO



ANALISIS DE ALGORITMOS

PROFESOR TITULAR: FRANCO MARTINEZ  
EDGARDO ADRIAN

1° PARCIAL

EJERCICIOS #3:  
SIMULACION PRODUCTO2MAYORES

LEMUS RUIZ MARIANA ELIZABETH  
2020630211

GRUPO: 3CM12



## EJERCICIOS 03: SIMULACIÓN PRODUCTO2MAYORES

### INSTRUCCIONES:

Para el algoritmo analizado por casos en clase y video lección "Producto2Mayores", realice la simulación de su mejor, peor y caso medio; realizando las modificaciones y adaptaciones necesarias para verificar los tres casos en  $n=2500$  y  $n=5000$  considerando al menos 10,000 iteraciones del algoritmo con cada  $n$  y diferente distribución de los números.

- Para el mejor caso basta con tener un archivo de números que coloque en los dos primeros números a los dos mayores
- Para el peor caso basta con tener un archivo ordenado ascendentemente para cada  $n$
- Para el caso medio se deberán de hacer al menos 10,000 iteraciones para cada  $n$  generando arreglos aleatorios en cada iteración y comprobar el número de operaciones básicas promedios totales para enfrentarlas al modelo del caso medio.

### PSEUDOCODIGO:

```
func Producto2Mayores(A,n)
    if(A[1] > A[2])
        mayor1 = A[1];
        mayor2 = A[2];
    else
        mayor1 = A[2];
        mayor2 = A[1];
    i = 3;

    while(i <= n)
        if(A[i] > mayor1)
            mayor2 = mayor1;
            mayor1 = A[i];
        else if (A[i] > mayor2)
            mayor2 = A[i];
        i = i + 1;

    return = mayor1 * mayor2;
```

→Primer comparación  
→Asignacion  
→Asignacion  
→Asignacion  
→Asignacion  
→n-2 Comparaciones  
→Asignacion  
→Asignacion  
→\*Si  $A[i] \leq \text{mayor2}$   
→Asignacion

### MEJOR CASO

$$\begin{aligned} F(n) &= 1 + (n - 2) + (n - 2) + 2 \\ &= 3 + 2(n - 2) \\ &= 2n - 1 \end{aligned}$$

## CODIGO DEL PROGRAMA:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#define VECES 10000
int Producto2Mayores(int *A, int n)
{
    int mayor1, mayor2, i, contador = 0;
    contador++; //Condicion
    if (A[0] > A[1]){
        mayor1 = A[0];
        contador++; //Asignacion
        mayor2 = A[1];
        contador++; //Asignacion
    }
    else{
        mayor1 = A[1];
        mayor2 = A[0];
    }
    i = 2;
    while (i < n){
        contador++; //Condicion
        if (A[i] > mayor1){
            mayor2 = mayor1;
            mayor1 = A[i];
        }
        else{
            contador++; //Condicion
            if (A[i] > mayor2){
                mayor2 = A[i];
            }
        }
        i++;
    }
    return contador;
    //return mayor1*mayor2
}

int main(int narg, char **varg)
{
    int n;
    int *A;
    double promedio = 0;
    int auxiliar;

    if (narg != 2)
    {
        printf("\nIntroduce una n");
        scanf("%d",&n);
        exit(1);
    }
    n = atoi(varg[1]);
    A = malloc(sizeof(int) * n);

    for (int i = 0; i < n; i++)
```

```
{
    scanf("%d", &A[i]);
}
auxiliar = Producto2Mayores(A, n);
promedio = promedio + (float)auxiliar;
printf("El promedio final es %lf", promedio);

return 0;
}
```

## PEOR CASO

$$\begin{aligned} F(n) &= 1 + (n - 2) + 2(n - 2) + 2 \\ &= 3 + 3(n - 2) \\ &= 3n - 3 \end{aligned}$$

CODIGO DEL PROGRAMA:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int Producto2Mayores(int *A, int n){
    int mayor1, mayor2, i, contador = 0;
    contador++; //Condicion
    if (A[0] > A[1]) {
        mayor1 = A[0];
        contador++; //Asignacion
        mayor2 = A[1];
        contador++; //Asignacion
    }
    else{
        mayor1 = A[1]; //Asignacion
        contador++;
        mayor2 = A[0];
        contador++; //Asignacion
    }
    i = 2;
    while (i < n) {
        contador++; //Condicion
        if (A[i] > mayor1) {
            mayor2 = mayor1;
            contador++; //Asignacion
            mayor1 = A[i];
            contador++; //Asignacion
        }
        else{
            if (A[i] > mayor2){
                mayor2 = A[i];
            }
        }
        i++;
    }
    return contador;
}
```

```
int main(int narg, char **varg)
{
    int n;
    int *A;
    double promedio = 0;
    int auxiliar;

    if (narg != 2)
    {
        /*printf("\nIntroduce una n");
        exit(1);*/
    }
    n = atoi(varg[1]);
    A = malloc(sizeof(int) * n);

    for (int i = 0; i < n; i++)
    {
        scanf("%d", &A[i]);
    }

    auxiliar = Producto2Mayores(A, n);
    promedio = promedio + (float)auxiliar;
    printf("Promedio final: %lf", promedio);

    return 0;
}
```

## CASO MEDIO

**PARA:  $A[i] > \text{MAYOR } 1$**

$$\frac{1}{3}(1+2+(n-2)+2(n-2)) = \frac{1}{3}(3n-3)$$

**PARA:  $A[i] \leq \text{MAYOR } 1$  &&  $A[i] > \text{MAYOR } 2$**

$$\frac{1}{3}(1+2+2(n-2)+(n-2)) = \frac{1}{3}(3n-3)$$

**PARA:  $A[i] \leq \text{MAYOR } 1$  &&  $A[i] \leq \text{MAYOR } 2$**

$$\frac{1}{3}(1+2+(n-2)+(n-2)) = \frac{1}{3}(2n-1)$$

## CASO MEDIO

$$F(n) = \frac{1}{3}(2(3n-3) + (2n-1)) = \frac{1}{3}(8n-7)$$

CODIGO DEL PROGRAMA:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define VECES 10000
int Producto2Mayores(int *A, int n)
{
    int mayor1, mayor2, i, contador = 0;
```

```
    contador++; //Condicion
    if (A[0] > A[1])
    {
        mayor1 = A[0];
        contador++; //Asignacion
        mayor2 = A[1];
        contador++; //Asignacion
    }
    else
    {
        mayor1 = A[1];
        contador++; //Asignacion
        mayor2 = A[0];
        contador++; //Asignacion
    }
    i = 2;
    while (i < n)
    {
        contador++; //Condicion
        if (A[i] > mayor1)
        {
            mayor2 = mayor1;
            contador++; //Asignacion
            mayor1 = A[i];
            contador++; //Asignacion
        }
        else
        {
            contador++; //Condicion
            if (A[i] > mayor2)
            {
                mayor2 = A[i];
                contador++; //Asignacion
            }
        }
        i++;
    }
    return contador;
}

int main(int narg, char **varg)
{
    int n,i,k;
    int *A;
    int promedio = 0;
    int auxiliar;

    if (narg != 2)
    {
        printf("\nIntroduce una n");
        exit(1);
    }
    n = atoi(varg[1]);
    A = malloc(sizeof(int) * n);

    srand(time(NULL));
```

```

for ( k = 0; k < VECES; k++)
{
    for (i = 0; i < n; i++)
    {
        *(A + i) = rand() % ((1000 * n));
        //printf("Vuelta %d. Numero %d\n",k,A[i]);
    }
    auxiliar = Producto2Mayores(A, n);
    promedio = promedio + (float)auxiliar;
    //printf("Iteracion %d, el contador vale %d\n",k,auxiliar);
}
//printf("El promedio es %lf\n", promedio);
promedio = promedio / VECES;
printf("El promedio es %d", promedio);

return 0;
}

```

#### TABLA COMPARATIVA DE RESULTADOS

	PRACTICO	TEORICO
PARA n= 2500		
MEJOR CASO	4997.0000	4999.0000
PEOR CASO	7497.0000	7497.0000
CASO MEDIO	5012.0000	6664.3333
PARA n= 5000		
MEJOR CASO	9997.0000	9999.00000
PEOR CASO	14997.0000	14997.00000
CASO MEDIO	10014.0000	13331.0000