

## Laboratorio Nro. 2

# COMPLEJIDAD DE ALGORITMOS

**Andrea Sánchez cortes**  
Universidad Eafit  
Medellín, Colombia  
asanchezc1@eafit.edu.co

### 3) Simulacro de preguntas de sustentación de Proyectos

#### 3.1

En la columna 1 encontramos 20 tamaños de medidas y en la columna 2 el tiempo correspondiente a cada tamaño, teniendo en cuenta que este tiempo esta medido en milisegundos para eso utilizamos 2 algoritmos:

#### INSERTION SORT :

Medida	Tiempo
5000	40
10000	38
15000	229
20000	119
25000	260
30000	210
35000	233
40000	352
45000	810
50000	1383
55000	690
60000	941
65000	1129
70000	939
75000	1378
80000	2195
85000	1983
90000	2258
95000	2412
100000	5098

**PhD. Mauricio Toro Bermúdez**  
Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

**ESTRUCTURA DE DATOS 1**  
**Código ST0245**

**MERGE SORT:**

<b>Medida</b>	<b>Tiempo</b>
1000000	198
2000000	850
3000000	580
4000000	761
5000000	808
6000000	954
7000000	1484
8000000	1281
9000000	1366
10000000	1570
11000000	1628
12000000	1954
13000000	2512
14000000	2308
15000000	2855
16000000	2712
17000000	3057
18000000	3301
19000000	4711
20000000	4126

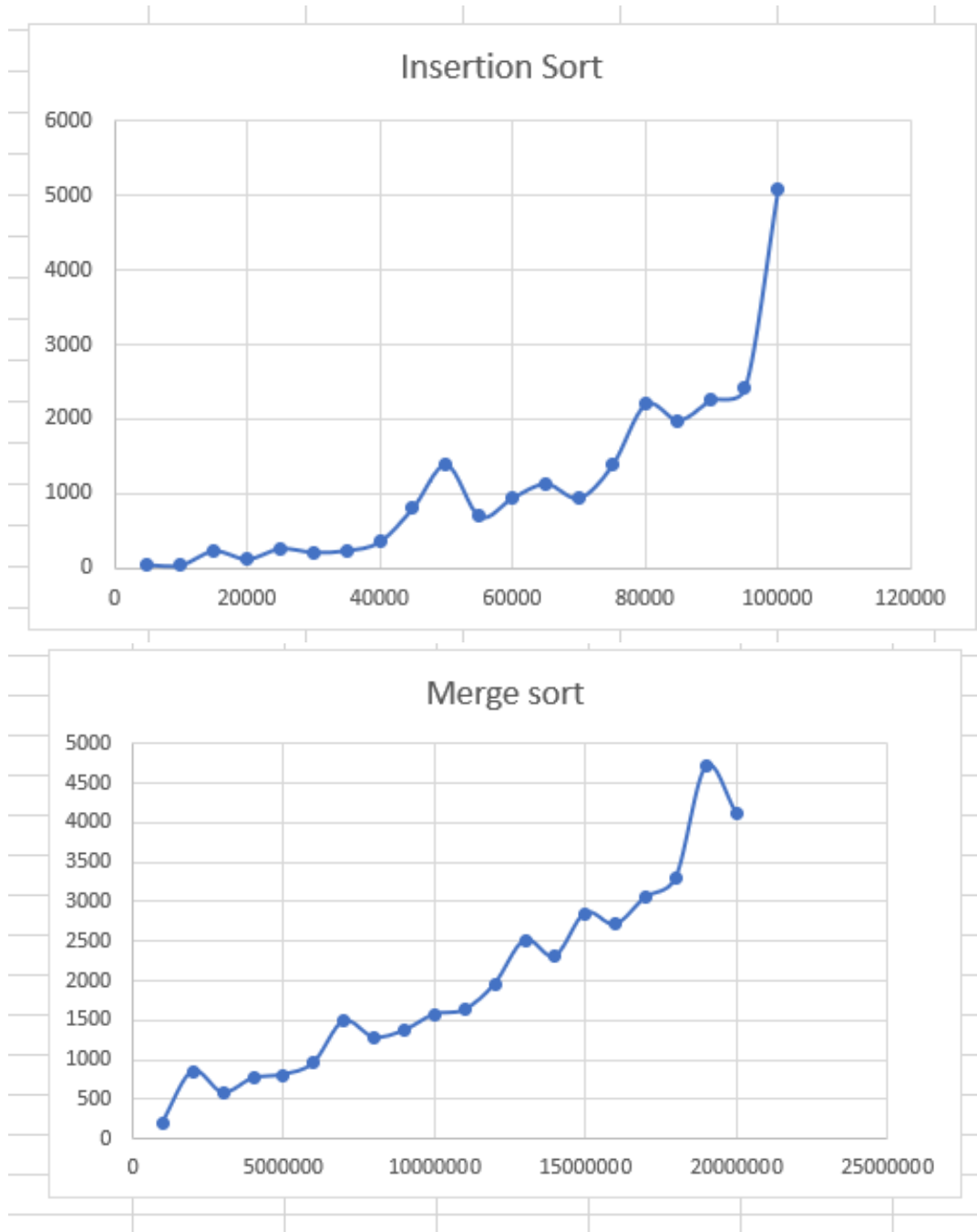
**3.2**

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 1

### Código ST0245



3.3 lo que observamos en los 2 algoritmos el mas eficiente seria el mergesort ya que es más eficiente de usar por que separa matriz por matriz haciéndolo mas optimo y eficiente que el insertionsort ya que recorre cada posición de la matriz haciendo que el tiempo se ejecute mas lento. Su complejidad es  $O(n \times \log(n))$  y la de insertionsort es  $O(n^2)$  esto significa que se demora menos en ejecutarse con matrices muy grandes.

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
 Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
 Tel: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 1

### Código ST0245

3.4 el merge sort divide el arreglo en 2 usando recursión por lo que su complejidad seria  $O(n/2)$  después de que pase a un complejidad asintótica se convierte en un logaritmo

### 3.5 Array 2

Su complejidad  $O = O(n)$

```
public int countEvens(int[] nums) {
    int cont = 0;
    for (int i = 0; i < nums.length; i++) {
        if (nums[i] % 2 == 0)
            cont++;
    }
    return cont;
}
```

Su complejidad  $O = O(n)$

```
public int sum13(int[] nums) {
    int a = 0;
    for (int i = 0; i < nums.length; i++) {
        if (nums[i] == 13 && i < nums.length - 1) {
            nums[i + 1] = 0;
        }
        if (nums[i] != 13) {
            a += nums[i];
        }
    }
    return a;
}
```

Su complejidad  $O = O(n)$

```
public boolean sum28(int[] nums) {
    int x = 0;
    for (int i = 0; i < nums.length; i++) {
        if (nums[i] == 2) {
            x += 2;
        }
    }
    return x == 8;
}
```

Su complejidad  $O = O(n)$

```
public int bigDiff(int[] nums) {
    int a = nums[0];
    int b = nums[0];
    for (int i = 0; i < nums.length; i++) {
        a = Math.max(a, nums[i]);
        b = Math.max(b, nums[i]);
    }
    return a - b;
}
```

Su complejidad  $O = O(n)$

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 1

### Código ST0245

```
public boolean has12(int[] nums) {
    boolean h1 = false;
    boolean h2 = false;
    for (int i = 0; i < nums.length; i++){
        if (nums[i] == 1){
            h1 = true;
        }
        if (nums[i] == 2 && h1){
            h2 = true;
        }
    }
    return h2;
}
```

### Array 3

Su complejidad es  $O(n^2)$

```
public int maxSpan(int[] nums) {
    int c = 0;
    int d = 0;
    for (int i = 0; i < nums.length; i++){
        for (int j = nums.length-1; j >= 0; j--){
            if (nums[i] == nums[j]){
                d = j-i+1;
                if (d > c) c = d;
            }
        }
    }
    return c;
}
```

Su complejidad es  $O(n^2)$

```
public boolean linearIn(int[] outer, int[] inner) {
    int a = 0;
    for (int i = 0; i < inner.length; i++){
        for (int j = 0; j < outer.length; j++){
            if (i < inner.length){
                if (inner[i] == outer[j]){
                    a++;
                    i++;
                }
            }
        }
    }
    return a == inner.length;
}
```

Su complejidad es  $O(n^2)$

```
public int [] seriesUp(int n){
    int [] a = new int[n*(n+1)/2];
    int b = 0;
    for (int i = 1; i < n; i++){
        for (int j = 1; j <= i; j++){
            a[b++] = j;
        }
    }
    return a;
}
```

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
 Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
 Tel: (+57) (4) 261 95 00 Ext. 9473

## ESTRUCTURA DE DATOS 1

### Código ST0245

Su complejidad es  $O(n^2)$

```
public int[] squareUp(int n){
    int[] a= new int [n*n];
    int b= 0;
    for (int i =1; i<=n; i++){
        for (int j = n; j>=1; j--){
            a[b++]=j;
            if(i<j)
                a[b-1] = 0;
        }
    }
    return a;
}
```

Su complejidad es  $O(n^2)$

```
public int countClumps(int [] nums){
    int a =0;
    int b = -1;
    for (int i = 0; i < nums.length -1; i++){
        if (nums[i] == nums[i+1]&& nums[i] != b){
            b = nums[i];
            a++;
        }
        for (int j=0; j <nums.length -1; j++){
            if (nums[i] !=b){
                b= -1;
            }
        }
    }
    return a;
}
```

3.6 lo que significa “n” es la cantidad que hay de elementos del arreglo dado

#### 4) Simulacro de Parcial

- 4.1 C
- 4.2 D
- 4.3 B
- 4.4 B
- 4.5 D
- 4.6 10s
- 4.7 Todas son verdaderas
- 4.8 A
- 4.9 A
- 4.10 C
- 4.11 C
- 4.12 B
- 4.13 C
- 4.14 A

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
 Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
 Tel: (+57) (4) 261 95 00 Ext. 9473