

Laboratorio Nro. 2

Notación O grande

Daniel Alejandro Hincapié Sánchez
Universidad Eafit
Medellín, Colombia
dahincapis@eafit.edu.co

Anthony García Moncada
Universidad Eafit
Medellín, Colombia
agarciam@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1

Tabla de datos para los algoritmos

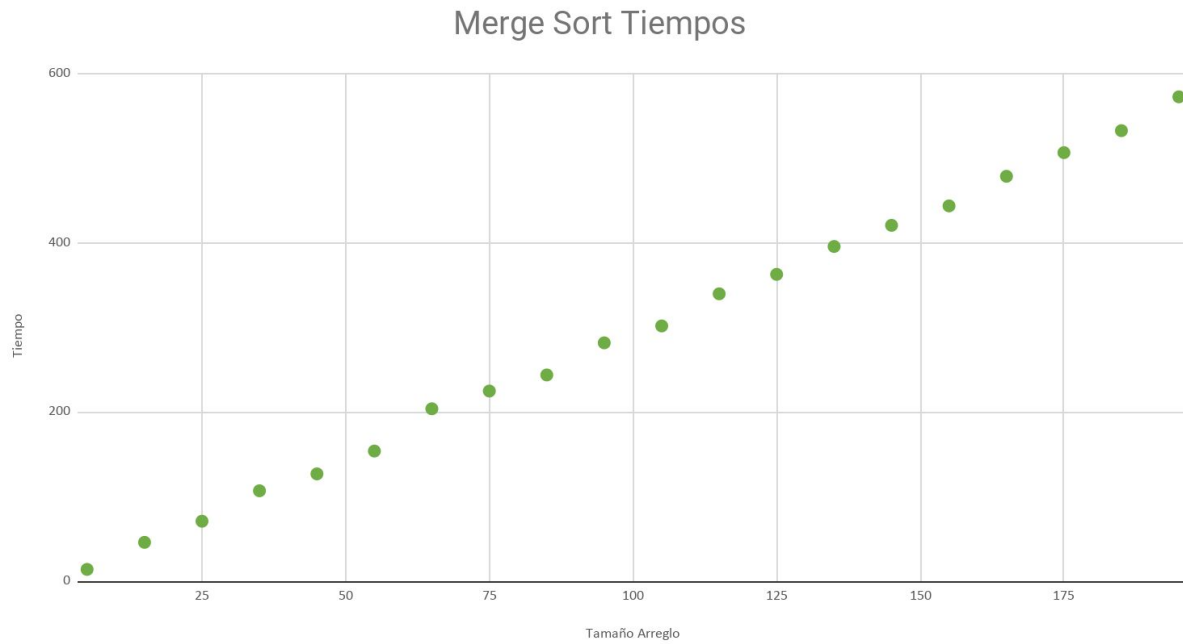
Tamaño Arreglo	Tiempo Insertion	Tiempo Merge
5	8	14
15	24	46
25	40	71
35	51	107
45	66	127
55	93	154
65	96	204
75	108	225
85	130	244
95	135	282
105	151	302
115	173	340
125	173	363
135	182	396
145	186	421
155	221	444
165	228	479
175	243	507
185	258	533
195	273	573

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

3.2

Gráfica Merge Sort

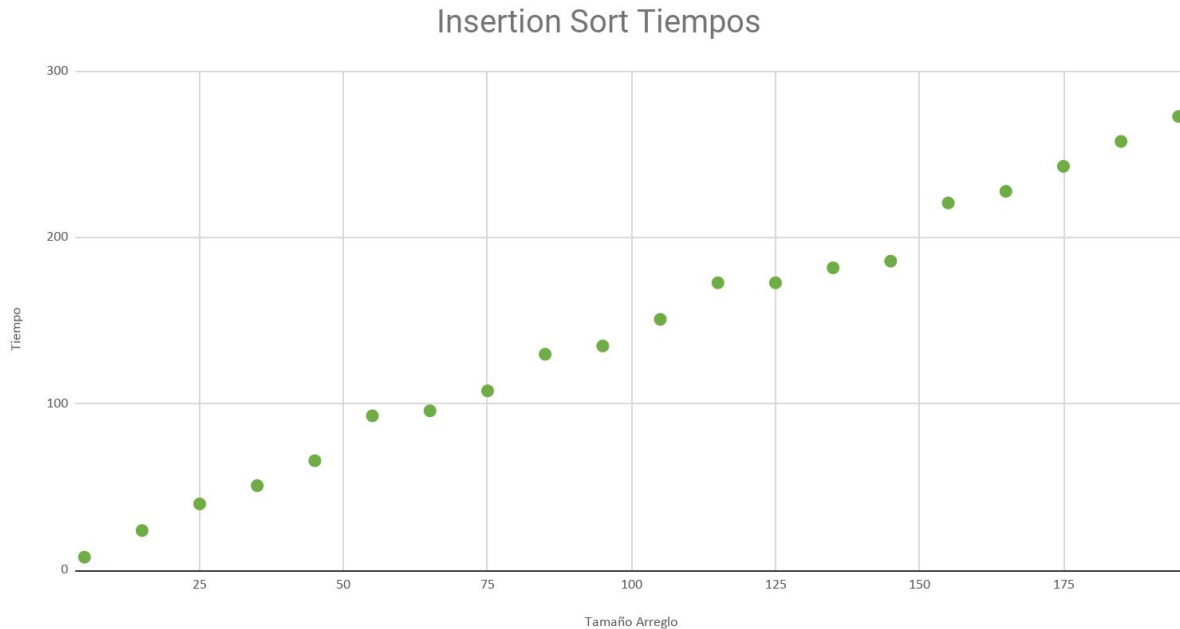


PhD. Mauricio Toro Bermúdez
Docente | Escuela de Ingeniería | Informática y Sistemas
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627
Tel: (+57) (4) 261 95 00 Ext. 9473

ESTRUCTURA DE DATOS 1

Código ST0245

Gráfica Insertion Sort



- 3.3** En comparación a Insertion Sort, Merge Sort es un algoritmo poco eficiente en cuanto a arreglos grandes se refiere, esto se debe a las altas cantidades de tiempo que toma, que llegan incluso a duplicar los tiempos empleados por el insertion sort.
- 3.4** A pesar de que Insertion Sort es más rápido que Merge Sort, sigue sin ser lo suficientemente rápido para ser efectivo a la hora de trabajar con arreglos muy grandes, por lo tanto, para un videojuego que implemente millones de objetos, no es buena idea utilizar este método de ordenamiento.
- 3.5** En general, sin importar el tamaño de los datos, el método Insertion es mucho más rápido que el Merge, por lo cual siempre será mucho más rápido, especialmente cuando se trata de arreglos de gran tamaño.
- 3.6** El algoritmo del ejercicio maxSpan funciona de forma muy simple. Este problema se puede resolver fácilmente sin ciclos, y es que solo son necesarias tres condiciones de retorno para satisfacer los requerimientos planteados por el ejercicio. En dichas tres condiciones se plantea lo siguiente, si el primer y el último int del arreglo son iguales, se retorna el número total de elementos del array, si el tamaño de elementos es 0 o 1, se retornan estos valores

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

respectivamente, y si no se cumplen estas dos condiciones, se retorna el número total de elementos restando uno.

3.7 countEvents:

```
public int countEvens(int[] nums) {
    int cont = 0; //C1
    for(int i = 0; i < nums.length; i++) { //C2*n
        if(nums[i] % 2 == 0) { //C3*n
            cont = cont + 1; //C4*n
        }
    }
    return cont; //C5
}
```

$T(n) = C1 + n*(C2+C3+C4)+C5$, para el peor de los casos

$T(n) = C1 + C2*n$

$T(n)$ es $O(C1+C2*n)$

$T(n)$ es $O(C2*n)$

$T(n)$ es $O(n)$

La variable n representa el tamaño del arreglo.

bigDiff:

```
public int bigDiff(int[] nums) {
    int vm = nums[0]; //C1
    int vM = 0; //C2
    for(int i = 0; i < nums.length; i++) { //C3*n
        if(nums[i] >= vM) { //C4*n
            vM = nums[i]; //C5*n
        }
    }
    for(int i = 0; i < nums.length; i++) { //C6*n
        if(nums[i] <= vm) { //C7*n
            vm = nums[i]; //C8*n
        }
    }
    return vM - vm; //C9
}
```

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

$T(n) = C_1 + C_2 + C_3 \cdot n$, para el peor de los casos

$T(n) = C_1 + C_2 \cdot n$

$T(n)$ es $O(C_1 + C_2 \cdot n)$

$T(n)$ es $O(C_2 \cdot n)$

$T(n)$ es $O(n)$

La variable n representa el tamaño del arreglo.

centeredAverage:

```
public int centeredAverage(int[] nums) {
    int may = nums[0]; //C1
    int men = nums[0]; //C2
    int sum = 0; //C3
    for(int i = 0; i < nums.length; i++) { //C4*n
        sum = sum + nums[i]; //C5*n
        if(nums[i] < men) //C6*n
            men = nums[i]; //C7*n
        if(nums[i] > may) //C8*n
            may = nums[i]; //C9*n
    }
    return (sum - (may + men)) / (nums.length - 2); //C10
}
```

$T(n) = C_1 + C_2 \cdot n$, para el peor de los casos

$T(n)$ es $O(C_1 + C_2 \cdot n)$

$T(n)$ es $O(C_2 \cdot n)$

$T(n)$ es $O(n)$

La variable n representa el tamaño del arreglo.

sum13: $T(n)$ es $O(n)$

```
public int sum13(int[] nums) {
    int cont = 0; //C1
    for(int i = 0; i < nums.length; i++) { //C2*n
        if(nums[i] == 13) { //C3*n
            i++; //C4*n
        } else {
            cont += nums[i]; //C5*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

```

    }
    return cont; //C6
}

```

$T(n) = C1 + C2 \cdot n$, para el peor de los casos

$T(n)$ es $O(C1 + C2 \cdot n)$

$T(n)$ es $O(C2 \cdot n)$

$T(n)$ es $O(n)$

La variable n representa el tamaño del arreglo.

sum67:

```

public int sum67(int[] nums) {
    int cont = 0; //C1
    for(int i = 0; i < nums.length; i++) { //C2*n
        if(nums[i] != 6) { //C3*n
            cont += nums[i]; //C4*n
        } else {
            while(nums[i] != 7) {
                i++;
            }
        }
    }
    return cont; //C5
}

```

$T(n) = C1 + C2 \cdot n$, para el peor de los casos

$T(n)$ es $O(C1 + C2 \cdot n)$

$T(n)$ es $O(C2 \cdot n)$

$T(n)$ es $O(n)$

La variable n representa el tamaño del arreglo.

4) Simulacro de Parcial

4.1 C

4.2 D

4.3 B

4.4 B

4.5 D

4.6 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

4.7 a. $T(n) = T(n-1) + c$ b. $O(n)$

4.8 D

4.9 D

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

