

Laboratorio Nro. 2

Complejidad de Algoritmos

Juan Jose Gabriel Cañon Diaz
Universidad Eafit
Medellín, Colombia
jjcanond@eafit.edu.co

Bryant Samuel Prada García
Universidad Eafit
Medellín, Colombia
bspradag@eafit.edu.co

3) Simulacro de preguntas de sustentación de Proyectos

3.1

mergeSort		insertionSort	
No. De datos	tiempo	No. De datos	tiempo
100000	117	100000	1436
110000	34	110000	1129
120000	32	120000	1421
130000	31	130000	1669
140000	58	140000	2065
150000	43	150000	2199
160000	48	160000	2504
170000	74	170000	2887
180000	48	180000	3246
190000	44	190000	3797
200000	45	200000	4011

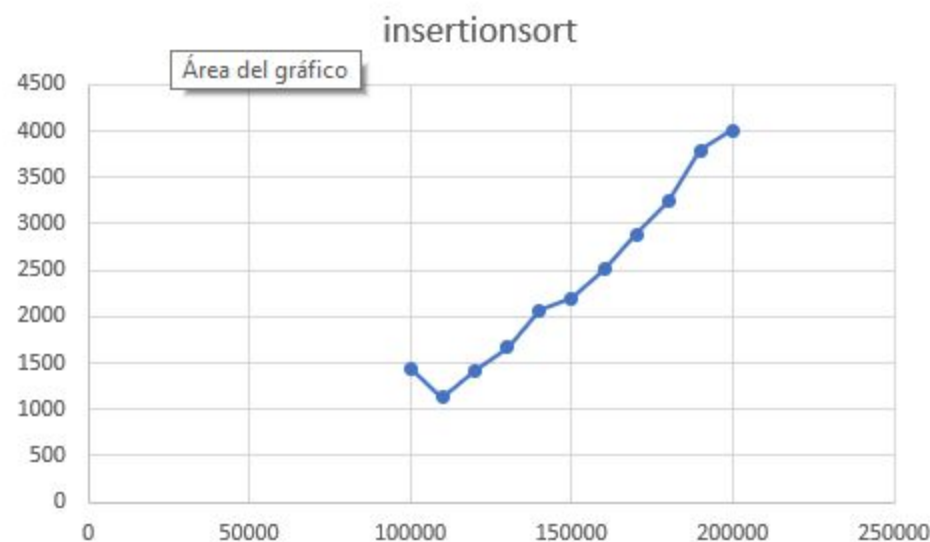
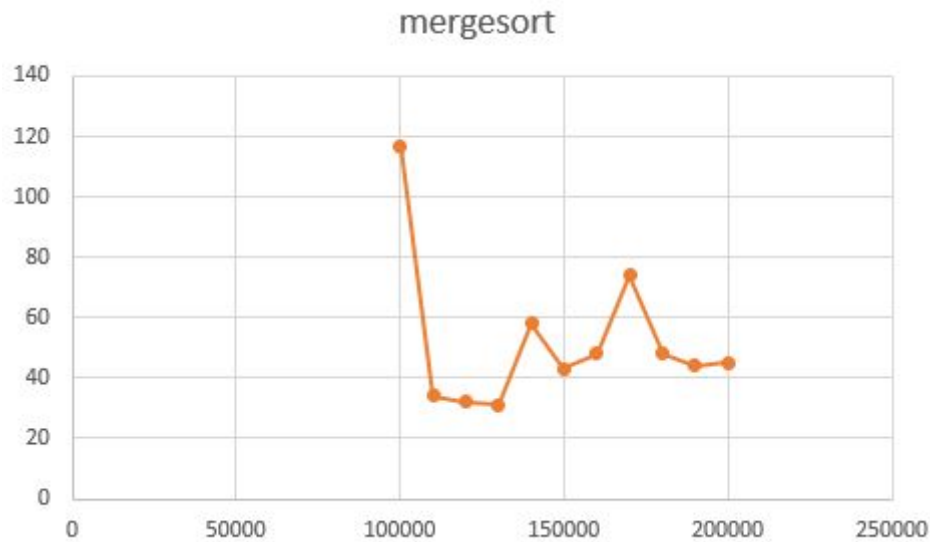
3.2

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** el merge sort es mucho más eficiente, ya que mientras merge sort gastó entre 20 y 120 milisegundos, el insertion sort gastó entre 1000 y 4000 milisegundos
- 3.4** no, ya que el solo es más efectivo para tamaños pequeños, mientras que el merge sort es más eficiente para tamaños grandes
- 3.5** para que insertion sort sea más efectivo que merge sort el arreglo debe estar en su mayoría ordenados
- 3.6**

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

3.7

- Complejidad #1

```

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

```

- Complejidad #2

```

public int[] fizzArray(int n) {
    int [] arr = new int [n]; //O(1)
    for(int i=0; i<n;i++){ //O(n)
        arr[i]=i; //O(1)
    }
    return arr; //O(1)
}

```

- Complejidad #3

```

public boolean lucky13(int[] nums) {
    for(int i=0;i<nums.length;i++){ //O(n)
        if(nums[i]==1||nums[i]==3){ //O(1)
            return false; //O(1)
        }
    }
    return true;
}

```

- Complejidad #4

```

public int[] shiftLeft(int[] nums) {
    int temp = 0; //O(1)
    for(int i=0; i<nums.length-1; i++){ //O(n)
        temp = nums[i]; //O(1)

        nums[i]= nums[i+1]; //O(1)
        nums[i+1] = temp; //O(1)
    }
    return nums; //O(1)
}

```

- Complejidad #5

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 sum28(int[] nums) {
    int cont=0;//O(1)
    for(int i=0;i<nums.length;i++){//O(n)
        if (nums[i]==2){//O(1)
            cont=cont+nums[i];//O(1)
        }
    }
    if(cont==8){//O(1)
        return true;//O(1)
    }
    return false;//O(1)
}
```

3.8

- **Complejidad #1**
cont -> Acumula el número de entradas pares (resultado final)
i -> Variable que ayuda a controlar el inicio, final y paso del ciclo for.
nums -> Arreglo recibido como parámetro que contiene la serie de datos que se deben evaluar.
- **Complejidad #2**
arr -> Arreglo en el que se almacenarán los resultados
i -> Controlador del ciclo for (inicio, fin y paso)
n -> Variable que recibe el método como parámetro y que a su vez indica el número de resultados que debe tener el arreglo final (arr).
- **Complejidad #3**
nums -> Arreglo recibido como parámetro con los valores a examinar.
i -> iterador y variable de parada del método for.
- **Complejidad #4**
nums -> Arreglo recibido por parámetro con los valores que se deben modificar
temp -> Variable auxiliar que ayudará a almacenar temporalmente el valor en que se está recorriendo actualmente el arreglo.
i -> Variable del ciclo for, que controla el inicio, final y paso de este mismo.
- **Complejidad #5**
nums -> Arreglo recibido como parámetro del método que contiene los números que se deben evaluar.
cont -> Variable que ayudará a almacenar la suma de todos los "2" del arreglo para hacer la comparación final.
i -> Variable del ciclo for (inicio, fin e incremento).

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) Simulacro de Parcial

4.1 d
4.2 b
4.3 d
4.4c
4.5
 4.5.1 b
 4.5.2 a
4.6
4.7 1
4.8a
4.9 a
4.10 d
4.11c
4.12b
4.13
4.14 b

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

