

# **Análisis de Algoritmos y Estructura de Datos**

## **Laboratorio Búsqueda y ordenamiento**

Prof. Violeta Chang C

**Semestre 2 – 2023**



# Búsqueda y ordenamiento

- **Contenidos:**

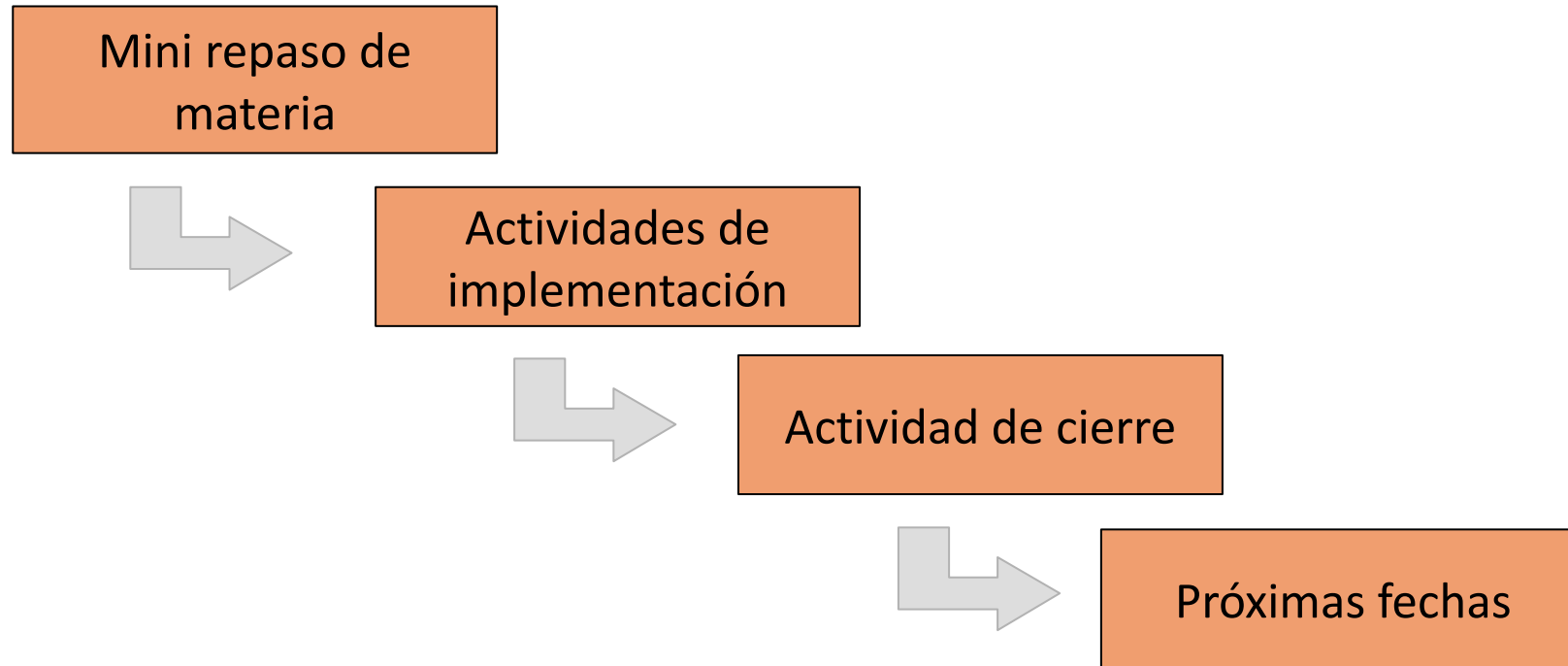
- Algoritmos de búsqueda en C
- Algoritmos de ordenamiento en C

- **Objetivos:**

- Implementar algoritmos de búsqueda en C
- Implementar algoritmos de ordenamiento en C



# Ruta de trabajo





# Algoritmos de búsqueda

Conjunto de  
datos del mismo  
tipo

Valor exacto  
que se busca



V/F si lo encuentra  
o posición en  
conjunto de datos  
de valor encontrado



# Búsqueda lineal/secuencial

```
busquedaSecuencial(arreglo, datoBuscado) : num  
    n ← tamaño(arreglo)  
    Para i ← 1 hasta n paso 1  
        Si arreglo(i) = datoBuscado entonces  
            devolver(i)  
    devolver(0)
```

$O(1)$   
 $O(n)$   
 $O(1)$  }  $O(n)$



# Búsqueda binaria

```
busquedaBinaria(arreglo, inicio, final, datoBuscado) : num
    centro ← piso((inicio + final) / 2)
    Si arreglo[centro] = datoBuscado entonces }
        devolver(centro)
    sino
        { Si arreglo[centro] > datoBuscado entonces
            devolver(busquedaBinaria(arreglo, inicio, centro - 1, datoBuscado))
        sino
            devolver(busquedaBinaria(arreglo, centro + 1, final, datoBuscado))
```

$$T(n) = T(n/2) + O(1)$$

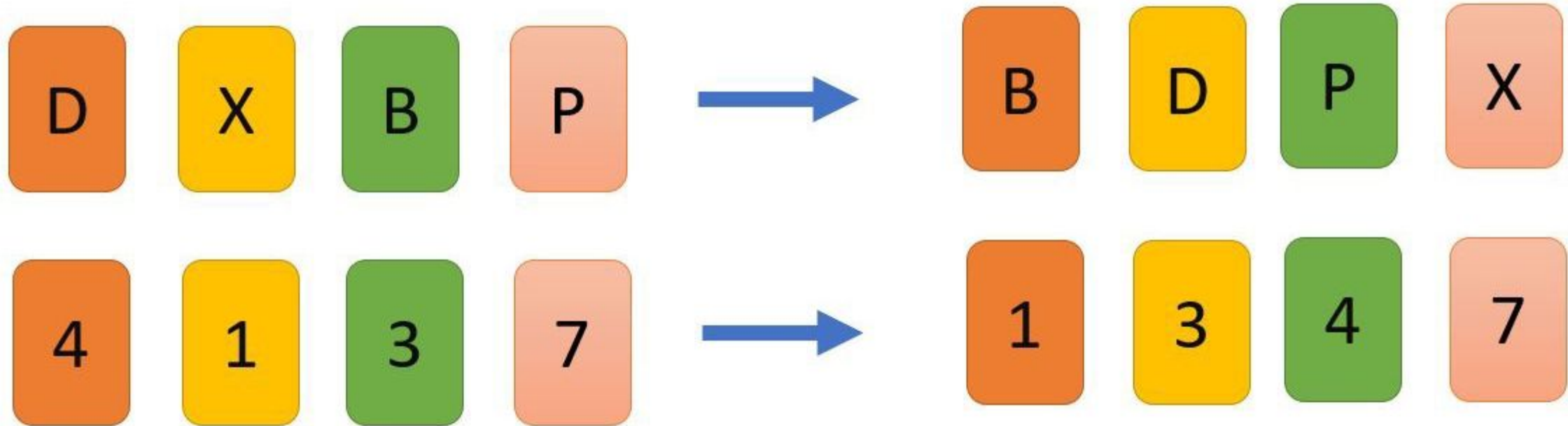
$$\bullet T(n) = aT\left(\frac{n}{b}\right) + O(n^k), \text{ si } n \geq b$$

$$O(\quad) = \begin{cases} n^k & \text{si } a < b^k \\ n^k \log_b n & \text{si } a = b^k \\ n^{\log_b a} & \text{si } a > b^k \end{cases}$$

$$\Rightarrow O(\log_2 n)$$



# Algoritmos de ordenamiento







# Ordenamiento por selección

```
ordenamientoSeleccion(arregloA) : arreglo
  n ← tamaño(arregloA)
  Para i ← 1 hasta n-1 paso -1
    indiceMejor ← i
    valorMejor ← arregloA(i)
    Para j ← i+1 hasta n paso 1
      Si arregloA(j) < valorMejor entonces
        indiceMejor ← j
        valorMejor ← arregloA(j)
    Si indiceMejor <> i entonces
      arregloA ← intercambiar(arregloA, i, indiceMejor)
  devolver (arregloA)
```

$O(1)$

$O(n^2)$

$O(1)$

$\Rightarrow O(n^2)$





# Ordenamiento por inserción

```
ordenamientoInsercion(arregloA) : arreglo
  n ← tamaño(arregloA)
  Para i ← 2 hasta n paso 1
    j ← i
    Mientras j ≥ 2 y arregloA(j) < arregloA(j-1) hacer
      arregloA ← intercambiar(arregloA, j, j-1)
      j ← j-1
  devolver (arregloA)
```

$O(1)$

$O(n^2)$

$O(1)$

$\Rightarrow O(n^2)$



# Ordenamiento por burbuja

```
ordenamientoBurbuja(arregloA): arreglo
```

```
  n ← tamaño(arregloA)
```

```
  Para i ← n hasta 1 paso -1
```

```
    Para j ← 1 hasta i-1 paso 1
```

```
      Si arregloA(j) > arregloA(j+1) entonces
```

```
        arregloA ← intercambiar(arregloA, j, j+1)
```

```
  devolver (arregloA)
```

} O(1)

} O(n<sup>2</sup>)

} O(1)

⇒ O(n<sup>2</sup>)



# Ordenamiento rápido (Quicksort)

```
quickSort(arregloA, inicio, fin)
  Si inicio < fin entonces
    pivote ← particiona(arregloA, inicio, fin)
    { quickSort(arregloA, inicio, pivote-1)
      quickSort(arregloA, pivote+1, fin)
```

$$T(n) = aT\left(\frac{n}{b}\right) + O(n^k), \text{ si } n \geq b$$
$$O() = \begin{cases} n^k & \text{si } a < \text{otro caso} \\ n^k \log_b n & \text{si } a = b^k \\ n^{\log_b a} & \text{si } a > b^k \end{cases}$$

$$T(n) = 2T(n/2) + O(n)$$

$\Rightarrow O(n \log_2 n)$  ...en caso **promedio**



# Actividad de implementación 1

1. Implementar la función en C llamada *leerArchivo*, que lea un archivo cuyo nombre se ingresa por consola al momento de la ejecución y devuelva un arreglo de  $n$  enteros
  - El formato del archivo consiste en un número entero por línea: el primer número corresponde al número  $n$  de elementos, mientras que desde la fila 2 hasta la fila  $n+1$  se encuentran los  $n$  elementos
  - Los  $n$  elementos pueden contener números repetidos
2. Realizar llamado de la función creada desde función *main()* en `lab03-busquedaOrdenamiento.c` (se debe crear `.c`)
3. Imprimir por consola los elementos del arreglo generado



# Actividad de implementación 2

1. Implementar la siguiente función en C

```
int* ordenamientoBurbuja(int* arreglo, int n)
```

que realiza el ordenamiento de los n elementos del arreglo usando el método por burbuja. La función debe devolver el arreglo ordenado

2. Realizar llamado de la función creada desde función *main()* en lab03-busquedaOrdenamiento.c usando el arreglo generado en actividad 1 como entrada a la función

3. Imprimir por consola los elementos del arreglo ordenado



# Actividad de implementación 3

1. Implementar la siguiente función en C

```
int busquedaBinaria(int* arreglo, int n, int dato)
```

que busca el dato en el arreglo de n elementos usando el método de búsqueda binaria. La función debe devolver 1 si encuentra el dato, en caso contrario devuelve 0

2. Realizar llamados de la función creada desde función *main()* en lab03-busquedaOrdenamiento.c usando el arreglo generado en actividad 2 como entrada a la función, además de un dato existente y otro inexistente

3. Imprimir por consola la respuesta de cada búsqueda



# Actividad de implementación 4

1. Implementar la siguiente función en C

```
int* ordenamientoInsercion(int* arreglo, int n)
```

que realiza el ordenamiento de los n elementos del arreglo usando el método por inserción. La función debe devolver el arreglo ordenado

2. Realizar llamado de la función creada desde función *main()* en lab03-busquedaOrdenamiento.c usando el arreglo generado en actividad 1 como entrada a la función

3. Imprimir por consola los elementos del arreglo ordenado





# Actividad de implementación 5

1. Implementar la siguiente función en C

```
int busquedaSecuencial(int* arreglo, int n, int dato)
```

que busca el dato en el arreglo de n elementos usando el método de búsqueda secuencial. La función debe devolver 1 si encuentra el dato, en caso contrario devuelve 0

2. Realizar llamados de la función creada desde función *main()* en lab03-busquedaOrdenamiento.c usando el arreglo generado en actividad 1 como entrada a la función, además de un dato existente y otro inexistente
3. Imprimir por consola la respuesta de cada búsqueda



# Entrega de actividad de laboratorio

- Entrega obligatoria
- Subir SOLO actividades 1, 2 y 3 de esta sesión en buzón de uVirtual, en único archivo **s3\_apellido\_nombre.zip**
- Plazo: **hoy** dentro del horario de laboratorio

# Actividad de cierre



- Ir a [menti.com](https://menti.com) e ingresar código 7810 8026



<div>  <b>Octubre 2023</b> </div>						
Domingo	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado
1	2	3 	4 	5	6 	7
8	9 	10	11	12 	13 	14
15	16 <small>Encuentro de Dos Mundos</small>	17	18	19 	20 	21
22	23	24	25 	26 	27 <small>Día Nacional de las Iglesias Evangélicas y Protestantes</small>	28