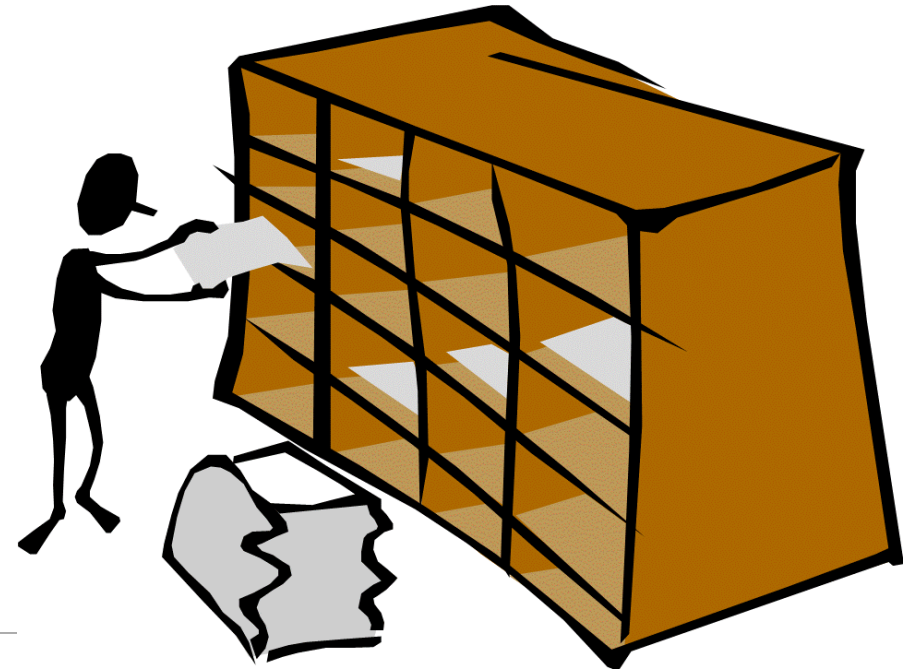


# Ordenamiento y Búsqueda

---



M.C YOLANDA MOYAO MARTÍNEZ

# Aplicaciones

---

## Búsqueda

Proceso mediante el cual podemos localizar un elemento con un valor específico dentro de un conjunto de datos.



# Búsqueda lineal

---

Consiste en empezar al inicio del conjunto de elementos , e ir a través de ellos hasta encontrar el elemento indicado ó hasta llegar al final de arreglo.



# VALOR a Buscar = 21

**A[1]=1 <> VALOR**

A = 1 11 21 25 26 33 38 40 42 48

i=1    ↑

**Segunda iteración: A[2]=11 <> VALOR**

A = 1 **11** 21 25 26 33 38 40 42 48

i=2    ↑

**Tercera iteración: A[3]=21 = VALOR**

A = 1 11 **21** 25 26 33 38 40 42 48

i=3    ↑

# Algoritmo

Nombre: bus\_sec

N 20

Entrada: A[N] Salida: cadenas

---

Inicio

Para ( $i \leftarrow 1$  hasta N Paso 1)

Si ( $a[i] = b$ ) entonces

band  $\leftarrow 1$

Fin\_Si

Fin\_Para

Si band = 1

Escribir("Encontrado")

Si\_No

Escribir ("Valor no encontrado")

Fin\_Si

Fin\_bus\_sec

# Búsqueda binaria

---

El método requiere que la información sobre la cual se va a buscar este ordenada.

Al estar ésta ordenada puede descartarse la mitad que se sabe no es posible que este la información



# Algoritmo

Nombre:bus\_bin

N 20

---

Entrada: A[N] Salida:

Inicio

$m \leftarrow N \text{ DIV } 2$

$i \leftarrow 1$     $j \leftarrow N$

Coment: Recorrido del arreglo buscando VALOR

mientras ( ( A[ m ] <> VALOR ) Y ( i <= j ) )

    Si A[m] > VALOR

$j \leftarrow m - 1$

    Si\_no

$i \leftarrow m + 1$

    Fin\_si

$m \leftarrow ( i + j ) \text{ DIV } 2$

Fin\_mientras

Coment: Determinar si encontró o no

Si (A[m] = VALOR)

    POS  $\leftarrow$  m

Fin\_bus\_bin

# Tipos de Ordenamiento

---

## Ordenamiento interno.

- Se lleva a cabo completamente en memoria principal.

## Ordenamiento externo.

- No cabe toda la información en memoria principal y es necesario ocupar memoria secuencial





# Criterios de Eficiencia

---

1. El número de pasos.
2. El número de comparaciones necesarios para ordenar  $n$  elementos.
3. El número de movimientos de elementos que se requieren para ordenar  $n$  elementos.



# Métodos Simples

Método de Burbujeo

Método de Inserción

Método de Selección





# Selección

25	3	12	19	2	1	9	6
1	3	12	19	2	25	9	6
	2	12	19	3	25	9	6
		3	19	12	25	9	6
			6	12	25	9	19
				9	25	12	19
					12	25	19
						19	25

Nombre: selección

N 8

Entrada: A[N] Salida:

Inicio

Para (i ← 1 Hasta N Paso 1)

imin ← i

Para (j ← i+1 Hasta N Paso 1)

si (a[j] < a [imin])

imin ← j

fin\_si

Fin\_Para

temp ← a[i]

a[i] ← a[imin]

a[imin] ← temp

Fin\_Para

Fin\_seleccion

# Inserción

25	3	12	19	2	1	9	6
3	25	12					
3	12	25	19				
3	12	19	25	2			
2	3	12	19	25	1		
1	2	3	12	19	25	9	
1	2	3	9	12	19	25	6
1	2	3	6	9	12	19	25

Nombre: insercion

N 8

Entrada: a[N] Salida:

Inicio

Para ( $i \leftarrow 2$  Hasta N Paso 1)

$j \leftarrow i$

Mientras ( $j > 1$ ) AND ( $a[j] < a[j-1]$ )

temp  $\leftarrow$  a[i]

a[i]  $\leftarrow$  a[j-1]

a[j-1]  $\leftarrow$  temp

$j \leftarrow j-1$

Fin\_Mientras

Fin\_Para

Fin\_insercion

# Tarea

---

1. Hacer el pseudocódigo para guardar números hasta dar el 0 en un arreglo de tamaño 50 y posteriormente contar todos los datos positivos y negativos.
2. Hacer el pseudocódigo para almacenar 5 números enteros y posteriormente calcular el factorial para cada uno de estos.
3. Hacer el pseudocódigo para almacenar 10 números enteros y posteriormente contar cuantos son primos y cuantos no lo son.
4. Investigar el método de shell