

# Algoritmos de Ordenamiento Interno

---

UNIDAD 7. ALGORITMOS Y ESTRUCTURAS DE DATOS



# Definición

---

- En muchos casos se desea que la información que se maneja en memoria se encuentre ordenada
- Para lograr esto, se cuentan con diferentes métodos de ordenamiento interno

# Ordenamiento por Burbuja

---

- Es el más sencillo de los algoritmos de ordenamiento
- Revisa cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si están en el orden equivocado
- Es necesario revisar varias veces toda la lista hasta que no se necesiten más intercambios
- Este algoritmo obtiene su nombre de la forma con la que suben por la lista los elementos durante los intercambios, como si fueran "burbujas"

# Ejemplo

---

4,3,5,2,1	3,4,5,2,1	3,2,1,4,5	2,3,1,4,5
3,4,5,2,1	3,4,5,2,1	2,3,1,4,5	2,1,3,4,5
3,4,5,2,1	3,4,2,5,1	2,1,3,4,5	2,1,3,4,5
3,4,2,5,1	3,4,2,1,5	2,1,3,4,5	2,1,3,4,5
3,4,2,1,5	3,4,2,1,5	2,1,3,4,5	1,2,3,4,5
3,4,2,1,5	3,2,4,1,5	1,2,3,4,5	1,2,3,4,5
3,2,4,1,5	3,2,1,4,5	1,2,3,4,5	1,2,3,4,5
3,2,1,4,5	3,2,1,4,5	1,2,3,4,5	1,2,3,4,5

# Ordenamiento por Inserción Directa

---

- No se recorre todo el arreglo, solamente aquellos elementos que están antes del elemento que se está evaluando
- Es un algoritmo que no necesita tener todos los elementos a ordenar, si se insertan nuevos elementos, los ordena conforme se integran
- Se basa en generar una lista ordenada a partir de los elementos que componen un arreglo o una lista.
- Se deben realizar varias pasadas sobre el arreglo, en cada paso se analiza un elemento y se obtiene su orden relativo a los elementos ya revisados

# Ejemplo

---

Suponga que se tiene el siguiente arreglo:

45, 52, 21, 37, 49, 72, 14

1ra iteración.

45, 52, 21, 37, 49, 72, 14

# Ejemplo

---

2da iteración

45, 52, 21, 37, 49, 72, 14

45, 21, 52, 37, 49, 72, 14

21, 45, 52, 37, 49, 72, 14

3ra iteración

21, 45, 52, 37, 49, 72, 14

21, 45, 37, 52, 49, 72, 14

21, 37, 45, 52, 49, 72, 14

# Ejemplo

---

4ta iteración

21, 37, 45, 52, 49, 72, 14

21, 37, 45, 49, 52, 72, 14

5ta iteración

21, 37, 45, 49, 52, 72, 14

# Ejemplo

---

6ta iteración

21, 37, 45, 49, 52, 72, 14

21, 37, 45, 49, 52, 14, 72

21, 37, 45, 49, 14, 52, 72

21, 37, 45, 14, 49, 52, 72

21, 37, 14, 45, 49, 52, 72

21, 14, 37, 45, 49, 52, 72

14, 21, 37, 45, 49, 52, 72

# Análisis

---

- El algoritmo de ordenamiento por inserción tiene el siguiente orden:
  - En el mejor de los casos, cuando todo el arreglo está ordenado:  $n$
  - En el peor de los casos, que sería cuando el arreglo está desordenado (ordenado de manera inversa):  $n^2$

# Mergesort

---

- Es una mejora al algoritmo por Inserción Directa
- Se basa en saltos mayores de una posición (intercambio) entre valores
- Al finalizar el intercambio por saltos mayores a uno, se realiza un ordenamiento por Inserción Directa que necesita menos movimientos
- Se inicia considerando el número de elementos, a partir de esto, el tamaño del salto para la comparación viene dado por
- $K = n/2$
- En donde K es la distancia de comparación y n el número de elementos
- Se comparan los elementos a K elementos de distancia y si el primero es menor, se intercambian
- Se sigue dividiendo tomando  $n=K$  hasta que K sea 1, en ese momento se ordena con Inserción Directa

# Ejemplo

---

18 46 25 8 2 36 16 50 30 2

$n = 10$

$K = 10/2 = 5$

18 46 25 8 2 36 16 50 30 2

18 46 25 8 2 36 16 50 30 2

18 16 25 8 2 36 46 50 30 2

18 16 25 8 2 36 46 50 30 2

18 16 25 8 2 36 46 50 30 2

18 16 25 8 2 36 46 50 30 2

18 16 25 8 2 36 46 50 30 2

18 16 25 8 2 36 46 50 30 2

18 16 25 8 2 36 46 50 30 2

# Ejemplo

---

18 16 25 8 2 36 46 50 30 2

$K = 5/2 = 2.5 = 2$

18 16 25 8 2 36 46 50 30 2

18 16 25 8 2 36 46 50 30 2

18 8 25 16 2 36 46 50 30 2

18 8 2 16 25 36 46 50 30 2

18 8 2 16 25 36 46 50 30 2

18 8 2 16 25 36 46 50 30 2

18 8 2 16 25 36 46 50 30 2

18 16 25 8 2 36 46 50 30 2

18 8 25 16 2 36 46 50 30 2

18 8 2 16 25 36 46 50 30 2

18 8 2 16 25 36 46 50 30 2

18 8 2 16 25 36 46 50 30 2

18 8 2 16 25 36 46 50 30 2

18 8 2 16 25 36 30 50 46 2

# Ejemplo

---

18 8 2 16 25 36 30 50 46 2

$K = K/2 = 1$  : Inserción Directa

18 8 2 16 25 36 30 2 46 50

8 18 2 16 25 36 30 2 46 50

8 2 18 16 25 36 30 2 46 50

2 8 18 16 25 36 30 2 46 50

2 8 16 18 25 36 30 2 46 50

2 8 16 18 25 36 30 2 46 50

2 8 16 18 25 36 30 2 46 50

18 8 2 16 25 36 30 2 46 50

8 18 2 16 25 36 30 2 46 50

8 2 18 16 25 36 30 2 46 50

2 8 18 16 25 36 30 2 46 50

2 8 16 18 25 36 30 2 46 50

2 8 16 18 25 36 30 2 46 50

2 8 16 18 25 36 30 2 46 50

2 8 16 18 25 30 36 2 46 50

# Ejemplo

---

2 8 16 18 25 30 36 2 46 50

2 8 16 18 25 30 2 36 46 50

2 8 16 18 25 2 30 36 46 50

2 8 16 18 2 25 30 36 46 50

2 8 16 2 18 25 30 36 46 50

2 8 2 16 18 25 30 36 46 50

2 2 8 16 18 25 30 36 46 50

2 2 8 16 18 25 30 36 46 50

2 2 8 16 18 25 30 36 46 50

2 8 16 18 25 30 2 36 46 50

2 8 16 18 25 2 30 36 46 50

2 8 16 18 2 25 30 36 46 50

2 8 16 2 18 25 30 36 46 50

2 8 2 16 18 25 30 36 46 50

2 2 8 16 18 25 30 36 46 50

2 2 8 16 18 25 30 36 46 50

2 2 8 16 18 25 30 36 46 50

2 2 8 16 18 25 30 36 46 50

# Ordenamiento Rápido

---

# Funcionamiento

---

- El algoritmo básico de *Quicksort* permite tomar cualquier elemento como pivote
- De la elección del pivote depende la eficiencia del algoritmo
- Algunos métodos para elegir el elemento que se utilizará como pivote son:
  - Elección aleatoria del pivote
    - No se requiere ningún cálculo adicional
    - El algoritmo puede tener un gran número de iteraciones
- Recorrer la lista para ver que pivote es el conveniente
- Tomar el elemento de que está en medio del arreglo

# Ejemplo

---

13,81,92,43,65,31,57,26,75,0

13,43,31,26,0, 65, 81,92,75

13,26,0, 31, 43, 65 ,81,92,75

13,0, 26, 31, 43, 65, 81,92,75

0,13,26,31,43,65, 81,92,75

0,13,26,31,43,65, 81, 75, 92

0,13,26,31,43,65, 81,75,92

# Ordenamiento por Mezcla

---

- En cada paso el menor de los valores A y B se copia en la próxima posición de C, actualizándose los contadores apropiados
- Cuando uno de los vectores de entrada se acaba, el resto del otro vector se copia en C

# Ordenamiento por Mezcla

---

- El algoritmo de mezcla parte de dos vectores de entrada A y B produciendo un vector de salida C
- Se apoya en tres contadores:
  - Acont
  - Bcont
  - Ccont

# Ejemplo

---

1,13,24,26  
2,15,24,38

1,13,24,26  
2,15,24,38

1,13,24,26  
2,15,24,38

1,13,24,26  
2,15,24,38

1,13,24,26  
2,15,24,38

1,13,24,26  
2,15,24,38

1,13,24,26  
2,15,24,38

1,13,24,26  
2,15,24,38

1,13,24,26  
2,15,24,38

1

1,2

1,2,13

1,2,13,15

1,2,13,15,24

1,2,13,15,24,24

1,2,13,15,24,24,26

1,2,13,15,24,24,26,38