



TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

Licenciatura en Ingeniería en Tecnologías de la Información e Innovación Digital
(DSM)

ESTRUCTURA DE DATOS

UNIDAD II

R1

Grupo: GTID0141

Alumno: Avalos Melado Rey Gerardo

No.Control: 1224100509

Docente: Gabriel Barron Rodriguez

Ordenamiento de arreglos

PARTE 5 - ORDENAMIENTO DE ARREGLOS

Parte 1: Ordenamiento de arreglos

Ejercicio 1: Ordenamiento por burbuja

Dado el arreglo:

[7, 3, 5, 2]

Realiza todas las pasadas del algoritmo bubble sort

Pasada 1

1. (7, 3) \rightarrow 7 > 3 \rightarrow swap \rightarrow [3, 7, 5, 2]

2. (7, 5) \rightarrow 7 > 5 \rightarrow swap \rightarrow [3, 5, 7, 2]

3. (7, 2) \rightarrow 7 > 2 \rightarrow swap \rightarrow [3, 5, 2, 7]

En pasada 1: [3, 5, 2, 7]

Pasada 2

1. (3, 5) \rightarrow OK

2. (5, 2) \rightarrow swap \rightarrow [3, 2, 5, 7]

3. (5, 7) \rightarrow OK

En pasada 2: [3, 2, 5, 7]

3. (5, 7) \rightarrow OK

En pasada 2: [3, 2, 5, 7]

Pasada 3

1. (3, 2) \rightarrow swap \rightarrow [2, 3, 5, 7]

2. (3, 5) \rightarrow OK

3. (5, 7) \rightarrow OK

En pasada 3: [2, 3, 5, 7]

Arreglo final: [2, 3, 5, 7]

Ejercicio 2: Ordenamiento por selección

Dado el arreglo

[9, 4, 6, 1, 3]

Realiza para 4 pasos

Iteración 1

minimo = 1 (indice 3)

Swap con posición 0

\rightarrow [1, 4, 6, 9, 3]

Iteración 2

minimo = 3 (indice 4)

Swap con posición 1

\rightarrow [1, 3, 6, 9, 4]

Iteración 3

minimo = 4 (indice 4)

Swap con posición 2

\rightarrow [1, 3, 9, 6, 4]

Iteración 4
mínimo = 6 (índice 4)

Swap con posición 3
 $\rightarrow [1, 3, 2, 6, 9]$

Preguntar: Cuántos intercambios se hicieron en total? 4

Ejercicio 3 - Ordenamiento por intercambio

Arreglo: [8, 2, 5, 1, 4]

i = 1 (clave = 2)

2 < 8 \rightarrow 8

$\rightarrow [8, 8, 5, 1, 4]$

2

$\rightarrow [2, 8, 5, 1, 4]$

Comparaciones: 1

i = 2 (clave = 5)

8 < 8 \rightarrow 8

$\rightarrow [2, 8, 8, 1, 4]$

5

$\rightarrow [2, 8, 5, 1, 4]$

Comparaciones: 2

i = 3 (clave = 1)

1 < 8 \rightarrow mover

1 < 5 \rightarrow mover

1 < 2 \rightarrow mover

1

$\rightarrow [1, 2, 5, 8, 4]$

Comparaciones: 3

i = 4 (clave = 1)

4 < 8

4 < 5

4 > 2

$\rightarrow [1, 2, 4, 5, 8]$

Comparaciones: 3

Total de comparaciones: $1+2+3+3=9$

Ejercicio 4: MergeSort

[10, 7, 3, 8, 2, 6]

División (Cál, boll)

[10, 7, 3, 8, 2, 6]

$\rightarrow [10, 7, 3] + [8, 2, 6]$

$\rightarrow [10, 7] + [3] \text{ y } [8, 2] + [6]$

$\rightarrow [10] + [7] \text{ y } [8] + [2]$

Combinaciones:

• [10] + [7] $\rightarrow [7, 10]$

• [7, 10] + [3] $\rightarrow [3, 7, 10]$

• [8] + [2] $\rightarrow [2, 8]$

• [2, 8] + [6] $\rightarrow [2, 6, 8]$

• [3, 7, 10] + [3, 6, 8] \rightarrow

Resultado: [2, 3, 6, 7, 8, 10]

PARTE 2: Búsqueda en arreglos

Ejercicio 5: Búsqueda lineal

[4, 9, 2, 7, 5, 8] Buscar el número 7

1. Comp 1: $4 \neq 7$
2. Comp 2: $9 \neq 7$
3. Comp 3: $2 \neq 7$
4. Comp 4: $7 = 7$

Se realizaron 4 comparaciones

Ejercicio 6 - Búsqueda Binaria

Arreglo [3, 6, 8, 12, 15, 18, 20]

Buscar: 15

1. Inicio = 0 fin = 6 mid = 3 $\rightarrow \text{ar}[3] = 12 < 15 \rightarrow$ Buscar derecha
2. Inicio = 4 fin = 6 mid = 5 $\rightarrow \text{ar}[5] = 18 > 15 \rightarrow$ Buscar izquierda
3. Inicio = 4 fin = 4 mid = 4 $\rightarrow \text{ar}[4] = 15$

Fue encontrado en el paso 3

Ejercicio 7 - Búsqueda Binaria (NO existe)

Arreglo [2, 5, 9, 14, 21, 30]

Buscar: 10

1. $i=0 f=5 \text{ mid}=2 \rightarrow 9 > 10 \rightarrow$ Buscar derecha
2. $i=3 f=5 \text{ mid}=4 \rightarrow 21 > 10 \rightarrow$ Buscar izquierda
3. $i=3 f=3 \text{ mid}=3 \rightarrow 14 > 10 \rightarrow$ izquierda

Ahora: inicio (4) > fin (3) \rightarrow detener

Condición para terminar

inicio > fin (introducir valor)

P

Parte 3 - Introducción

Ejercicio 8 - Bubble Sort + Búsqueda binaria

Anexo [12, 3, 19, 5, 8, 1]

Bubble Sort

[1, 3, 5, 8, 12, 19]

Búsqueda binaria del 8

1. $i = 0 \ f = 5 \ mid = 2 \rightarrow 5 > 8$ Encuentro en el paso 3

2. $i = 3 \ f = 5 \ mid = 4 \rightarrow 12 > 8$

3. $i = 3 \ f = 3 \ mid = 3 \rightarrow 8$

Ejercicio 9 - Elegir el mejor algoritmo

Dado el anexo

[5, 4, 3, 2, 1]

Que algoritmo de ordenamiento sea más eficiente en este caso? Insertion Sort

Hace menos movimientos

Es muy eficiente cuando el anexo está casi ordenado

Ejercicio 10. Simulación de QuickSort

Anexo: [9, 4, 7, 3, 10, 1]

Pivote inicial = 9

Partición

menores que 9: [4, 7, 3, 1]

mayores que 9: [10]

QuickSort izquierdo: [4, 7, 3, 1]

pivote: 4

menores: [3, 1]

mayores: [7]

QuickSort de [3, 1]

pivote: 3

menores: [1]

Mayores: []

resultados $\rightarrow [1, 3]$

Ramiendo todo

1) q. ordenado: [1, 3, 4, 7]

pivote: 4

Derecha: [10]

resultado final: [1, 3, 4, 7, 9, 10]