



Guía de laboratorio

Laboratorio de Contenido: Bubble Sort en Python

Explicación básica

Bubble Sort ordena comparando elementos adyacentes e intercambiándolos si están en orden incorrecto. En cada pasada, el valor mayor “sube” hasta su posición final. Su complejidad es $O(n^2)$.

Objetivos

- Comprender el funcionamiento de Bubble Sort.
- Implementarlo en Python.
- Analizar su número de comparaciones e intercambios.

Caso práctico

Un sistema registra tiempos de respuesta (en ms) de un servidor:

tiempos = [120, 85, 150, 95, 110]

Se requiere ordenarlos de menor a mayor para identificar los tiempos más críticos.

Problema

Implementar Bubble Sort para ordenar la lista y mostrar el estado de la lista en cada pasada.

Ejemplo paso a paso (Algoritmo básico)

Lista inicial:

120, 85, 150, 95, 110

Primera pasada:

120–85 ✗

120–150 ✓

150–95 ✗

150–110 ✗

Resultado:

85, 120, 95, 110, 150

(150 queda fijo)

Implementación en Python

```
# -----  
# Método Bubble Sort en Python  
# -----
```

```
def bubble_sort(lista):
```



Guía de laboratorio

```
n = len(lista)
comparaciones = 0
intercambios = 0

for i in range(n):
    hubo_intercambio = False
    print(f"\nPasada {i+1}:")

    for j in range(0, n - i - 1):
        comparaciones += 1
        print(f" Comparando {lista[j]} y {lista[j+1]}")

        if lista[j] > lista[j + 1]:
            # Intercambio
            lista[j], lista[j + 1] = lista[j + 1], lista[j]
            intercambios += 1
            hubo_intercambio = True
            print(f" → Se intercambia: {lista}")

    # Optimización: si no hubo intercambio, la lista ya está ordenada
    if not hubo_intercambio:
        break

print("\nLista ordenada:", lista)
print("Total comparaciones:", comparaciones)
print("Total intercambios:", intercambios)

return lista

# -----
# Programa principal
# -----

if __name__ == "__main__":
    datos = [5, 8, 3, 7, 6]
    print("Lista original:", datos)
    bubble_sort(datos.copy())
```

Reflexión o Análisis

- ¿Cuántas comparaciones realiza?



Guía de laboratorio

- ¿Qué sucede si la lista ya está ordenada?
- ¿Por qué no es eficiente para grandes volúmenes de datos?