Autor
Prof. Sabrina Lizbeth Vega Maldonado

Burbuja

Este algoritmo compara 2 elementos consecutivos del arreglo, coloca el mayor en la posición derecha y el menor en la posición izquierda.

Repite este procedimiento por todo el arreglo para ubicar un dato en su posición.
Por lo tanto se repite *n cuadrada* para ubicar los *n elementos*.

Autor
Prof. Sabrina Lizbeth Vega Maldonado

Burbuja

15	12	65	45	1	2

Primera pasada

15-12-65-45-1-2
12-15-65-45-1-2
12-15-65-45-1-2
12-15-45-65-1-2
12-15-45-1-65-2
12-15-45-1-2-65

Segunda pasada

12-15-45-1-2
12-15-45-1-2
12-15-45-1-2
12-15-1-45-2
12-15-1-2-45

Autor
Prof. Sabrina Lizbeth Vega Maldonado

Burbuja

15	12	65	45	1	2

Tercera pasada

12-15-1-2
12-15-1-2
12-1-15-2
12-1-2-15

Cuarta pasada

12-1-2
1-12-2
1-2-12

Autor
Prof. Sabrina Lizbeth Vega Maldonado

Complejidad

- **Ventajas:**
  - Fácil implementación.
  - No requiere memoria adicional.
- **Desventajas:**
  - Muy lento.
  - Realiza numerosas comparaciones.
  - Realiza numerosos intercambios.
- **Total comparaciones-intercambios**

  n(n-1)/2
- **Complejidad:**

  O(n2)

ALGORITMO

```
void burbuja (int array[ ], int tam) {
    int i, j, temp;
    for (i = tam-1; i > 0; i--) {
        for (j = 0; j < i; j++) {
            if (array[j] > array[j+1]) {
                temp = array[j];
                array[j] = array[j+1];
                array[j+1] = temp;
            }
        }
}
```

Autor
Prof. Sabrina Lizbeth Vega Maldonado

Burbuja Mejorada

ALGORITMO

```
void burbujaMejorada (int array[ ], int tam) {
      int i = tam-1 , j , temp, bandera = 0;
      while  ( i > 0   &&  ! bandera  ) {
            bandera = 1 ;
             for (j = 0; j < i; j++) {
                  if (array[j] > array[j+1]) {
                        temp = array[j];
                        array[j] = array[j+1];
                        array[j+1] = temp;
                        bandera = 0 ;
                   }
             }
            i-- ;
}
```