

Métodos de Ordenamiento

Métodos de Ordenamiento

Métodos Iterativos:

- Burbuja
 - Simple
 - Mejorada
- Inserción
- Selección

Métodos Iterativos

Burbuja Simple:

Este método es uno de los más simples, es tan fácil como comparar todos los elementos de una lista (arreglo) contra todos, si se cumple que uno es mayor o menor a otro, entonces los intercambia de posición.

Métodos de Burbuja Simple

Ejemplo:

| | |
|---|---|
| 0 | 5 |
| 1 | 6 |
| 2 | 1 |
| 3 | 0 |
| 4 | 3 |

ESTRATEGIA

Comparamos todos contra todos, aunque estemos generando comparaciones extras.

Métodos de Burbuja Simple

Ejemplo:

Paso 1

para $i = 0$

| | |
|---|---|
| 0 | 5 |
| 1 | 6 |
| 2 | 1 |
| 3 | 0 |
| 4 | 3 |

$j=0$

| | |
|---|---|
| 0 | 5 |
| 1 | 6 |
| 2 | 1 |
| 3 | 0 |
| 4 | 3 |

$j=1$

| | |
|---|---|
| 0 | 5 |
| 1 | 1 |
| 2 | 6 |
| 3 | 0 |
| 4 | 3 |

$j=2$

| | |
|---|---|
| 0 | 5 |
| 1 | 1 |
| 2 | 0 |
| 3 | 6 |
| 4 | 3 |

$j=3$

| | |
|---|---|
| 0 | 5 |
| 1 | 1 |
| 2 | 0 |
| 3 | 3 |
| 4 | 6 |

Métodos de Burbuja Simple

Ejemplo:

Paso 2

para $i = 1$

| | |
|---|---|
| 0 | 5 |
| 1 | 1 |
| 2 | 0 |
| 3 | 3 |
| 4 | 6 |

$j=0$

| | |
|---|---|
| 0 | 1 |
| 1 | 5 |
| 2 | 0 |
| 3 | 3 |
| 4 | 6 |

$j=1$

| | |
|---|---|
| 0 | 1 |
| 1 | 0 |
| 2 | 5 |
| 3 | 3 |
| 4 | 6 |

$j=2$

| | |
|---|---|
| 0 | 1 |
| 1 | 0 |
| 2 | 3 |
| 3 | 5 |
| 4 | 6 |

$j=3$

| | |
|---|---|
| 0 | 1 |
| 1 | 0 |
| 2 | 3 |
| 3 | 5 |
| 4 | 6 |

Métodos de Burbuja Simple

Ejemplo:

Paso 3

para $i = 2$

| | |
|---|---|
| 0 | 1 |
| 1 | 0 |
| 2 | 3 |
| 3 | 5 |
| 4 | 6 |

$j=0$

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 5 |
| 4 | 6 |

$j=1$

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 5 |
| 4 | 6 |

$j=2$

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 5 |
| 4 | 6 |

$j=3$

| | |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 3 |
| 3 | 5 |
| 4 | 6 |

Métodos de Burbuja Simple

```
public static void simpleBubbleSort(int[] A, int n) {  
    for(int i = 0; i < n; i++) {  
        for(int j = 0; j < n - 1; j++) {  
            if(A[j] > A[j + 1]) {  
                int aux = A[j];  
                A[j] = A[j + 1];  
                A[j + 1] = aux;  
            }  
        }  
    }  
}
```


Método de Selección

Este algoritmo consiste en lo siguiente:

- Buscar el elemento más pequeño de la lista.
- Intercambiarlo con el elemento ubicado en la primera posición de la lista.
- Buscar el segundo elemento más pequeño de la lista.
- Intercambiarlo con el elemento que ocupa la segunda posición en la lista.
- Repetir este proceso hasta que toda la lista se encuentre ordenada.

Método de Selección

Por ejemplo:

Lista a ordenar: 4 - 3 - 5 - 2 - 1

intercambiamos 1 con el 4 y la lista queda así:

1 - 3 - 5 - 2 - 4

Lo intercambiamos 2 con el elemento en la segunda posición, es decir el 3.

La lista queda así:

1 - 2 - 5 - 3 - 4

intercambiamos 3 con el 5:

1 - 2 - 3 - 5 - 4

intercambiamos 4 con el 5:

1 - 2 - 3 - 4 - 5

Método de Selección

```
public static void selectionSort(int[] A, int n) {  
    for(int i = 0; i < n; i++) {  
        int p = i;  
        for(int j = i + 1; j < n; j++) {  
            if(A[j] < A[p]) {  
                p = j;  
            }  
        }  
        if(p != i) {  
            int temp = A[p];  
            A[p] = A[i];  
            A[i] = temp;  
        }  
    }  
}
```

Método de Inserción

Consideremos que en un momento dado, el segmento izquierdo del arreglo está ordenado y el segmento derecho no.

Podemos tomar un elemento del segmento derecho e *insertarlo* en su lugar correcto en el lado izquierdo.



Parte ordenada Parte desordenada

Métodos de Inserción

- Se considera la lista partida en dos sublistas, una ordenada y la otra para ordenar
- En el primer paso, la lista ordenada contiene un elemento y la lista por ordenar, $n-1$,
- Se van extrayendo uno a uno los elementos de la lista desordenada y se colocan en la sublista ordenada, mediante comparaciones sucesivas, hasta encontrar la posición correcta.
- El proceso continúa hasta que la lista desordenada queda vacía.

Métodos de Inserción

Ejemplo:

8 12 4 9 3

8 12 4 9 3

4 8 12 9 3

4 8 9 12 3

3 4 8 9 12

Métodos Iterativos

```
public static void insertionSort(int[] A, int n){  
    for(int i = 0; i < n; i++){  
        int current = A[i];  
        int position = i;  
        while(position > 0 && A[position - 1] > current){  
            A[position] = A[position - 1];  
            position--;  
        }  
        A[position] = current;  
    }  
}
```