



Selección



La idea básica de este algoritmo consiste en buscar el menor elemento del arreglo y colocarlo en la primera posición.

Luego se busca el segundo elemento mas pequeño del arreglo y lo colocamos en la segunda posición. El proceso continua hasta que todos los elementos del arreglo hayan sido ordenados.

La ventaja de éste algoritmo es que a pesar de hacer muchas comparaciones, solo hace un intercambio después de recorrer todo el arreglo.

Es decir solo en total hace máximo **n intercambios**

Selección

45 15 12 65 1 2

Primera pasada

45-15-12-65- 1-2

45-15-12-65- 1-2

1-15-12-65-45-2

Segunda pasada

15-12-65-45-2

15-12-65-45-2

2-12-65-45-15

Selección

45 15 12 65 1 2

Tercer pasada

12-65-45-15

Cuarta pasada

65-45-15

65-45-15

15-45-65

Complejidad

■ Ventajas:

- Fácil implementación.
- No requiere memoria adicional.

■ Desventajas:

- Muy lento.
- Numerosas comparaciones.
- Pocos intercambios.

■ Total comparaciones

$$C = (n-1) + (n-2) + \dots + 2 + 1 = \frac{n*(n+1)}{2}$$

$$C = \frac{n^2 - n}{2}$$

■ Complejidad:

$O(n^2)$

ALGORITMO

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