



INSTITUTO POLITÉCNICO NACIONAL

---

ESCUELA SUPERIOR DE CÓMPUTO

## PRÁCTICA I: ORDENAMIENTO POR INSERCIÓN Y POR SELECCIÓN

ELÍAS LÓPEZ RIVERA

1CV1

ALGORITMOS Y ESTRUCTURAS DE DATOS

MAESTRO: MANUEL PORTILLO

Fecha de entrega: 01/04/2025



# Ordenamiento por inserción

## A. Algoritmo

### insertion sort

***Paso 1:** Nos posicionamos sobre la posición  $i$ -ésima del arreglo y lo fijamos, suponemos que todos los elementos con índice menor a  $i$  se encuentran ordenados.*

***Paso 2:** Revisamos los valores de las posiciones menores a  $i$ , con el contador  $j$ , si encontramos un elemento menor a nuestro valor fijado, detenemos el proceso, pues estamos suponiendo que todo lo que se encuentra detrás de la posición  $i$  está ordenado, en caso contrario intercambiamos las posiciones de la posición  $j$  y  $j-1$*

***Paso 3:** Repetimos este proceso desde la primera posición del arreglo hasta la antepenúltima*

## B. Pseudocódigo

---

### Algorithm 1 Insertion Sort

---

**Require:**  $[10000, 2, 400, \dots, n]$

**Ensure:**  $[1, 2, 3, \dots, n]$

#### INICIO

Desde  $i = 1$  hasta  $i < \text{tamarray}$

    elemento = array[i]

    Desde  $j = i - 1$  hasta  $j \geq 0 \ \& \ \text{array}[j] > \text{elemento}$

        array[j + 1] = array[j]

$j = j - 1$

        array[j + 1] = elemento

$i = i + 1$

#### FIN

---

## C. Codigo fuente en C

```

1 #include <stdio.h>
2
3 void swap(char*a,char*b)/*funcion swap, nos ayuda a intencambiar
4 los valores de dos variables tipo char*/
5 {
6     char aux=*a;
7     *a=*b;
8     *b=aux;
9 }
10
11 void imprimirarreglo(char c[4])/*imprime un arreglo de
12 caracteres a traves de un bucle for*/
13 {
14     for (int l = 0; l < 4; l++)
15     {
16         printf("%c ",*(c+l));
17     }
18     printf("\n");
19 }
20
21 void insertionsort(char A[4])//algoritmo de ordenamiento por insercion
22 {
23     for(int i=0;i<4-1;i++)//iteramos sobre todas las posiciones del
24     arreglo
25     {
26         int aux=*(A+i+1);/*Tomamos como elemento a
27         comparar al elemento con indice i+1*/
28         for(int j=i+1;(j>=1)&(aux<*(A+j-1));j--)/*Iteramos sobre las
29         posicones anteriores al indice i+1,
30         si y solo si las posiciones existen y se cumple
31         que nuestro a elemento a comparar es mayor*/
32         {
33             swap((A+j),(A+j-1));/*intercambiamos los valores
34             entre los indice j y j-1*/
35         }
36     }
37
38 int main()
39 {
40     char A[4]={'A','H','Z','C'};//inicializamos el arreglo a ordenar
41     imprimirarreglo(A);
42     insertionsort(A);
43     imprimirarreglo(A);
44 }

```

[Link copiar codigo](#)

## Compilación

```
PS C:\Users\Elías López\downloads\programacion> gcc practica_1_1_AlgyEst.c -o ejecutable_1.exe
PS C:\Users\Elías López\downloads\programacion> .\ejecutable_1.exe
A H Z C
A C H Z
PS C:\Users\Elías López\downloads\programacion> |
```

## Ordenamiento por selección

### A. Algoritmo

#### Selection sort

**Paso 1:** Nos posicionamos sobre la posición  $i$  del arreglo, fijamos el valor de esta posición y su índice

**Paso 2:** Iteramos desde la posición  $j=i$ , hasta el final del arreglo, si encontramos un valor que sea menor al fijado, fijamos ahora este valor y su índice

**Paso 3:** Al terminar esta iteración intercambiamos los valores de la posición  $i$  y el valor final fijado de índice, este corresponderá al mínimo

**Paso 4:** Repetimos este proceso con todas las posiciones del arreglo

## B. Pseudocodigo

---

### Algorithm 2 Selection Sort

---

**Require:**  $[10000, 2, 400, \dots, n]$

**Ensure:**  $[1, 2, 3, \dots, n]$

#### INICIO

```

Desde  $i = 1$  hasta  $i < n$ 
    elemento=array[i]
    int l
    Desde  $j = i + 1$  hasta  $j < n$ 
        Si  $A[j] \leq \text{elemento}$ 
            elemento=A[j]
             $l = j$ 
         $j = j + 1$ 
     $aux = A[j]$ 
     $A[j] = A[l]$ 
     $A[l] = aux$ 
     $i = i + 1$ 

```

#### FIN

---

## C. Codigo fuente

```

1  #include <stdio.h>
2
3  void swap(char*a, char*b) /*funcion swap nos ayuda a intercambiar
4  los valores de dos variables*/
5  {
6      char aux=*a;
7      *a=*b;
8      *b=aux;
9  }
10
11 void selectionsort(char A[5])
12 {
13     for(int i=0; i<5; i++) /*Iteramos sobre todas las posiciones del
14     arreglo*/
15     {
16         int aux=*(A+i); /*Definimos una variable auxiliar que
17         nos servira para guardar el minimo, la inicializamos
18         con el elemento i_esimo del arreglo*/
19         int l; /*variable donde almacenaremos el indice del
20         elemento menos*/
21         for(int j=i; j<5; j++) /*iteramos sobre los elementos
22         restantes del i_esimo al n-1_esimo*/
23         {

```

```

23         if (*(A+j) <= aux) /*si encontramos un elemento
24             menor a nuestro auxiliar*/
25         {
26             aux=*(A+j); /*Redefinimos a nuestro auxiliar*/
27             l=j; /*guardamos el nuevo indice del elemento menor*/
28         }
29     }
30     swap((A+i), (A+l)); /*Al encontra el elemento menor
31     simplemente lo intercambiamos con la posicion i_esima
32     de nuestro arreglo, pues es primera*/
33 }
34 }
35 void imprimirarreglo(char c[4]) /*imprime un arreglo de
36 caracteres a traves de un bucle for*/
37 {
38     for (int l = 0; l < 5; l++)
39     {
40         printf("%c ", *(c+l));
41     }
42     printf("\n");
43 }
44
45 int main()
46 {
47     char A[5]={'D','L','A','Z','W'}; //inicilalizamos el arreglo a
ordenar
48     imprimirarreglo(A);
49     selectionsort(A);
50     imprimirarreglo(A);
51 }

```

[Link copiar codigo](#)

## Compliación

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\Elías López\downloads\programacion> gcc practica_1_2_AlgvEst.c -o Ejecutable_2.exe
PS C:\Users\Elías López\downloads\programacion> .\Ejecutable_2.exe
D L A Z W
A D L W Z
PS C:\Users\Elías López\downloads\programacion>

```