

# ORDENAMIENTO

Algoritmos y Estructuras de Datos II

Lic. Ana María Company

# INTRODUCCIÓN

En algunos casos se necesita tener ordenados, siguiendo algún criterio, los datos con los que se trabaja para facilitar su tratamiento.

La *ordenación(clasificación)* es la operación de organizar un conjunto de datos en algún orden dado, por ejemplo:

- *Creciente/decreciente (ascendente/descendente) en datos numéricos*
- *En orden alfabético (A - Z / Z - A)*

# MÉTODOS DIRECTOS

- Selección directa
- Inserción directa (Baraja)
- Intercambio directo (Burbuja)

# SELECCIÓN DIRECTA

También conocido como *Obtención sucesiva de menores*.

Se recorre el vector y se selecciona en cada recorrido el menor elemento para situarlo en su lugar correspondiente.

El esquema básico del algoritmo es:

1. Se sitúa en  $v_1$  el menor valor entre  $v_1; \dots; v_n$   
Para ello se intercambian los valores de  $v_1$  y  $v_m$   
siendo  $v_m = \min\{v_k\}$  con  $k = 1; \dots; n$ .  
Ej: para un vector  $V(4, 5, 7, 1, 9, 8, 2)$

4 5 7 1 9 8 2

# SELECCIÓN DIRECTA

2. Se sitúa en  $v_2$  el menor valor entre  $v_2; \dots; v_n$

- Para ello se intercambian los valores de  $v_2$  y  $v_m$   
siendo  $v_m = \min\{v_k\}$  con  $k = 2; \dots; n$ .

*Siguiendo el ejemplo, en este paso se realiza el siguiente intercambio:*

**1** **5** 7 4 9 8 **2**

...

j-1. Se sitúa en  $v_{j-1}$  el menor valor entre  $v_{j-1}; \dots; v_n$

Para ello se intercambian los valores de  $v_{j-1}$  y  $v_m$   
siendo  $v_m = \min\{v_k\}$  con  $k = j-1; \dots; n$ .

...

n-1. Se sitúa en  $v_{n-1}$  el menor valor entre  $v_{n-1}; \dots; v_n$

Para ello se intercambian los valores de  $v_{n-1}$  y  $v_n$  si es necesario

*Para el ejemplo, el estado final es:* **1 2 4 5 7 8 9**

# SELECCIÓN DIRECTA - Ejemplo

```
70 void SeleccionDirecta( tVector pVector ) {  
71     // Efecto: se ordena v ascendentemente  
72     int i, j, posMenor;  
73     int valMenor, aux;  
74  
75     for ( i = 0; i < max-1; i++ ) { // n-1 pasadas  
76         valMenor = pVector[i];  
77         posMenor = i;  
78  
79         for ( j = i+1; j < max; j++ ) { // el número de comparaciones decrece  
80             if ( pVector[j] < valMenor ) {  
81                 //se actualiza el nuevo valor menor y la posición donde se encuentra  
82                 valMenor = pVector[j];  
83                 posMenor = j;  
84             }  
85         }  
86         if ( posMenor != i ) {  
87             //Si el menor no es pVector[i], se intercambian los valores  
88             aux = pVector[i];  
89             pVector[i] = pVector[posMenor];  
90             pVector[posMenor] = aux;  
91         }  
92     }  
93 }
```

# INSERCIÓN DIRECTA

También conocido como **Baraja**

Se recorre el vector **V** insertando el elemento  $v_i$  en su lugar correcto entre los ya ordenados  $v_1; \dots; v_{i-1}$ .

El esquema general de este algoritmo es:

1. Se considera  $v_1$  como primer elemento.
2. Se inserta  $v_2$  en su posición correspondiente en relación a  $v_1$  y  $v_2$ .
3. Se inserta  $v_3$  en su posición correspondiente en relación a  $v_1; \dots; v_3$ .
- ...
- i. Se inserta  $v_i$  en su posición correspondiente en relación a  $v_1; \dots; v_i$ .
- ...
- n. Se inserta  $v_n$  en su posición correspondiente en relación a  $v_1; \dots; v_n$ .

# INSERCIÓN DIRECTA - Ejemplo

```
95 void InsercionDirecta( tVector pVector ) {  
96     int i, j;  
97     int aux;  
98  
99     for ( i = 1; i < max; i++ ) {  
100         aux = pVector[i];  
101         j = i - 1;  
102         while ( (j >= 0) && (pVector[j] > aux) ) {  
103             pVector[j+1] = pVector[j];  
104             //Desplazamiento de los valores mayores que pVector[i]  
105             j = j-1;  
106         }  
107         pVector[j+1] = aux;  
108     }  
109 }
```



# INTERCAMBIO DIRECTO (Burbuja)

Compara cada elemento de un vector  $V$  con el siguiente, intercambiándolos de posición si están en el orden equivocado.

Será necesario recorrer varias veces todo el vector hasta que no se necesiten más intercambios.

# INTERCAMBIO DIRECTO (Burbuja)

- Se realizan  $(n-1)$  pasadas, transportando en cada pasada el menor o mayor elemento (según sea el caso) a su posición ideal.
- Al final de las  $(n-1)$  pasadas los elementos del arreglo estarán ordenados.
- También se conoce a este algoritmo como método de la **Burbuja**, por la forma en que se mueven los elementos durante los intercambios, como si fueran pequeñas "burbujas".

# INTERCAMBIO DIRECTO (Burbuja)

Pasos:

1. Comparar elemento en la 1ra posición con el elemento en la 2da posición, si están ordenados, se deja como está, caso contrario se realiza el intercambio.
2. Se comparan los dos elementos siguientes adyacentes: elemento de la 2da posición con el elemento de la 3ra posición, y de nuevo se intercambia si es necesario.
3. El proceso continúa hasta que cada elemento del arreglo haya sido comparado con sus elementos adyacentes y hayan sido intercambiados en los casos necesarios.

# INTERCAMBIO DIRECTO - Ejemplo

```
110 void OrdenacionPorIntercambio( tVector pVector ) {  
111     //Efecto: se ordena v ascendentemente  
112     int i, j;  
113     int aux;  
114  
115     for ( i = 0; i < max-1; i++ ) { //n-1 pasadas  
116         for ( j = 0; j < max-1; j++ ) { //n-1 comparaciones  
117             //for (j = 1; j < max-i; j++) //->opcion2  
118             if (pVector[j] > pVector[j+1]) {  
119                 //intercambio  
120                 aux = pVector[j];  
121                 pVector[j] = pVector[j+1];  
122                 pVector[j+1] = aux;  
123             }  
124         }  
125     }  
126 }
```

# BIBLIOGRAFÍA

Mark Allen Weiss - Estructuras de Datos y Algoritmos - Florida International University - Año: 1995 - Editorial: Addison-Wesley Iberoamericana .

Joyanes Aguilar, Luis - Programación en Pascal - 4ª Edición - Año: 2006 - Editorial: McGraw-Hill/Interamericana de España, S.A.U.

Cristóbal Pareja Flores, Manuel Ojeda Aciego, Ángel Andeyro Quesada, Carlos Rossi Jiménez - Algoritmos y Programación en Pascal.

Joyanes Aguilar, Luis - Fundamentos de la Programación. Algoritmos, Estructuras de Datos y Objetos - 3ª Edición - Editorial: McGraw-Hill