

## Algoritmos y Programación

### Métodos de Ordenamiento

2018

Prof. Miguel A. Fernández

1

## Métodos de Ordenamiento

- La necesidad de datos ordenados.
- Funciones o estrategias incluidas en los diseños de los lenguajes de programación.
- Algunas Metodologías de ordenamiento:
  - a) Inserción
  - b) Burbuja
  - c) Shell
  - d) Quick sort



Universidad Nacional  
de Entre Ríos

2

## Métodos de Ordenamiento

### ¿ Qué métodos vamos a analizar ?

#### BURBUJA

- a) Burbuja Básico
- a) Burbuja con Acotamiento
- b) Burbuja con Retroceso



Universidad Nacional  
de Entre Ríos

3

## Métodos de Ordenamiento - Burbuja

El procedimiento opera sobre el mismo vector donde se encuentran los datos que se requiere ordenar. No necesita vectores auxiliares y se basa en la comparación de elementos contiguos.

#### Procedimiento:

1. Se compara el elemento ubicado en la 1ra. posición con el de la 2da. posición. Si no se encuentran en el orden que se requiere (ascendente o descendente, según el caso) se los intercambia de lugar.
2. Se compara el elemento ubicado en la 2da. posición con el de la 3ra. posición. Si no se encuentran en el orden que se requiere (ascendente o descendente, según el caso) se los intercambia de lugar.
3. El procedimiento se repite hasta comparar el elemento ubicado en la penúltima posición con el de la última, intercambiándose si es necesario.

4

## Métodos de Ordenamiento – Burbuja

#### Rutina Elemental:

Sea un arreglo unidimensional "V" de "n" elementos, se requiere ordenarlos de menor a mayor:

Es necesario codificar una rutina para recorrer el arreglo comparando sus elementos contiguos e intercambiarlos si corresponde

```

Para i = 1, n-1, 1
  Si V(i) > V(i + 1)
    Aux := V(i)
    V(i) := V(i + 1)
    V(i + 1) := Aux
  Finsi
FinPara
  
```

¿ RESUELVE  
EL  
PROBLEMA  
ESTA  
RUTINA ?

Universidad Nacional  
de Entre Ríos

5

## Ordenamiento Burbuja

ARRAY	PASOS
1 2 3 4 5 6	
5 2 6 0 9 4	1. Compara 1er. con 2do. Elemento
Cambia	
2 5 6 0 9 4	2. Compara 2do. con 3er. Elemento
No Cambia	
2 5 6 0 9 4	3. Compara 3er. con 4to. Elemento
Cambia	
2 5 0 6 9 4	4. Compara 4to. con 5to. Elemento
No Cambia	
2 5 0 6 9 4	5. Compara 5to. con 6to. Elemento
Cambia	
2 5 0 6 4 9	¿ Quedó el array ordenado ?

6

## Ordenamiento Burbuja

### SEGUNDA RECORRIDA

ARRAY	PASOS
1 2 3 4 5 6	
2 5 0 6 4 9	1. Compara 1er. con 2do. Elemento
No Cambia	
2 5 0 6 4 9	2. Compara 2do. con 3er. Elemento
Cambia	
2 0 5 6 4 9	3. Compara 3er. con 4to. Elemento
No Cambia	
2 0 5 6 4 9	4. Compara 4to. con 5to. Elemento
Cambia	
2 0 5 4 6 9	5. Compara 5to. con 6to. Elemento
No Cambia	
2 0 5 4 6 9	¿ Quedó el array ordenado ?

7

## Ordenamiento Burbuja

### TERCERA RECORRIDA

ARRAY	PASOS
1 2 3 4 5 6	
2 0 5 4 6 9	1. Compara 1er. con 2do. Elemento
Cambia	
0 2 5 4 6 9	2. Compara 2do. con 3er. Elemento
No Cambia	
0 2 5 4 6 9	3. Compara 3er. con 4to. Elemento
Cambia	
0 2 4 5 6 9	4. Compara 4to. con 5to. Elemento
No Cambia	
0 2 4 5 6 9	5. Compara 5to. con 6to. Elemento
No Cambia	
0 2 4 5 6 9	¿ Quedó el array ordenado ?

8

## Ordenamiento Burbuja

### Luego: ¡ Necesidad de varias recorridas !

Como es necesario contemplar varias recorridas (ya que no hay certeza de que el arreglo quede ordenado), la rutina anterior debe incluirse dentro de otro esquema repetitivo.

El problema es cuando salir de dicho esquema (cuando queda ordenado el arreglo).

La estrategia consiste en determinar cuando en una recorrida completa no se hace ningún intercambio y para saber esto se recurre a la utilización de un campo de tipo booleano.



9

## Métodos Ordenamiento - Burbuja

### Una Solución posible:

```

Begin
  B := False
  Mientras B = False
  B := True
    Para i = 1, n-1, 1
      Si V(i) > V(i+1)
        Aux := V(i)
        V(i) := V(i+1)
        V(i+1) := Aux
        B := False
      Finsi
    FinPara
  FinMientras
End
  
```

Para indicar que faltan recorridas



10

## Métodos de Ordenamiento - Burbuja

### Rutina para arreglos compuestos

En estos casos se compara por el campo por el cual el arreglo debe quedar ordenado y en el intercambio de posiciones, cuando es necesario, puede asignarse el elemento compuesto haciendo referencia al nombre común.

#### Ejemplo:

Dado un arreglo de 80 elementos, cada uno de los cuales contiene el documento y el nombre de alumnos inscriptos en una asignatura, ordenarlos por número de documento.

Estructura del Arreglo:

```

Type
  Registro = Record
    NUD : integer;
    NOM : string;
  end;
  Alumnos = array [ 1..80 ] of Registro;
Var
  Alumno : Alumnos;
  
```



11

## Métodos de Ordenamiento - Burbuja

### Ordenamiento de arreglos compuestos:

```

Begin
  B := False
  Mientras B = False
  B := True
    Para i = 1, n-1, 1
      Si ALUMNO(i).NUD > ALUMNO(i+1).NUD
        Aux := Alumno(i)
        Alumno(i) := Alumno(i+1)
        Alumno(i+1) := Aux
        B := False
      Finsi
    FinPara
  FinMientras
End
  
```

Se compara por el campo NUD de ALUMNO

Se transfieren todos los campos del elemento.



12

## Métodos de Ordenamiento - Burbuja

¿ De qué tipo debe ser la variable auxiliar ?

```

Type
  Alumnado = Record
    NUD : integer;
    NOM : string;
  end;
Var
  Alumno : Alumnos;
  AUX : Alumnado;

```

Del tipo registro



13

## Ordenamiento Burbuja

Primer Caso de mejora: Burbuja con Acotamiento

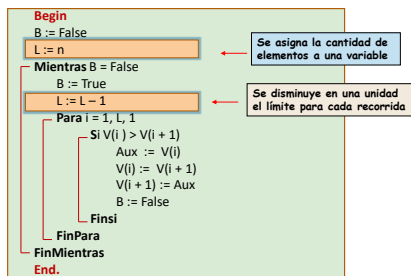
PRIMERA RECORRIDA						Posiciones	SEGUNDA RECORRIDA					
1	2	3	4	5	6		1	2	3	4	5	6
5	2	6	0	9	4		2	5	0	6	4	9
2	5	6	0	9	4		2	5	0	6	4	9
2	5	6	0	9	4		2	0	5	6	4	9
2	5	0	6	9	4		2	0	5	6	4	9
2	5	0	6	9	4		2	0	5	4	6	9
2	5	0	6	4	9		2	0	5	4	6	9

Dado que en la recorrida el elemento de mayor valor (en caso de estar ordenando de menor a mayor) queda siempre en la última posición, en la segunda recorrida el mayor (con excepción del anterior) queda en la antepenúltima posición y así sucesivamente, es posible mejorar la rutina acortando las sucesivas recorridas en un elemento.

14

## Métodos de Ordenamiento

La rutina para ordenar un vector "V" de "n" elementos sería:



15

## Métodos de Ordenamiento

Segundo caso de mejora: Una sola recorrida con retroceso.

5	7	0	9	4	6
5	0	7	9	4	6
0	5	7	9	4	6
0	5	7	4	9	6
0	5	4	7	9	6
0	4	5	7	9	6
0	4	5	7	6	9
0	4	5	6	7	9
0	2	4	5	6	9

Se compara: 5 con 7: No Cambia  
Se compara: 7 con 0: Cambia  
Retroceso:  
Se compara: 5 con 0 Cambia.  
Retroceso: Fin Retroceso. Retoma  
Se compara: 7 con 9: No cambia.  
Se compara: 9 con 4: Cambia.  
Retroceso:  
Se compara: 7 con 4: Cambia.  
Retroceso:  
Se compara: 5 con 4: Cambia.  
Retroceso:  
Se compara: 0 con 4: No Cambia.  
Retoma:  
Se compara: 9 con 6: Cambia.  
Retroceso:  
Se Compara: 7 con 6: Cambia  
Retroceso:  
Se compara : 5 con 6 : No cambia  
Retoma:  
Fin de Recorrida.

16

## Métodos de Ordenamiento

Segundo caso de mejora: Una sola recorrida con retroceso.

Consiste en hacer una sola recorrida en el vector, pero cuando se efectúa un intercambio se debe retroceder y comparar los elementos anteriores para asegurar el orden de los elementos.

El proceso de retroceso se realiza mientras se efectúen intercambios y se interrumpe por dos motivos:

- en una comparación no se efectúa intercambio y
- se llegó al inicio del vector.

Al detenerse el retroceso, se retoma la comparación en el punto en que inició la recorrida hacia atrás.



17

## Métodos de Ordenamiento

Rutina Burbuja con Retroceso

```

Para i = 1, n-1, 1
  z := i
  Mientras i > 0 y V(i) > V(i + 1)
    Aux := V(i)
    V(i) := V(i + 1)
    V(i + 1) := Aux
    i := i - 1
  Finmientras
  i := z
FinPara

```



18

## Métodos de Ordenamiento

### Burbuja con acotamiento

```

Begin
B := False
L := n
Mientras B = True
  B := False
  L := L - 1
  Para i = 1, L, 1
    Si V(i) > V(i + 1)
      Aux := V(i)
      V(i) := V(i + 1)
      V(i + 1) := Aux
  B := False
FinSi
FinPara
FinMientras
End.

```

### Burbuja con retroceso

```

Begin
Para i = 1, n-1, 1
  Z := i
  Mientras i > 0 and V(i) > V(i + 1)
    Aux := V(i)
    V(i) := V(i + 1)
    V(i + 1) := Aux
    i := i - 1
  Finmientras
  i := z
FinPara
End.

```

19

## Arreglos Compuestos - Registros

**¡ TIEMPO  
DE  
TRABAJO !**



## Métodos de Ordenamiento

1. Ingresar en forma desordenada 150 números reales e imprimirlos ordenados de menor a mayor junto con el orden en que ingresó.
2. Ingresar el Nombre, Nro.de Documento, y fecha de ingreso de los 120 detenidos de en un penal y emitir un listado ordenado por Nro.de Documento.  
El ingreso se realiza sin seguir orden alguno.
3. Se deben ingresar los siguientes datos de los libros que forman parte de una biblioteca ( el ingreso se realiza sin seguir orden alguno).
  - Nombre del Autor
  - Título del Libro
  - Nombre de la Editorial
 Se requiere ingresar dichos datos y almacenarlos en una estructura de datos de tipo compuesto y luego imprimir un listado ordenado por título del libro.
4. Ingresar los datos de los siniestros de automotores que denuncian los asegurados de una compañía de seguros:
  - Nro.de Dominio del automotor
  - Fecha del Siniestro
  - Código de siniestro (rango 1 a 280).
 Con estos datos se debe imprimir:
  - a) Un listado que contenga la cantidad total de siniestros ingresados discriminados por código de siniestro con el siguiente formato:  
**Código de Siniestro - Descripción - Cantidad**
  - b) Un listado de todos los siniestros ingresados ordenado por número de dominio  
**Nro.de Dominio - Fecha - Descripción del Siniestro**
 La descripción de cada código de siniestro se ingresa al principio del algoritmo.

21