

## Bloque 3. Tema 3.

### Operaciones avanzadas con Vectores

1. Búsqueda en vectores ordenados: búsqueda binaria
2. Matrices
  1. Introducción
  2. Sintaxis
  3. Ejemplos
3. Operaciones con matrices
  1. Recorrido de una matriz: por filas, por columnas
  2. Traspuesta, suma, producto

#### Bibliografía:

- Biondi y Clavel, 1989
- Joyanes
- Weiss
- Cairó

### 3.1. Búsqueda en vectores ordenados: búsqueda binaria

**Algoritmo** Búsqueda\_binaria (V, N, X, pos, existe) es

V: Vector [1..M] de T; {p. dato}

N: numérico; {p. dato}

X: T; {p. dato}

Pos: numérico; {p. resultado}

Existe: lógico; {p. resultado}

Inf, Sup, Med: numérico;

#### Inicio

Inf := 1;

Sup := N;

Mientras Inf <> Sup hacer

    Med := (Inf + Sup) div 2;

    Si X > V(Med) entonces

        Inf := Med + 1;

    Sino {X <= V(Med) }

        Sup := Med;

    finsi

Fin mientras

Si V(inf) = X entonces

    existe := cierto;

    pos := inf;

sino

    existe := falso;

finsi

**Fin**

## 3.2. Matrices

### 3.2.1. Introducción

- ¿Qué ocurre cuando el tipo base de un *array* es otro *array*?

Mat = vector[1..4] de (vector[1..5] de enteros);

X: Mat;

X

X(1)					X(2)					X(3)					X(4)					

Tenemos un array de 20 enteros, organizados en 4 bloques de 5 enteros cada uno: X(1) tendrá X(1,1), X(1,2), X(1,3), X(1,4), X(1,5) y así X(2), X(3) y X(4)

- Obviamente, para acceder a cada elemento individual serán necesarios dos índices: uno por cada dimensión.

X(1)	X[1,1]	X[1,2]	X[1,3]	X[1,4]	X[1,5]
X(2)	X[2,1]	X[2,2]	X[2,3]	X[2,4]	X[2,5]
X(3)	X[3,1]	X[3,2]	X[3,3]	X[3,4]	X[3,5]
X(4)	X[4,1]	X[4,2]	X[4,3]	X[4,4]	X[4,5]

### 3.2.2. Sintaxis

Además de la sintaxis anterior, que es genérica:

**<tipo-matriz> ::=**

**<tipo> = vector [<tipo-indice>{,<tipo-indice>}] de <tipo-base>**

**<tipo-indice> ::= <elemento-inicial> .. <elemento-final>**

### 3.2.3. Ejemplos

Notas = vector[1..4,1..9] de numérico; {Notas: 4 cursos x 9 asignaturas}

Temperaturas = vector[1..12,1..31] de numérico; {Temperaturas anuales x meses}

Nombres = vector[1..4,1..40] de cadenas; {Nombres de 40 alumnos x 4 cursos}

N: Notas;

T: Temperaturas; {T(6, 1): temperatura asociada al 1 de junio}

## 3.3. Operaciones sobre matrices

### 3.3.1. Recorrido

- También se puede aplicar el tratamiento secuencial sobre los arrays en general y sobre las matrices en particular.

En el caso de un vector:

– **Esquema 1**

iniciar tratamiento;

**para** I de 1 a N-1 **hace**

tratar X(I); {el elemento i-ésimo de la secuencia es X(I)}

**finpara**;

tratar X(N);

– **Esquema 2**

iniciar tratamiento;

**para** I de 1 a N **hace**

tratar X(I); {el elemento i-ésimo de la secuencia es X(I)}

**finpara**;

Los elementos de la secuencia son escalares.

En el caso de un vector multidimensional (por ejemplo,  $A_{M \times N}$ )

- las filas (o las columnas) de la matriz forman también una secuencia  
(fila1), (fila2), ..., (filaM)  
(X(1,1), ..., X(1,N)), (X(2,1), ..., X(2,N)), ..., (X(M,1), ..., X(M,N))
- el elemento i-ésimo de la secuencia será una fila o una columna
- a su vez, ese elemento puede necesitar también un tratamiento secuencial

### Recorrido de una matriz por filas

**Algoritmo Leer\_matriz (A, M, N) es**

{declaración de tipos}

matriz = vector[1..100, 1..100] de numérico;

{declaración de parámetros}

**A:** matriz; {p. dato-resultado}

**M, N:** numérico; {p. dato; porción utilizable de la matriz}

**I:** numérico; {entero, se usará como índice}

**Inicio**

**para** I desde 1 hasta M **hacer**

leer\_fila (A, I, N);

**fin para**

**Fin**

**Algoritmo Leer\_fila (A, fila, max\_col) es**  
{declaración de tipos}  
matriz = vector[1..100, 1..100] de numérico;  
{declaración de parámetros}  
A: matriz; {p. dato-resultado}  
fila, max\_col: numérico; {p. dato}  
J: numérico; {entero, se usará como índice}  
**Inicio**  
    para J desde 1 hasta max\_col hacer  
        escribir "Dame el elemento ", J, " de la fila ", fila, ": ";  
        leer A(fila, J);  
    fin para  
**Fin**

**Generalmente se escribirá de forma compacta:**

    para I desde 1 hasta M hacer  
        para J desde 1 hasta N hacer  
            tratar A(I, J);  
        fin para  
    fin para

## **Recorrido de una matriz por columnas**

**Algoritmo Mostrar\_matriz (A, M, N) es**  
{declaración de tipos}  
matriz = vector[1..100, 1..100] de numérico;  
{declaración de parámetros}  
A: matriz; {p. dato-resultado}  
M, N: numérico; {p. datos; porción utilizable de la matriz}  
J: numérico; {entero, se usará como índice}  
**Inicio**  
    para J desde 1 hasta N hacer {J=1..N, cada columna J de A}  
        mostrar\_columna (A, M, J);  
    fin para  
**Fin**

**Algoritmo mostrar\_columna (A, max\_fil, col) es**  
 {declaración de tipos}  
 matriz = vector[1..100, 1..100] de numérico;  
 {declaración de parámetros}  
 A: matriz; {p. dato-resultado}  
 col, max\_fil: numérico; {p. dato}  
 I: numérico; {entero, se usará como índice}  
**Inicio**  
     para I desde 1 hasta max\_fil hacer  
         escribir "El elemento ", I, " de la columna ", col, "es: ";  
         escribir A(I, col);  
     fin para  
**Fin**

**Generalmente se escribirá de forma compacta:**

para J desde 1 hasta N hacer  
     para I desde 1 hasta M hacer  
         tratar A(I, J);  
     fin para  
 fin para

### 3.3.2. Traspuesta, suma y producto

- **Suma**

**Resumen:**

Dadas dos matrices A y B de tamaños 100 x 100, en las que están ocupadas M x N posiciones, calcúlese en C la suma de A + B

**Algoritmo Suma (A, B, C, M, N) es**

A, B: matriz; {p. dato}

C: matriz; {p. resultado}

M, N: numérico; {p. dato}

- **Traspuesta**

**Resumen:**

Dada una matriz A de tamaño 100 x 100, en la que están ocupadas M x N posiciones, transfórmese A en su traspuesta, sin utilizar ninguna estructura auxiliar

**Algoritmo Traspuesta (A, M, N) es**

A: matriz; {p. dato-resultado}

M, N: numérico; {p. dato}

- **Producto**

**Resumen:**

Dadas dos matrices A y B de tamaños máximos 100 x 100, en las que están ocupadas  $A_{M \times N}$  y  $B_{N \times P}$  posiciones, calcúlese en C el producto de A x B

Algoritmo Producto (A, B, C, M, N, P) es

A, B: matriz; {p. dato;  $A_{M \times N}$  y  $B_{N \times P}$ }

C: matriz; {p. resultado;  $C_{M \times P}$ }

M, N, P: numérico; {p. dato}

### 3.3.3. Ejemplos

#### Uso de matrices en soluciones a problemas

- **El tres en raya**

TABLERO = vector[1..3, 1..3] de carácter; {'X', 'O', ' '}

- **El juego de los barcos**

JUGADOR = registro de

propios = vector [1..8, 1..8] de carácter; {' ', 'B'}

jugados = vector[1..8,1..8] de carácter; {' ', 'T', 'H', 'A'}

fin registro

- **El ajedrez**

casilla = registro de

ocupada: lógico;

ficha: carácter; {'p', 't', 'c', 'a', 'R'/'r', 'd'/'r'}

color: numérico; {1 blanco, 2 negro}

fin registro

TABLERO = vector[1..8, 1..8] de casilla;