Lección 5.1: Arreglo de dos dimensiones

Ha obtenido 0 punto(s) sobre 0 hasta ahora.

Las matrices

Las matrices aparecen por primera vez hacia el año 1850, introducidas por J.J. Sylvester  
El desarrollo inicial de la teoría se debe al matemático W.R. Hamilton en 1853  
En 1858, A. Cayley introduce la notación matricial como una forma abreviada de escribir un sistema de *m* ecuaciones lineales con *n* incógnitas.

Las matrices se utilizan en el cálculo numérico, en la resolución de sistemas de ecuaciones lineales, de las ecuaciones diferenciales y de las derivadas parciales. Además de su utilidad para el estudio de sistemas de ecuaciones lineales, las matrices aparecen de forma natural en geometría, estadística, economía, informática, física, etc...

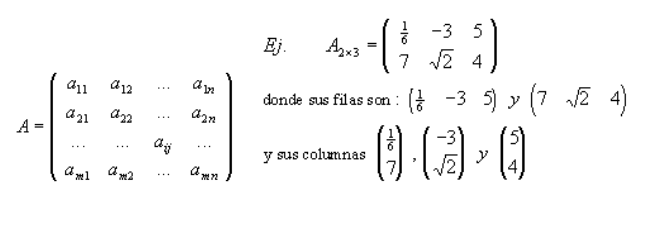
La utilización de matrices (arrays) constituye actualmente una parte esencial de los lenguajes de programación, ya que la mayoría de los datos se introducen en los ordenadores como tablas organizadas en filas y columnas : hojas de cálculo, bases de datos,...

**Concepto de matriz**

Una matriz es un conjunto de elementos de cualquier naturaleza aunque, en general, suelen ser números ordenados en filas y columnas.

Se llama **matriz** de orden  "m × n"   a un conjunto rectangular de elementos  aij  dispuestos en   **m  filas** y en  **n  columnas.** El orden de una matriz también se **denomina dimensión o tamaño**, siendo  m  y  n  números naturales.

Las matrices se denotan con letras mayúsculas: A, B, C, ... y los elementos de las mismas con letras minúsculas y subíndices que indican el lugar ocupado: a, b, c, ... Un elemento genérico que ocupe la fila  i  y la columna  j   se escribe  aij . Si el elemento genérico aparece entre paréntesis también representa a toda la matriz : A = (aij)

  
Cuando nos referimos indistintamente a filas o columnas hablamos de líneas.

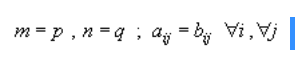
El número total de elementos de una matriz  A*m*×*n*  es   m·n

En matemáticas, tanto las **Listas** como las **Tablas** reciben el nombre genérico de matrices.

*Una lista numérica es un conjunto de números dispuestos uno a continuación del otro.*

         **MATRICES IGUALES**

Dos matrices  A = (*aij*)*m*×*n*  y  B = (*bij*)*p*×*q*  son iguales, sí y solo si, tienen en los mismo lugares elementos iguales, es decir :



         **ALGUNOS TIPOS DE MATRICES**

Hay algunas matrices que aparecen frecuentemente y que según su forma, sus elementos, ... reciben nombres diferentes :

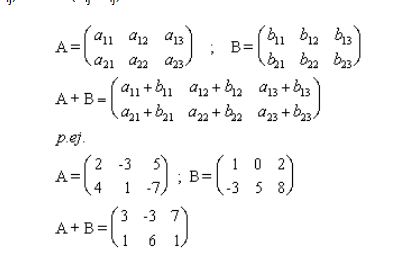
|  |  |  |
| --- | --- | --- |
| *Tipo de matriz* | *Definición* | *Ejemplo* |
| FILA | *Aquella matriz que tiene una sola fila, siendo su orden*1×n |  |
| COLUMNA | *Aquella matriz que tiene una sola columna, siendo su orden*m×1 |  |
| RECTANGULAR | *Aquella matriz que tiene distinto número de filas que de columnas,  siendo su orden* |  |
| TRASPUESTA | *Dada una matriz*A*, se llama traspuesta de*A*a la matriz que se obtiene cambiando ordenadamente las filas por las columnas. Se representa por*At  ó  AT |  |
| OPUESTA | *La matriz opuesta de una dada es la que resulta de sustituir cada elemento por su opuesto. La opuesta de*A*es*-A. |  |
| NULA | *Si todos sus elementos son cero. También se denomina matriz cero y se denota por 0m×n* |  |
| CUADRADA | *Aquella matriz que tiene igual número de filas que de columnas, m = n, diciendose que la matriz es de orden n.* Diagonal principal : *son los elementos  a*11*, a*22*, ..., a*nn Diagonal secundaria : *son los elementos  a*ij*con*i+j = n+1 Traza de una matriz cuadrada : *es la suma de los elementos de la diagonal principal*trA. |  |
| SIMÉTRICA | *Es una matriz cuadrada que es igual a su traspuesta.* A = At  , *a*ij= *a*ji |  |
| ANTISIMÉTRICA | *Es una matriz cuadrada que es igual a la opuesta de su traspuesta.* A = -At  , *a*ij= *-a*ji  *Necesariamente  a*ii*=*0 |  |
| DIAGONAL | *Es una matriz cuadrada que tiene todos sus elementos nulos excepto los de la diagonal principal* |  |
| ESCALAR | *Es una matriz cuadrada que tiene todos sus elementos nulos excepto los de la diagonal principal que son iguales* |  |
| IDENTIDAD | *Es una matriz cuadrada que tiene todos sus elementos nulos excepto los de la diagonal principal que son iguales a*1. *Tambien se denomina matriz unidad.* |  |
| TRIANGULAR | *Es una matriz cuadrada que tiene todos los elementos por encima (por debajo) de la diagonal principal nulos.* |  |
| ORTOGONAL | *Una matriz ortogonal es necesariamente cuadrada e invertible :*A-1 = AT La inversa de una matriz ortogonal es una matriz ortogonal. El producto de dos matrices ortogonales es una matriz ortogonal. El determinante de una matriz ortogonal vale +1 ó -1. |  |
| NORMAL | *Una matriz es normal si conmuta con su traspuesta. Las matrices simétricas, antisimétricas u ortogonales son necesariamente normales.* |  |
| INVERSA | *Decimos que una matriz cuadrada*A*tiene inversa,*A-1, *si se verifica que :* A·A-1 = A-1·A = I |  |

Para establecer las reglas que rigen el cálculo con matrices se desarrolla un álgebra semejante al álgebra ordinaria, pero en lugar de operar con números lo hacemos con matrices.

**Operaciones con matrices**

**Suma de matrices**

La suma de dos matrices  A = (aij)m×n  y  B = (bij)p×q  de la misma dimensión (equidimensionales) : m = p  y  n = q  es otra matriz  C = A+B = (cij)m×n = (aij+bij)



Es una ley de composición interna con las siguientes

**Propiedades :**

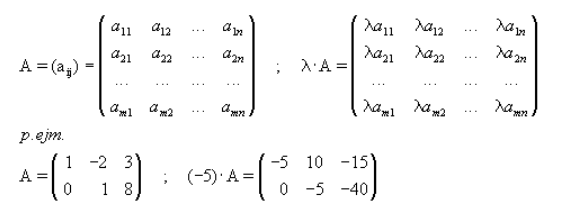
· Asociativa : A+(B+C) = (A+B)+C  
· Conmutativa : A+B = B+A  
· Elem. neutro : ( matriz cero 0m×n ) , 0+A = A+0 = A  
· Elem. simétrico : ( matriz opuesta -A ) , A + (-A) = (-A) + A = 0

Al conjunto de las matrices de dimensión  m×n cuyos elementos son números reales lo vamos a representar por  Mm×n  y como hemos visto, por cumplir las propiedades anteriores,  ( M, + ) es un grupo abeliano.

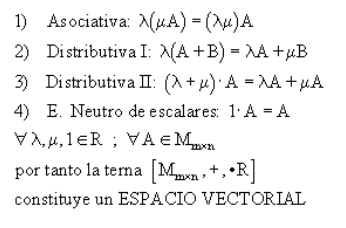
¡¡ La suma y diferencia de dos matrices NO está definida si sus dimensiones son distintas. !!

**Producto de un número real por una matriz**

Para multiplicar un escalar por una matriz se multiplica el escalar por todos los elementos de la matriz, obteniéndose otra matriz del mismo orden.



Es una ley de composición externa con las siguientes propiedades:



**Producto de matrices**

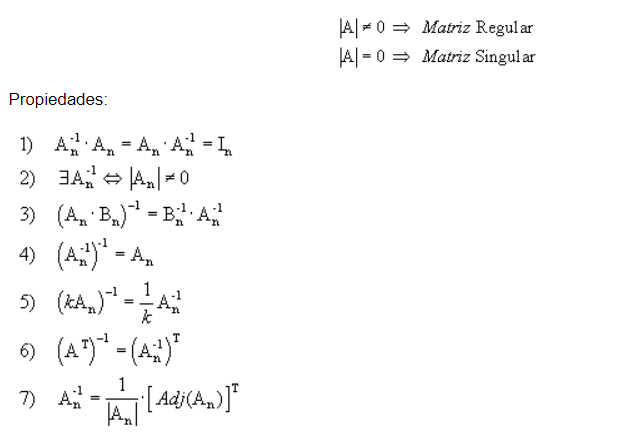
Dadas dos matrices  A = (aij)*m*×*n*  y  B = (bij)*p*×*q  donde  n = p, es decir, el número de columnas de la primera matriz*A*es igual al número de filas de la matriz*B*, se define el producto*A·B *de la siguiente forma :*

El elemento a que ocupa el lugar  (i, j)  en la matriz producto se obtiene sumando los productos de cada elemento de la fila  i  de la matriz  A por el correspondiente de la columna  j  de la matriz B.

**Matriz inversa**

Se llama matriz inversa de una matriz cuadrada  An  y la representamos por  A-1  , a la matriz que verifica la siguiente propiedad : A-1·A = A·A-1 = I

Decimos que una matriz cuadrada es  *"regular"*  si su determinante es distinto de cero, y es  *"singular"*  si su determinante es igual a cero.



         Sólo existe matriz inversa de una matriz cuadrada si ésta es *regular.*

         La matriz inversa de una matriz cuadrada, si existe, es única.

         Entre matrices NO existe la operación de división, la matriz inversa realiza funciones análogas.

**Métodos para hallar la matriz inversa:**

       Aplicando la definición

       Por el método de Gauss

       Por determinantes

Arreglos Bidimensionales

**Arreglos Bidimensionales (Matrices)**

**Introducción**

Una MATRIZ o TABLA, es un arreglo de dos dimensiones, por lo cual se manejan dos índices; el primer índice se refiere a la fila o renglón y el segundo a la columna; gráficamente lo podemos entender así:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Col 1 | Col 2 | Col 3 | Col 4 |
| Fila 1 |  |  |  |  |
| Fila 2 |  |  |  |  |
| Fila 3 |  |  |  |  |
| Fila 4 |  |  |  |  |

Para hacer referencia a un elemento de la matriz se tiene que indicar con dos índices:

**Matriz [Fila][Columna]**

Ejemplo: M[3][4]=7, M[1][2]=4, M[5][3]=ERROR, M[2][1]=8

|  |  |  |  |
| --- | --- | --- | --- |
| Col 1 | Col 2 | Col 3 | Col 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 4 | 1 | 1 |
| 8 | 3 | 0 | 0 |
| 2 | 6 | 1 | 7 |
| 3 | 7 | 8 | 4 |

**Declaración de Matrices**

La declaración de una matriz es similar a la de un Arreglo Unidimensional (Vector), con la diferencia de que hay que agregar un índice para referenciar la nueva dimensión de la matriz, la sintaxis es la siguiente:

NombreDelArreglo: ARRAY [# de Filas][# de Columnas] DE TipoDeDatos

Ejemplo: La declaración de la matriz mostrada en el ejemplo de arriba sería de la siguiente forma:

M: ARRAY [4][4] DE Enteros

**Ejemplo con Matrices**

Diseñe un Pseucocódigo que contenga en la memoria de la computadora una matriz de 3\*3 con números enteros leídos del teclado.

Pseudocódigo MATRIZ

Variables

MAT: ARRAY [3][3] DE Enteros

Fila, COlumna: Enteros

INICIO

FOR Fila=1 TO 3 DO

INICIO

FOR Columna=1 TO 3 DO

INICIO

ESCRIBE 'Teclee un número: '

LEE MAT[Fila][Columna]

FIN

FIN

FIN

### Animador de algoritmos en Matrices

**Recorrido de una matriz por filas**

<http://www.lsi.us.es/docencia/asignaturas/ip1/trabajos/Subhtmls/Recorrido%20de%20una%20matriz%20por%20filas.htm>

**Recorrido de una matriz por columnas**

<http://www.lsi.us.es/docencia/asignaturas/ip1/trabajos/Subhtmls/Recorrido%20de%20una%20matriz%20por%20columnas.htm>

**Recorrido en espiral de una matriz**

<http://www.lsi.us.es/docencia/asignaturas/ip1/trabajos/Subhtmls/Recorrido%20de%20una%20matriz%20por%20espiral.htm>

Algoritmos de Búsqueda

Los procesos de búsqueda involucran recorrer un arreglo completo con el fin de encontrar algo. Lo más común es buscar el menor o mayor elemento (cuando es puede establecer un orden), o buscar el índice de un elemento determinado.

Para buscar el menor o mayor elemento de un arreglo, podemos usar la estrategia, de suponer que el primero o el último es el menor (mayor), para luego ir comparando con cada uno de los elementos, e ir actualizando el menor (mayor). A esto se le llama Búsqueda Lineal.

**Definición:**

**–**Para encontrar un dato dentro de un arreglo, para ello existen diversos algoritmos que varían en complejidad, eficiencia, tamaño del dominio de búsqueda.

**Algoritmos de Búsqueda:**

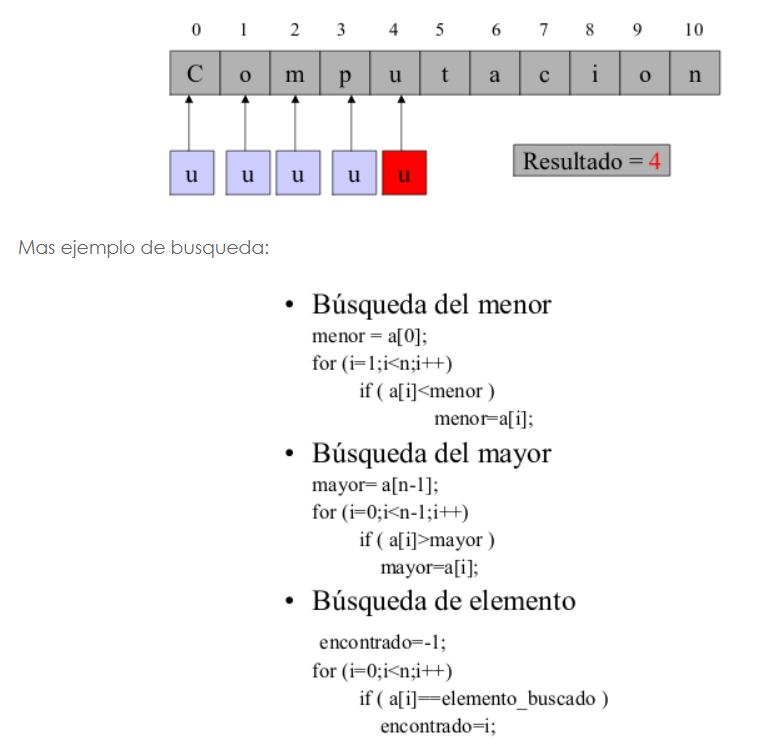
– Búsqueda Secuencial

– Búsqueda Binaria

**Búsqueda Secuencial:**

Consiste en ir comparando el elemento que se busca con cada elemento del arreglo hasta cuando se encuentra.

Busquemos el elementos **‘u’**



**Eficiencia y Complejidad**

Considerando la Cantidad de Comparaciones

**– Mejor Caso:**El elemento buscado está en la primera posición. Es decir, se hace una sola comparación. **(1)**

**– Peor Caso:**El elemento buscado está en la última posición. Necesitando igual cantidad de comparaciones que de elementos el arreglo. **(n)**

**– En Promedio:**El elemento buscado estará cerca de la mitad. Necesitando en promedio, la mitad de comparaciones que de elementos. **(n/2)**

Por lo tanto, la velocidad de ejecución depende linealmente del tamaño del arreglo.

**Búsqueda Binaria**

En el caso anterior de búsqueda se asume que los elementos están en cualquier orden. En el peor de los casos deben hacerse **(n)** operaciones de comparación.

Una búsqueda más eficiente puede hacerse sobre un arreglo ordenado. Una de éstas es la Búsqueda Binaria.

La Búsqueda Binaria, compara si el valor buscado está en la mitad superior o inferior. En la que esté, subdivido nuevamente, y así sucesivamente hasta encontrar el valor.

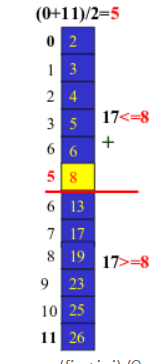
**Ejemplo:**

Supuesto: Arreglo con datos ordenados en forma ascendente:

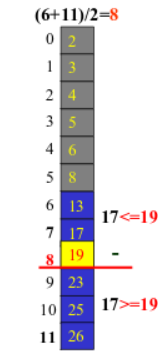
**i<k a[i]<a[k]**

•Estamos buscando la posición del **valor 17**

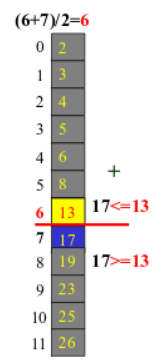
**1er ciclo :**ini=0 y fin=11 calculamos m=(fin+ini)/2, **m=(11+0)/2=5**, el valor en la posicion m es **8**, y el valor que buscamos esta en la **2da mitad** del arreglo.



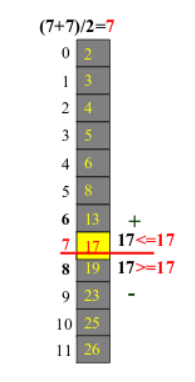
**2do ciclo :**ini=m+1 y fin=11 calculamos m=(fin+ini)/2, **m=(11+6)/2=8**, el valor en la posicion m es **19**, y el valor que buscamos esta en la **1ra mitad** del subarreglo.



**3er ciclo :**ini=6 y fin=m-1 calculamos m=(fin+ini)/2, **m=(7+6)/2=6**, el valor en la posicion m es **13**, y el valor que buscamos esta en la **2da mitad** del subarreglo.



**4to ciclo :**ini=7 y fin=7 calculamos m=(fin+ini)/2, **m=(7+7)/2=7**, el valor en la posicion m es **17**, y es el valor que buscamos.



**Complejidad y Eficiencia**

• Contando Comparaciones

**– Mejor Caso:** El elemento buscado está en el centro. Por lo tanto, se hace una sola comparación **(1)**.

**– Peor Caso:** El elemento buscado está en una esquina. Necesitando **log2(n)** cantidad de comparaciones.

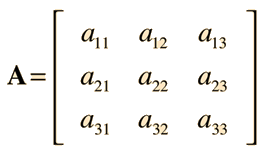
**– En Promedio:** Serán algo como **log2(n/2)**

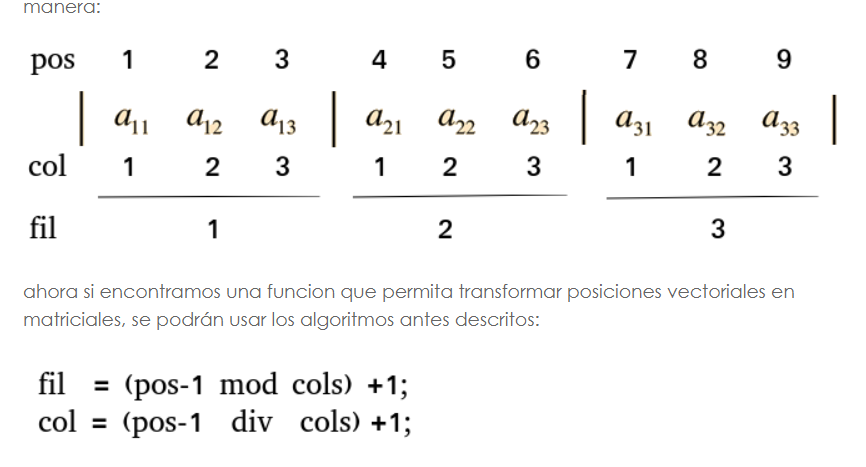
Por lo tanto, la velocidad de ejecución depende logarítmicamente del tamaño del arreglo.

### Ordenamientos en Matrices

Las técnicas de ordenamientos descritos en los arreglos de una dimensión, aplican en los arreglo de dos dimensiones.

para poder usar estas técnicas debemos encontrar un mecanismo que me permita relaciones la dirección vectoriales(Posición) con direcciones matriciales(filas y columnas):



dicha matriz podemos ver como si fuera un arreglo de una dimensión de la siguiente  manera:

ahora si encontramos una funcion que permita transformar posiciones vectoriales en matriciales, se podrán usar los algoritmos antes descritos:

una ves encontrado la funciones que me permitirán usar los ordenamiento vectoriales en matrices, podemos, realizar un ordenamiento para su demostración :

**Ejemplo 1: realizar la busqueda secuencial usando el algoritmos vectoriales.**

**procedure** Matriz.**busq\_sec**(ele: integer; var f, c: integer);  
**var**  
  dim,i:integer;  
  sw:boolean;  
**begin**  
 **dim:=fils\*cols;**  
  sw:=false;  
  i:=1;  
  while(i<=dim)and(NOT sw)do  
  begin  
     **f:=((i-1) MOD cols)+1;**  
     **c:=((i-1) DIV cols)+1;**  
     if(ele=**elem[f,c]**)then  
       sw:=true  
     else  
       i:=i+1;  
  end;  
  if(sw=false)then  
  begin  
    f:=-1;  
    c:=-1;  
  end;  
end;

**Ejemplo 2: realizar el ordenamiento por intercambio usando el algoritmos vectoriales.**

**procedure**Matriz**.ord\_inter();**  
**var**  
  p,d,dim,aux:integer;  
**begin**  
 **dim:=fils\*cols;**  
  for p:=1 to dim-1 do  
  begin  
    for d:=p+1 to dim do  
    begin  
     **fp:=((p-1) MOD cols)+1;  
      cp:=((p-1) DIV cols)+1;  
      fd:=((d-1) MOD cols)+1;  
      cd:=((d-1) DIV cols)+1;**  
      if(**elem[fd,cd]<elem[fp,cp]**)then  
      begin  
        **aux:=elem[fd,cd];  
        elem[fd,cd]:=elem[fp,cp];  
        elem[fp,cp]:=aux;**  
      end;  
    end;  
  end;  
**end;**

y asi sucesivamenteo que el resto de los algoritmos.