5.4.4. Operaciones con matrices

Desarrollemos ahora algunos programas que nos ayuden a efectuar operaciones con matrices como la suma o el producto.

Empecemos por diseñar un programa que sume dos matrices. Recuerda que sólo es posible sumar matrices con la misma dimensión, así que solicitaremos una sola vez el número de filas y columnas:

suma_matrices_3.py **suma_matrices.py**

```
# Pedimos la dimensión de las matrices,
m = int(raw_input('Dime el número de filas: '))
n = int(raw_input('Dime el número de columnas: '))

# Creamos dos matrices nulas...
A = []
for i in range(m):
    A.append( [0] * n )

B = []
```

```
for i in range(m):
  B.append( [0] * n )

# ... y leemos sus contenidos de teclado.
print 'Lectura de la matriz A'
for i in range(m):
  for j in range(n):
    A[i][j] = float(raw_input('Dame el componente (%d,%d): ' % (i, j)))

print 'Lectura de la matriz B'
for i in range(m):
  for j in range(n):
    B[i][j] = float(raw_input('Dame el componente (%d,%d): ' % (i, j)))
```

Hemos de tener claro cómo se calcula $C = A + B$. Si la dimensión de A y de B es $m \times n$, la matriz resultante será de esa misma dimensión, y su elemento de coordenadas (i, j), es decir, $C_{i,j}$, se calcula así:

$$C_{i,j} = A_{i,j} + B_{i,j},$$

para $1 \leq i \leq m$ y $1 \leq j \leq n$. Recuerda que la convención adoptada en la notación matemática hace que los índices de las matrices empiecen en 1, pero que en Python todo empieza en 0. Codifiquemos ese cálculo en Python.

suma_matrices_4.py **suma_matrices.py**

```
⋮

# Construímos otra matriz nula para albergar el resultado.
C = []
for i in range(m):
  C.append( [0] * n )

# Empieza el cálculo de la suma.
for i in range(m):
  for j in range(n):
    C[i][j] = A[i][j] + B[i][j]

# Y mostramos el resultado por pantalla
print "Suma:"
for i in range(m):
  for j in range(n):
    print C[i][j],
  print
```

EJERCICIOS

▶ 250 Diseña un programa que lea dos matrices y calcule la diferencia entre la primera y la segunda.

▶ 251 Diseña un programa que lea una matriz y un número y devuelva una nueva matriz: la que resulta de multiplicar la matriz por el número. (El producto de un número por una matriz es la matriz que resulta de multiplicar cada elemento por dicho número.)

Multiplicar matrices es un poco más difícil que sumarlas (y, por descontado, el operador * no calcula el producto de matrices). Una matriz A de dimensión $p \times q$ se puede multiplicar por otra matriz B si ésta es de dimensión $q \times r$, es decir, si el número de columnas de la primera es igual al número de filas de la segunda. Hemos de pedir, pues, el número de filas y columnas de la primera matriz y sólo el número de columnas de la segunda.

multiplica_matrices_3.py **multiplica_matrices.py**

```python
# Pedimos la dimensión de la primera matriz y el número de columnas de la segunda.
p = int(raw_input('Dime el número de filas de A: '))
q = int(raw_input('Dime el número de columnas de A (y filas de B): '))
r = int(raw_input('Dime el número de columnas de B: '))

# Creamos dos matrices nulas...
A = []
for i in range(p):
  A.append( [0] * q )

B = []
for i in range(q):
  B.append( [0] * r )

# ... y leemos sus contenidos de teclado.
print 'Lectura de la matriz A'
for i in range(p):
  for j in range(q):
    A[i][j] = float(raw_input('Dame el componente (%d,%d): ' % (i, j)))

print 'Lectura de la matriz B'
for i in range(q):
  for j in range(r):
    B[i][j] = float(raw_input('Dame el componente (%d,%d): ' % (i, j)))
```

Sigamos. La matriz resultante del producto es de dimensión $p \times r$:

multiplica_matrices.py

```python
# Creamos una matriz nula más para el resultado...
C = []
for i in range(p):
  C.append( [0] * r )
```

El elemento de coordenadas $C_{i,j}$ se calcula así:

$$C_{i,j} = \sum_{k=1}^{q} A_{i,k} \cdot B_{k,j},$$

para $1 \leq i \leq p$ y $1 \leq j \leq r$.

multiplica_matrices_4.py **multiplica_matrices.py**

```python
⋮
30
31 # Y efectuamos el cálculo del producto.
32 for i in range(p):
33   for j in range(r):
34     for k in range(q):
35       C[i][j] += A[i][k] * B[k][j]
```

¿Complicado? No tanto: a fin de cuentas las líneas 34–35 corresponden al cálculo de un sumatorio, algo que hemos codificado en Python una y otra vez.

Sólo falta mostrar el resultado por pantalla, pero ya hemos visto cómo se hace. Completa tú el programa.

EJERCICIOS

▶ 252 La traspuesta de una matriz A de dimensión $m \times n$ es una matriz A^T de dimensión

Otros usos de las matrices

De momento sólo hemos discutido aplicaciones numéricas de las matrices, pero son útiles en muchos otros campos. Por ejemplo, muchos juegos de ordenador representan informaciones mediante matrices:

- El tablero de tres en raya es una matriz de 3×3 en el que cada casilla está vacía o contiene la ficha de un jugador, así que podríamos codificar con el valor 0 el que esté vacía, con el valor 1 el que tenga una ficha de un jugador y con un 2 el que tenga una ficha del otro jugador.
- Un tablero de ajedrez es una matriz de 8×8 en el que cada casilla está vacía o contiene una pieza. ¿Cómo las codificarías?
- El tablero del juego del buscaminas es una matriz. En cada celda se codifica si hay bomba o no y si el usuario la ha descubierto ya o no.
- …

Las cámaras de video digitales permiten recoger imágenes, cada una de las cuales no es más que una matriz de valores. Si la imagen es en blanco y negro, cada valor es un número que representa la intensidad de brillo en ese punto; si la imagen es en color, cada casilla contiene tres valores: la intensidad de la componente roja, la de la componente verde y la de la componente azul. Los sistemas de visión artificial aplican transformaciones a esas matrices y las analizan para tratar de identificar en ellas determinados objetos.

$n \times m$ tal que $A^T_{i,j} = A_{j,i}$. Por ejemplo, si

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 12 & 6 \\ 1 & 0 & -3 \\ 10 & -1 & 0 \end{pmatrix}$$

entonces:

$$A^T = \begin{pmatrix} 1 & 2 & 1 & 10 \\ 2 & 12 & 0 & -1 \\ 3 & 6 & -3 & 0 \end{pmatrix}$$

Diseña un programa que lea una matriz y muestre su traspuesta.

▶ 253 Diseña un programa tal que lea una matriz A de dimensión $m \times n$ y muestre un vector v de talla n tal que

$$v_i = \sum_{j=1}^{m} A_{i,j},$$

para i entre 1 y n.

▶ 254 Diseña un programa que lea una matriz A de dimensión $m \times n$ y muestre un vector v de talla $\min(n, m)$ tal que

$$v_i = \sum_{j=1}^{i} \sum_{k=1}^{i} A_{j,k},$$

para i entre 1 y $\min(n, m)$.

▶ 255 Diseña un programa que determine si una matriz es prima o no. Una matriz A es prima si la suma de los elementos de cualquiera de sus filas es igual a la suma de los elementos de cualquiera de sus columnas.

▶ 256 Una matriz es diagonal superior si todos los elementos por debajo de la diagonal principal son nulos. Por ejemplo, esta matriz es diagonal superior:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 12 & 6 \\ 0 & 0 & -3 \\ 0 & 0 & 0 \end{pmatrix}$$

Diseña un programa que diga si una matriz es o no es diagonal superior.