

Librería numpy. Dos ejercicios resueltos

Producto de matrices

Sean A y B dos matrices cuadradas de dimensión n . Definimos la matriz C , producto de las matrices dadas, A y B , así:

$$C_{i,j} = \sum_{k=0}^{n-1} A_{i,k} * B_{k,j}$$

Se pide definir una función que realice este producto sin usar la operación $@$ de la librería. Luego, puede compararse si nuestra función llega al mismo resultado.

In [1]:

```
# Creación arrays a partir de Listas:

import numpy as np

a = np.array([[1., 2., 3.],
              [4., 5., 6.],
              [7., 8., 9.]
              ])

b = np.array([[10., 11., 12.],
              [13., 14., 15.],
              [16., 17., 18.]
              ])

print("a = ", a)
print(a.ndim, a.shape, a.dtype)    # dimensión, forma y tipo de los elementos

print()

print("b = ", b)
print(b.ndim, b.shape, b.dtype)    # dimensión, forma y tipo de los elementos
```

```
a = [[ 1.  2.  3.]
      [ 4.  5.  6.]
      [ 7.  8.  9.]]
2 (3, 3) float64
```

```
b = [[ 10.  11.  12.]
      [ 13.  14.  15.]
      [ 16.  17.  18.]]
2 (3, 3) float64
```

In [2]:



```
# Producto:

def producto_mat(a, b):
    # Sup.cuadradas y de la misma dimensión
    n = a.shape[0]
    c = np.zeros((n, n))
    for i in range(n):
        for j in range(n):
            for k in range(n):
                c[i, j] = c[i, j] + a[i, k] * b[k, j]
    return c

print(producto_mat(a, b))

print()

print(a@b)
```

```
[[ 84.  90.  96.]
 [201. 216. 231.]
 [318. 342. 366.]]
```

```
[[ 84.  90.  96.]
 [201. 216. 231.]
 [318. 342. 366.]]
```

Determinante de una matriz cuadrada mediante la regla del corazón

Calcula el valor de un determinante mediante la regla del corazón:

https://eprints.ucm.es/10939/1/ct01_2010.pdf (https://eprints.ucm.es/10939/1/ct01_2010.pdf)

In [3]:



```
# Producto:

def determinante_4_esquinas(a):
    # Pre.: Sup.cuadradas y de la misma dimensión
    n = a.shape[0]
    return determ(a, 0, 0, n)

def determ(a, i, j, n):
    # Pre.: Los índices a_i, a_j, a_i + n-1, a_j + n-1 están en los rangos de aa
    # Y aa es una matriz cuadrada
    if n==0:
        return 1
    elif n==1:
        return a[i, j]
    else:
        return (determ(a, i, j, n-1) * determ(a, i+1, j+1, n-1)
                - determ(a, i+1, j, n-1) *determ(a, i, j+1, n-1)
                ) / determ(a, i+1, j+1, n-2)

a = np.array([[1., 2., 3.],
              [4., 5., 4.],
              [3., 2., 1.]
              ])

print(a)

print(determinante_4_esquinas(a))

print(np.linalg.det(a))

print(determ(a, 1, 1, 1))
```

```
[[ 1.  2.  3.]
 [ 4.  5.  4.]
 [ 3.  2.  1.]]
-8.0
-8.0
5.0
```