INSTITUTO DE INNOVACIÓN Y TECNOLOGÍA APLICADA

Matrices – Aplicación en Mapeo y Filtrado

Matrices

Una matriz se trata de un arreglo de elementos, los cuales se encuentran en determinada posición. La misma se organiza mediante filas y columnas, donde cada elemento le corresponderá una posición en cada fila y columna.

Matrices

Las matrices pueden pensarse como listas de n elementos que se anidarán a una lista mucho mas grande, que justamente recibirá el nombre de “Matriz”.

Cada matriz puede tener un tamaño variable tanto en la longitud de sus filas como la de las columnas.

Matrices

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad \begin{pmatrix} A & B \\ C & D \end{pmatrix} \qquad \begin{matrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 1 \end{matrix}$$

$$\begin{matrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{matrix}$$

Matrices – Python

Dentro de Python, la declaración e inicialización de matrices se realiza mediante la anidación de vectores (listas) las cuales conformarán la matriz.

Cada lista anidada corresponde a una fila de la matriz, y las mismas pueden tener *n* elementos que serán las columnas.

Matrices – Python

Al trabajar con matrices, si queremos obtener un valor no existe mucha diferencia con una lista, sin embargo, para obtener dicho valor se debe especificar el “index” o la posición del vector que lo contiene y la columna donde se encuentra.

V = [[1, 2, 3], [4, 5, 6]] -> V[1, 2] = ?

Matrices - Python

```python
fil = int(input("Ingrese la cantidad de filas: "))
col = int(input("Ingrese la cantidad de columnas: "))

matriz = []

for i in range(fil):
    matriz.append([""]*col)

for fila in range(fil):
    for columna in range(col):
        print(f'Elemento del espacio ({fila+1}, {columna+1})')
        matriz[fila][columna] = int(input())
```

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]
[1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

Matrices – Aplicación en Webots

Las matrices podemos utilizarlas para representar los resultados del mapeo en la plataforma Erebus, siendo que la matriz resultante se convertirá en un array (utilizando la librería Numpy) y representando un mapa.

Filtrado de Matrices

Filtrado – Filas y columnas

Muchas veces, nuestros programas generarán matrices que tal vez contengan filas adicionales que no representen algo significativo para nosotros, por lo cual es necesario eliminar esos elementos que están de más.

De igual manera, es bueno saber la cantidad de filas y columnas que hay en una matriz generada.

Filtrado – Filas y columnas

```python
c = 0
## Cuenta la cantidad de filas que tiene una matriz
for elemento in matriz:
    print(elemento)
    c += 1
print("Cantidad total de filas:", c)
print("cant de columnas;", len(matriz[0]))
```

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
Cantidad total de filas: 21
cant de columnas; 16
```

Filtrado – Filas

Para eliminar las filas innecesarias, existen muchos métodos. Podemos tomar una fila de referencia la cual posea los valores que sean nulos, y realizar el filtrado mediante la referencia respecto a esa fila.

Filtrado – Filas

```
column_reference = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

result = [elem for elem in matriz_procesar if elem != column_reference]
```

```
[0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0]
[0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0]
```

Filtrado – Columnas

Al igual que para eliminar filas, existen muchas maneras de filtrar las columnas. Tal vez, la más factible es el utilizar la matriz transpuesta a la misma.

La matriz transpuesta surge a partir de intercambiar las filas de una matriz por sus columnas.

Filtrado – Columnas

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]
[1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

```
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

Filtrado – Columnas

Como podemos observar, la matriz cambia de orden y las filas se han convertido en columnas, mientras que las columnas se convirtieron en las filas.

Para filtrar las columnas de una matriz basta con obtener su transpuesta, utilizar el filtrado de filas y obtener nuevamente su nueva transpuesta.