Exercici: quadrat màgic

ETSEIB/GIE

18 de desembre de 2025

Un quadrat màgic és una matriu quadradada $n \times n$ d'enters positius en el rang $1, 2, \ldots, n^2$, de forma que cada posició és ocupada per un enter diferent i la suma d'enters de cada fila, columna i diagonals és igual.
Per exemple, el quadrat

2	7	6
9	5	1
4	3	8

la suma de qualsevol fila, columna, o diagonal dona 15.
Es demana dissenyar la funció esMagic que donat un DataFrame de dimensions quadrades i que cada cel·la conté un enter diferent, indiqui si es tracta d'un quadrat màgic o no.

Exemple

```
>>> import pandas as pd
>>> from io import StringIO
>>> df = pd.DataFrame([[ 1,  23,     16,     4,      21],
...                    [15,  14,      7,    18,      11],
...                    [24,  17,     13,     9,       2],
...                    [20,   8,     19,    12,       6],
...                    [ 5,  3,      10,    22,      25]],
...                   columns=['c1','c2','c3','c4','c5'],
...                 index=['f1','f2','f3','f4','f5'])
>>>
```

Exemple

```
>>> import pandas as pd
>>> from io import StringIO
>>> df = pd.DataFrame([[ 1,  23,     16,     4,      21],
...                    [15,  14,      7,    18,      11],
...                    [24,  17,     13,     9,       2],
...                    [20,   8,     19,    12,       6],
...                    [ 5,  3,      10,    22,      25]],
...                   columns=['c1','c2','c3','c4','c5'],
...                 index=['f1','f2','f3','f4','f5'])
>>>
```

Exemple

```
>>> import pandas as pd
>>> from io import StringIO
>>> df = pd.DataFrame([[ 1,  23,     16,     4,      21],
...                    [15,  14,      7,    18,      11],
...                    [24,  17,     13,     9,       2],
...                    [20,   8,     19,    12,       6],
...                    [ 5,  3,      10,    22,      25]],
...                   columns=['c1','c2','c3','c4','c5'],
...                 index=['f1','f2','f3','f4','f5'])
>>>
```

Exemple

```
>>> import pandas as pd
>>> from io import StringIO
>>> df = pd.DataFrame([[ 1,  23,     16,     4,      21],
...                    [15,  14,      7,    18,      11],
...                    [24,  17,     13,     9,       2],
...                    [20,   8,     19,    12,       6],
...                    [ 5,   3,     10,    22,      25]],
...                   columns=['c1','c2','c3','c4','c5'],
...                 index=['f1','f2','f3','f4','f5'])
>>>
```

Exemple

```
>>> import pandas as pd
>>> from io import StringIO
>>> df = pd.DataFrame([[ 1,  23,     16,     4,      21],
...                    [15,  14,      7,    18,      11],
...                    [24,  17,     13,     9,       2],
...                    [20,   8,     19,    12,       6],
...                    [ 5,  3,      10,    22,      25]],
...                   columns=['c1','c2','c3','c4','c5'],
...                 index=['f1','f2','f3','f4','f5'])
>>>
```

Exemple

```
>>> import pandas as pd
>>> from io import StringIO
>>> df = pd.DataFrame([[ 1,  23,     16,     4,      21],
...                    [15,  14,      7,    18,      11],
...                    [24,  17,     13,     9,       2],
...                    [20,   8,     19,    12,       6],
...                    [ 5,  3,      10,    22,      25]],
...                   columns=['c1','c2','c3','c4','c5'],
...                 index=['f1','f2','f3','f4','f5'])
>>>
```

Exemple

```
>>> df
    c1  c2  c3  c4  c5
f1   1  23  16   4  21
f2  15  14   7  18  11
f3  24  17  13   9   2
f4  20   8  19  12   6
f5   5   3  10  22  25
>>> # Sumem fila f1
>>> df.loc['f1'].sum()
65
>>> # Sumem columna c1
>>> df['c1'].sum()
65
```

Exemple

```
>>> df
    c1  c2  c3  c4  c5
f1   1  23  16   4  21
f2  15  14   7  18  11
f3  24  17  13   9   2
f4  20   8  19  12   6
f5   5   3  10  22  25
>>> # Sumem fila f1
>>> df.loc['f1'].sum()
65
>>> # Sumem columna c1
>>> df['c1'].sum()
65
```

Exemple

```
>>> df
    c1  c2  c3  c4  c5
f1   1  23  16   4  21
f2  15  14   7  18  11
f3  24  17  13   9   2
f4  20   8  19  12   6
f5   5   3  10  22  25
>>> # Sumem fila f1
>>> df.loc['f1'].sum()
65
>>> # Sumem columna c1
>>> df['c1'].sum()
65
```

Exemple

```
>>> df
    c1  c2  c3  c4  c5
f1   1  23  16   4  21
f2  15  14   7  18  11
f3  24  17  13   9   2
f4  20   8  19  12   6
f5   5   3  10  22  25
>>> # Sumem fila f1
>>> df.loc['f1'].sum()
65
>>> # Sumem columna c1
>>> df['c1'].sum()
65
```

Exemple

```
>>> df
    c1  c2  c3  c4  c5
f1   1  23  16   4  21
f2  15  14   7  18  11
f3  24  17  13   9   2
f4  20   8  19  12   6
f5   5   3  10  22  25
>>> # Sumem fila f1
>>> df.loc['f1'].sum()
65
>>> # Sumem columna c1
>>> df['c1'].sum()
65
```

Exemple

```
>>> df
    c1  c2  c3  c4  c5
f1   1  23  16   4  21
f2  15  14   7  18  11
f3  24  17  13   9   2
f4  20   8  19  12   6
f5   5   3  10  22  25
>>> # Sumem fila f1
>>> df.loc['f1'].sum()
65
>>> # Sumem columna c1
>>> df['c1'].sum()
65
```

Exemple

```
>>> df
    c1  c2  c3  c4  c5
f1   1  23  16   4  21
f2  15  14   7  18  11
f3  24  17  13   9   2
f4  20   8  19  12   6
f5   5   3  10  22  25
>>> # Sumem fila f1
>>> df.loc['f1'].sum()
65
>>> # Sumem columna c1
>>> df['c1'].sum()
65
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les columnes

```
>>> df.sum()
c1    65
c2    65
c3    65
c4    65
c5    65
dtype: int64
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les columnes

```
>>> df.sum()
c1    65
c2    65
c3    65
c4    65
c5    65
dtype: int64
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les columnes

```
>>> df.sum()
c1    65
c2    65
c3    65
c4    65
c5    65
dtype: int64
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les columnes

```
>>> df.sum() == 65
c1    True
c2    True
c3    True
c4    True
c5    True
dtype: bool
>>> all(df.sum() == 65)
True
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les columnes

```
>>> df.sum() == 65
c1    True
c2    True
c3    True
c4    True
c5    True
dtype: bool
>>> all(df.sum() == 65)
True
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les columnes

```
>>> df.sum() == 65
c1     True
c2     True
c3     True
c4     True
c5     True
dtype: bool
>>> all(df.sum() == 65)
True
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les columnes

```
>>> df.sum() == 65
c1    True
c2    True
c3    True
c4    True
c5    True
dtype: bool
>>> all(df.sum() == 65)
True
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les files

```
>>> df.sum(axis=1)
f1    65
f2    65
f3    65
f4    65
f5    65
dtype: int64
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les files

```
>>> df.sum(axis=1)
f1    65
f2    65
f3    65
f4    65
f5    65
dtype: int64
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les files

```
>>> df.sum(axis=1)
f1    65
f2    65
f3    65
f4    65
f5    65
dtype: int64
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les files

```
>>> df.sum(axis=1) == 65
f1    True
f2    True
f3    True
f4    True
f5    True
dtype: bool
>>> all(df.sum(axis=1) == 65)
True
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les files

```
>>> df.sum(axis=1) == 65
f1    True
f2    True
f3    True
f4    True
f5    True
dtype: bool
>>> all(df.sum(axis=1) == 65)
True
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les files

```
>>> df.sum(axis=1) == 65
f1    True
f2    True
f3    True
f4    True
f5    True
dtype: bool
>>> all(df.sum(axis=1) == 65)
True
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem totes les files

```
>>> df.sum(axis=1) == 65
f1    True
f2    True
f3    True
f4    True
f5    True
dtype: bool
>>> all(df.sum(axis=1) == 65)
True
```

Exemple

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem diagonals

```
>>> sum(map(lambda f,c: df.at[f,c], df.index, df.columns))
65
>>> sum(map(lambda f,c: df.at[f,c], reversed(df.index),
...         df.columns))
65
```

Exemple

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem diagonals

```
>>> sum(map(lambda f,c: df.at[f,c], df.index, df.columns))
65
>>> sum(map(lambda f,c: df.at[f,c], reversed(df.index),
...         df.columns))
65
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem diagonals

```
>>> sum(map(lambda f,c: df.at[f,c], df.index, df.columns))
65
>>> sum(map(lambda f,c: df.at[f,c], reversed(df.index),
...         df.columns))
65
```

	c1	c2	c3	c4	c5
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Sumem diagonals

```
>>> sum(map(lambda f,c: df.at[f,c], df.index, df.columns))
65
>>> sum(map(lambda f,c: df.at[f,c], reversed(df.index),
...         df.columns))
65
```

Solució

```
def esMagic(df):
    s = df.iloc[0].sum()
    return (df.shape[0] == df.shape[1]) and \
        all(n in df.values \
            for n in range(1, df.shape[0]**2+1)) and \
        all(df.sum() == s) and  \
        all(df.sum(axis=1) == s) and \
        (sum(map(lambda f,c: df.at[f,c],
                 df.index,
                 df.columns)) == s) and \
        (sum(map(lambda f,c: df.at[f,c],
                 reversed(df.index),
                df.columns)) == s)
```

```
>>> from qMagic import esMagic
>>> esMagic(df)
True
```

Solució

```
def esMagic(df):
    s = df.iloc[0].sum()
    return (df.shape[0] == df.shape[1]) and \
        all(n in df.values \
            for n in range(1, df.shape[0]**2+1)) and \
        all(df.sum() == s) and  \
        all(df.sum(axis=1) == s) and \
        (sum(map(lambda f,c: df.at[f,c],
                 df.index,
                 df.columns)) == s) and \
        (sum(map(lambda f,c: df.at[f,c],
                 reversed(df.index),
                df.columns)) == s)
```

```
>>> from qMagic import esMagic
>>> esMagic(df)
True
```

Solució

```
def esMagic(df):
    s = df.iloc[0].sum()
    return (df.shape[0] == df.shape[1]) and \
        all(n in df.values \
            for n in range(1, df.shape[0]**2+1)) and \
        all(df.sum() == s) and  \
        all(df.sum(axis=1) == s) and \
        (sum(map(lambda f,c: df.at[f,c],
                 df.index,
                 df.columns)) == s) and \
        (sum(map(lambda f,c: df.at[f,c],
                 reversed(df.index),
                df.columns)) == s)
```

```
>>> from qMagic import esMagic
>>> esMagic(df)
True
```

Solució

```
def esMagic(df):
    s = df.iloc[0].sum()
    return (df.shape[0] == df.shape[1]) and \
        all(n in df.values \
            for n in range(1, df.shape[0]**2+1)) and \
        all(df.sum() == s) and  \
        all(df.sum(axis=1) == s) and \
        (sum(map(lambda f,c: df.at[f,c],
                 df.index,
                 df.columns)) == s) and \
        (sum(map(lambda f,c: df.at[f,c],
                 reversed(df.index),
                 df.columns)) == s)
```

```
>>> from qMagic import esMagic
>>> esMagic(df)
True
```

Provant

```
>>> tresxtres= df.loc['f2':'f4','c2':'c4']
>>> tresxtres
    c2  c3  c4
f2  14   7  18
f3  17  13   9
f4   8  19  12
>>> esMagic(tresxtres)
False
```

	c1	**c2**	**c3**	**c4**	**c5**
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Provant

```
>>> tresxtres= df.loc['f2':'f4','c2':'c4']
>>> tresxtres
    c2  c3  c4
f2  14   7  18
f3  17  13   9
f4   8  19  12
>>> esMagic(tresxtres)
False
```

	c1	**c2**	**c3**	**c4**	**c5**
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Provant

```
>>> tresxtres= df.loc['f2':'f4','c2':'c4']
>>> tresxtres
    c2  c3  c4
f2  14   7  18
f3  17  13   9
f4   8  19  12
>>> esMagic(tresxtres)
False
```

	c1	**c2**	**c3**	**c4**	**c5**
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Provant

```
>>> tresxtres= df.loc['f2':'f4','c2':'c4']
>>> tresxtres
    c2  c3  c4
f2  14   7  18
f3  17  13   9
f4   8  19  12
>>> esMagic(tresxtres)
False
```

	c1	**c2**	**c3**	**c4**	**c5**
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25

Provant

```
>>> tresxtres= df.loc['f2':'f4','c2':'c4']
>>> tresxtres
    c2  c3  c4
f2  14   7  18
f3  17  13   9
f4   8  19  12
>>> esMagic(tresxtres)
False
```

	c1	**c2**	**c3**	**c4**	**c5**
f1	1	23	16	4	21
f2	15	14	7	18	11
f3	24	17	13	9	2
f4	20	8	19	12	6
f5	5	3	10	22	25