Dr. Song
Cristofer Belen
Richie Molina

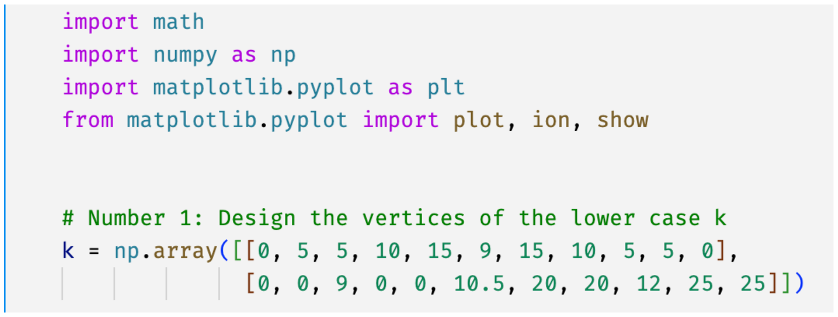
Fall 2022

AMAT 240-01 (Intro. to Linear Algebra)

Draw Lower Case Letter k

1. Design the vertices of the lower-case k.

a. $k = np.array([[0, 5, 5, 10, 15, 9, 15, 10, 5, 5, 0],$
$[0, 0, 9, 0, 0, 10.5, 20, 20, 12, 25, 25]])$

b. 

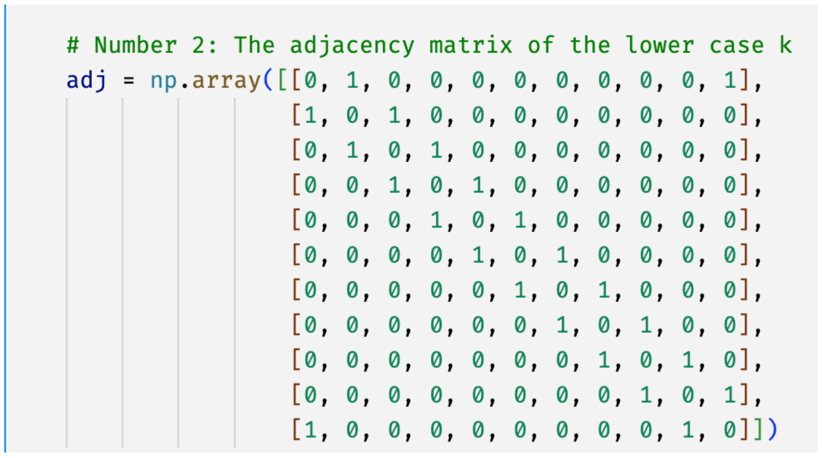
```
import math
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import plot, ion, show


# Number 1: Design the vertices of the lower case k
k = np.array([[0, 5, 5, 10, 15, 9, 15, 10, 5, 5, 0],
              [0, 0, 9, 0, 0, 10.5, 20, 20, 12, 25, 25]])
```

2. What is the adjacency matrix?

a. The **adjacency Matrix** is also called the **connection matrix.** This is a matrix containing rows and columns which are used to represent a simple labelled graph, with 0 or 1 in the position (V_i, V_j) according to the condition whether V_i and V_j are adjacent or not. It is a simple way to represent the finite graph containing *n* vertices of an *m* * *n* matrix M.

b. The **adjacency matrix** can also be referred to the vertex matrix. This is defined in the general form as the following: If the simple labelled graph has no self-loops, then the vertex matrix should have 0s in the diagonal. It is symmetric for the undirected graph The connection matrix is considered a square array where each row represents the out-nodes of a graph and each column represents the in-nodes of a graph. The first entry represents an edge between two nodes (vertices).

c. 

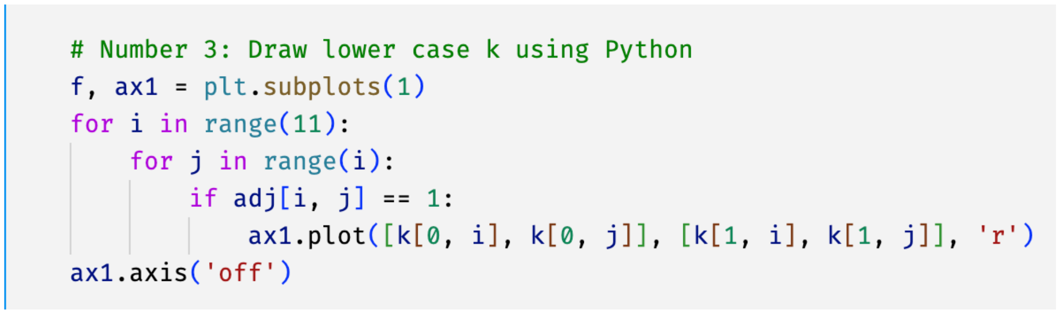
```
# Number 2: The adjacency matrix of the lower case k
adj = np.array([[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
                [1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
                [0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0],
                [0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0],
                [0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0],
                [0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
                [0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0],
                [0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0],
                [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0],
                [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1],
                [1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]])
```

Dr. Song
Cristofer Belen
Richie Molina

AMAT 240-01 (Intro. to Linear Algebra)
Draw Lower Case Letter k

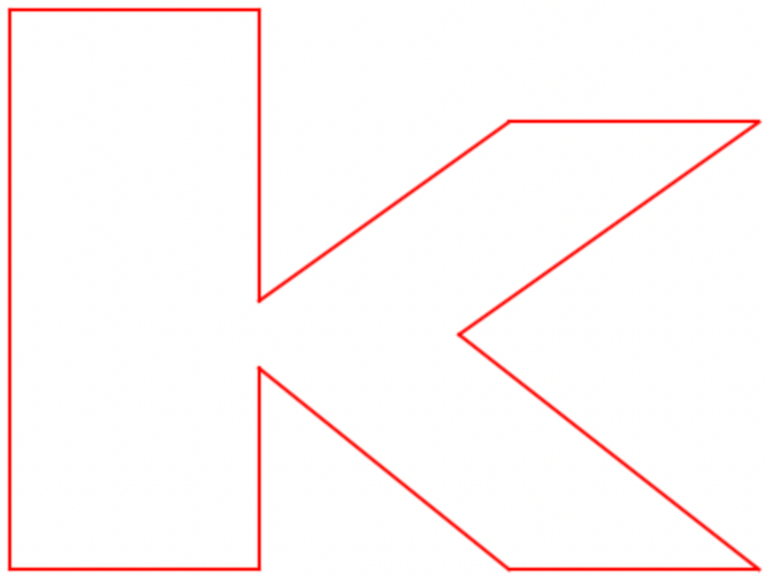
Fall 2022

3. Use Python to program a drawing of the lower-case k.

a. 

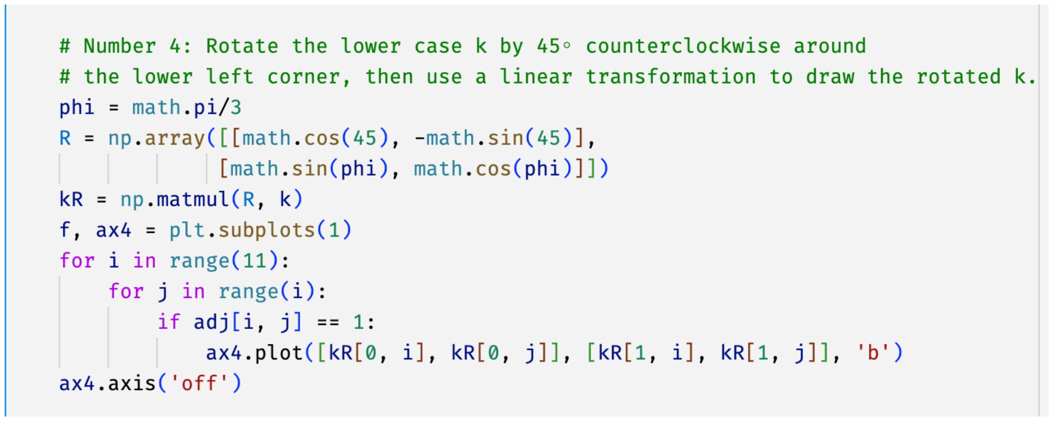
```
# Number 3: Draw lower case k using Python
f, ax1 = plt.subplots(1)
for i in range(11):
    for j in range(i):
        if adj[i, j] == 1:
            ax1.plot([k[0, i], k[0, j]], [k[1, i], k[1, j]], 'r')
ax1.axis('off')
```

...

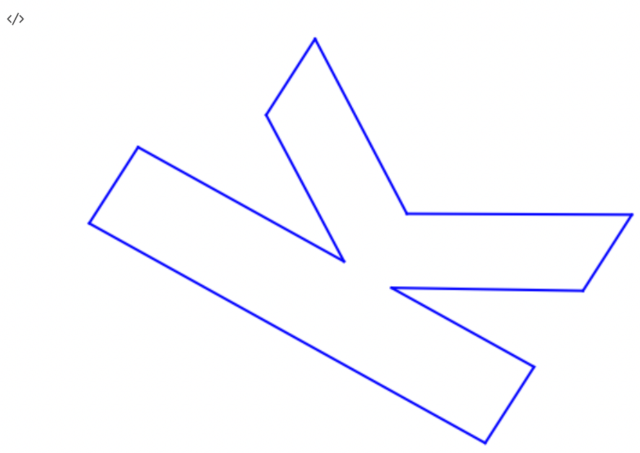


b.

4. Rotate the lower-case k by 45° counterclockwise around the lower left corner, then use a linear transformation to draw the rotated k.

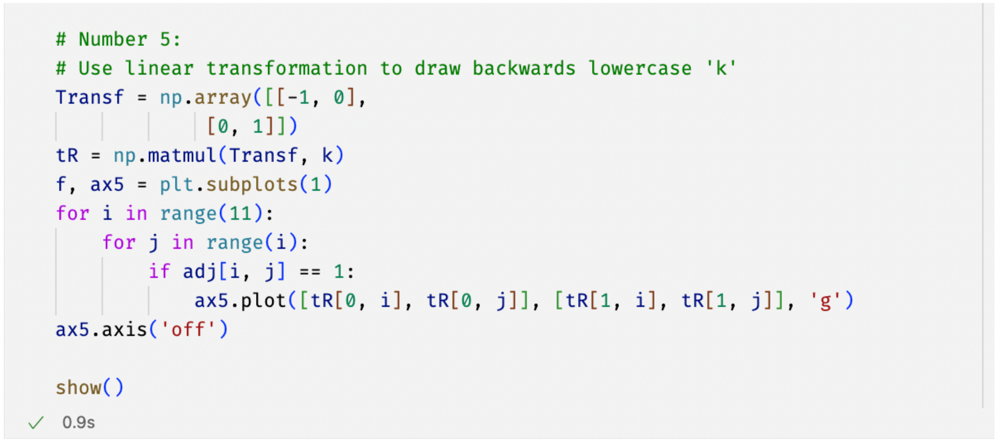
a. 

```
# Number 4: Rotate the lower case k by 45◦ counterclockwise around
# the lower left corner, then use a linear transformation to draw the rotated k.
phi = math.pi/3
R = np.array([[math.cos(45), -math.sin(45)],
              [math.sin(phi), math.cos(phi)]])
kR = np.matmul(R, k)
f, ax4 = plt.subplots(1)
for i in range(11):
    for j in range(i):
        if adj[i, j] == 1:
            ax4.plot([kR[0, i], kR[0, j]], [kR[1, i], kR[1, j]], 'b')
ax4.axis('off')
```



b.

5. Using a linear transformation to draw the backwards lowercase k.

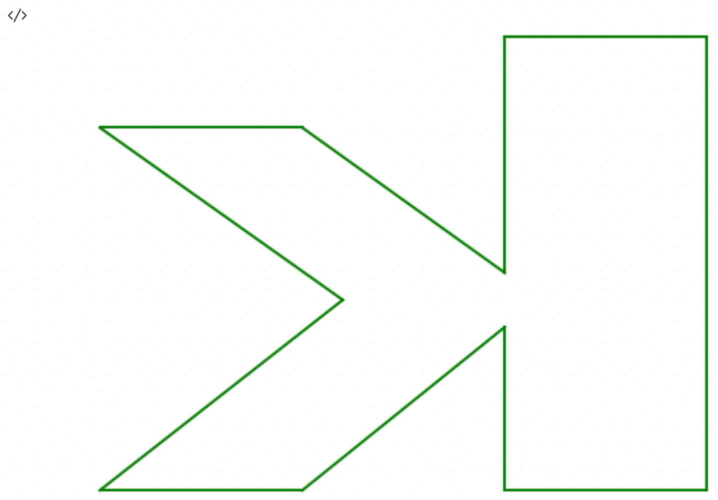


```
# Number 5:
# Use linear transformation to draw backwards lowercase 'k'
Transf = np.array([[-1, 0],
                   [0, 1]])
tR = np.matmul(Transf, k)
f, ax5 = plt.subplots(1)
for i in range(11):
    for j in range(i):
        if adj[i, j] == 1:
            ax5.plot([tR[0, i], tR[0, j]], [tR[1, i], tR[1, j]], 'g')
ax5.axis('off')

show()
```

[4] ✓ 0.9s

a.



b.