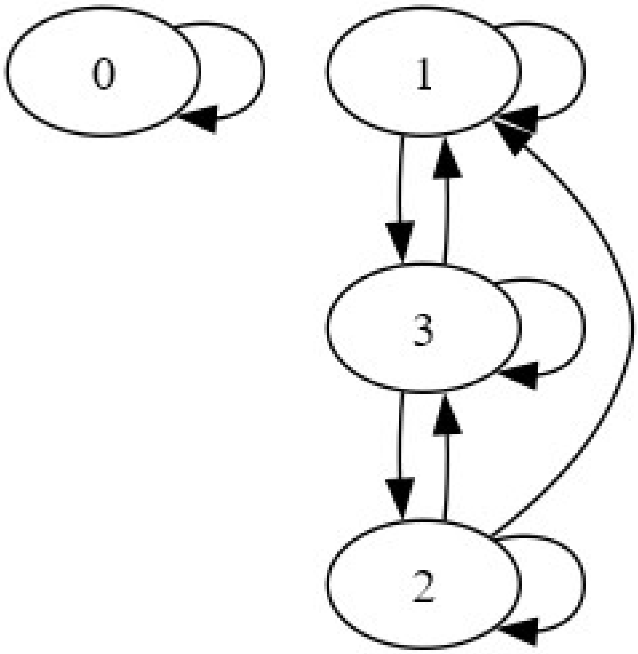
# 66310837

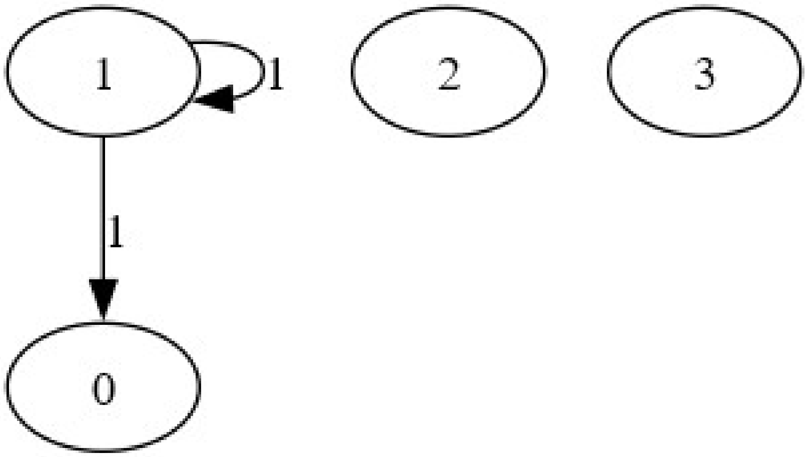
## นายจิรัฐ ฟองดา

```
!conda install --channel conda-forge pygraphviz

Channels:
 - conda-forge
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): ...working... done
Solving environment: ...working... done

# All requested packages already installed.
```

```
import numpy as np
import numpy.linalg as la
import matplotlib.pyplot as plt
from graph import *
import random

make_graph_adj_random(4)
```

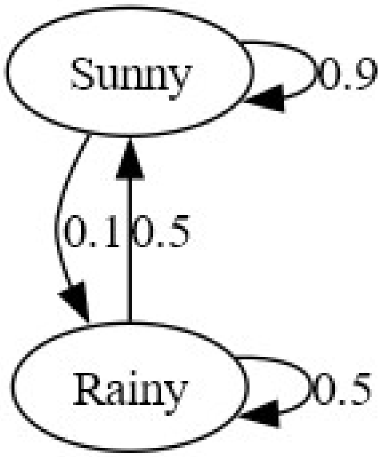


```
A=np.array([[0,1,0,0],
            [0,1,0,0],
            [0,0,0,0],
            [0,0,0,0]])

graph_matrix(A, mat_label=None, show_weights=True, round_digits=3)
```



## Example 1

```
A = np.array([[0.9,0.5],
              [0.1,0.5]])
labels = ["Sunny","Rainy"]
graph_matrix(A, mat_label=labels, show_weights=True, round_digits=3)
```



The weather today is sunny, What is the probability of sunny day tomorrow?

```
x0 = np.array([0,1])
x1 = A@x0
x2 = A@x1
x3 = A@x2
x3
```

```
array([0.78, 0.22])
```

```
its = 20
allx = np.zeros((2,its))
```

```
allx
```

```
array([[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.]])
```

```
x = np.array([0,1])
allx[:,0] = x
allx
```

```
array([[0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0.,
        0., 0., 0., 0.],
       [1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0.,
        0., 0., 0., 0.]])
```

```
for i in range(1,its):
    allx[:,i] = A@allx[:,i-1]
```

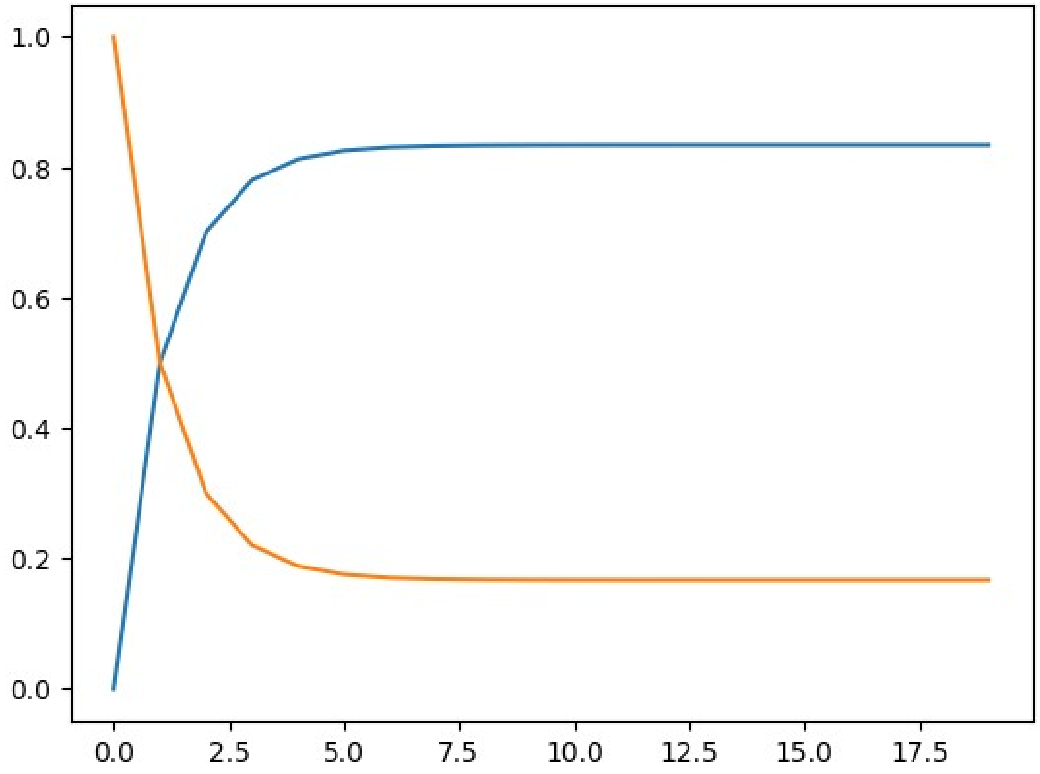
```
allx.T
```

```
array([[0.        , 1.        ],
       [0.5       , 0.5       ],
       [0.7       , 0.3       ],
       [0.78      , 0.22      ],
       [0.812     , 0.188     ],
       [0.8248    , 0.1752    ],
       [0.82992   , 0.17008   ],
       [0.831968  , 0.168032  ],
       [0.8327872 , 0.1672128 ],
       [0.83311488, 0.16688512],
       [0.83324595, 0.16675405],
       [0.83329838, 0.16670162],
       [0.83331935, 0.16668065],
       [0.83332774, 0.16667226],
       [0.8333311 , 0.1666689 ],
       [0.83333244, 0.16666756],
       [0.83333298, 0.16666702],
       [0.83333319, 0.16666681],
       [0.83333328, 0.16666672],
       [0.83333331, 0.16666669]])
```

```python
print('Probabilities of initial state:',allx[:,0])
print('Probabilities after 20 days:', allx[:,-1])
plt.plot(allx.T)
plt.xlabel('')
```

```
Probabilities of initial state: [0. 1.]
Probabilities after 20 days: [0.83333331 0.16666669]
```

```
Text(0.5, 0, '')
```



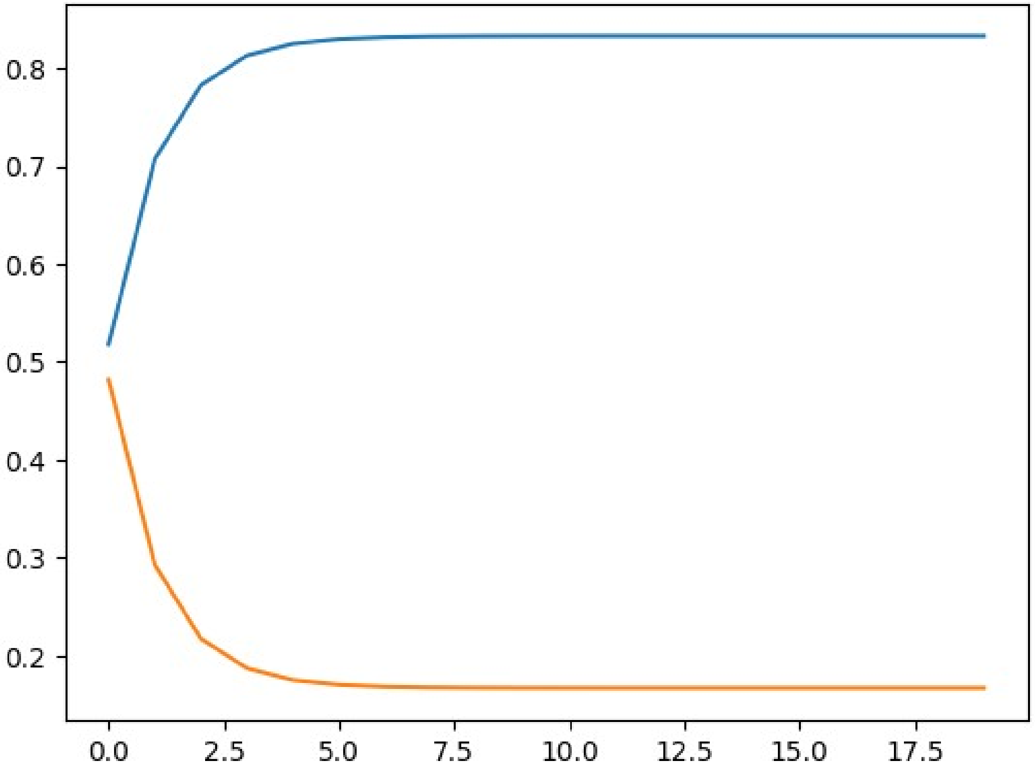
```python
a = random.random()
b = 1 - a
allx[:,0] = np.array([a,b])

for i in range(1,its):
    allx[:,i] = A@allx[:,i-1]

allx.T
```

```
array([[0.51814145, 0.48185855],
       [0.70725658, 0.29274342],
       [0.78290263, 0.21709737],
       [0.81316105, 0.18683895],
       [0.82526442, 0.17473558],
       [0.83010577, 0.16989423],
       [0.83204231, 0.16795769],
       [0.83281692, 0.16718308],
       [0.83312677, 0.16687323],
       [0.83325071, 0.16674929],
       [0.83330028, 0.16669972],
       [0.83332011, 0.16667989],
       [0.83332805, 0.16667195],
       [0.83333122, 0.16666878],
       [0.83333249, 0.16666751],
       [0.83333299, 0.16666701],
       [0.8333332 , 0.1666668 ],
       [0.83333328, 0.16666672],
       [0.83333331, 0.16666669],
       [0.83333332, 0.16666668]])
```

```
print('Probabilities of initial state:',allx[:,0])
print('Probabilities after 20 days:', allx[:,-1])
plt.plot(allx.T)
plt.xlabel('')
```

```
Probabilities of initial state: [0.51814145 0.48185855]
Probabilities after 20 days: [0.83333332 0.16666668]
```

```
Text(0.5, 0, '')
```



```python
lambdas, U = la.eig(A)
print(lambdas)
print(U[:,0])
```

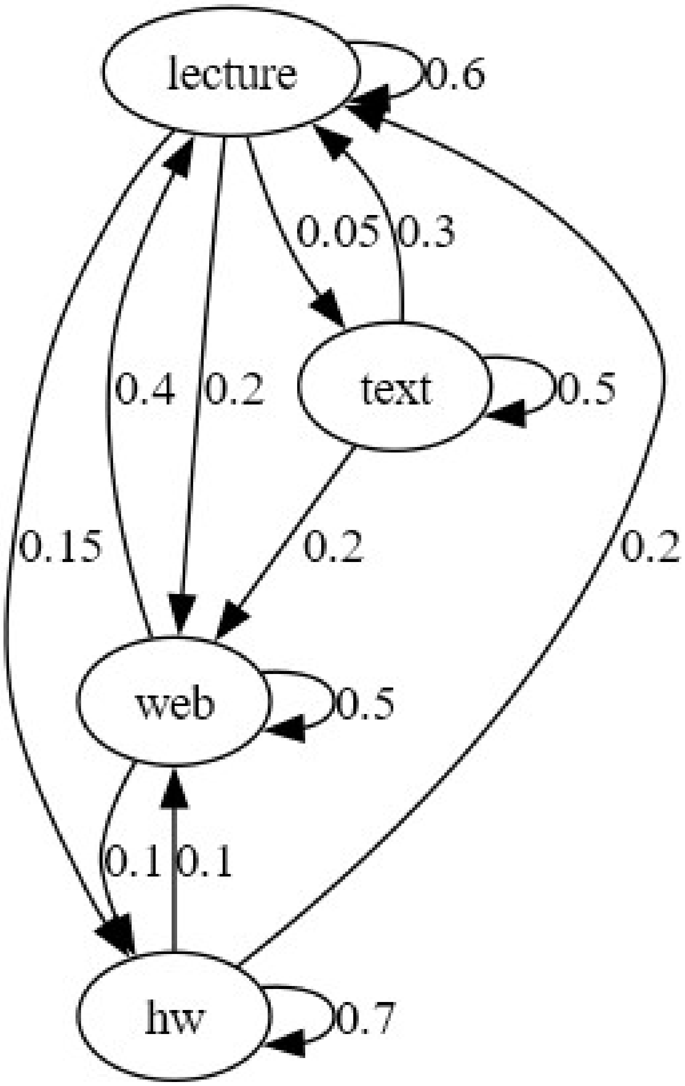
```
[1.  0.4]
[0.98058068 0.19611614]
```

```python
U[:,0] / la.norm(U[:,0], 1)
```

```
array([0.83333333, 0.16666667])
```

## Example 2

```python
activity_names = ['lecture','web','hw','text']
A = np.array([
    [0.6, 0.4, 0.2, 0.3], # lecture
    [0.2, 0.5, 0.1, 0.2], # web
    [0.15,0.1, 0.7, 0.0], # hw
    [0.05,0.0, 0.0, 0.5]  # text
])
graph_matrix(A, mat_label=activity_names, show_weights=True,
round_digits=3)
```

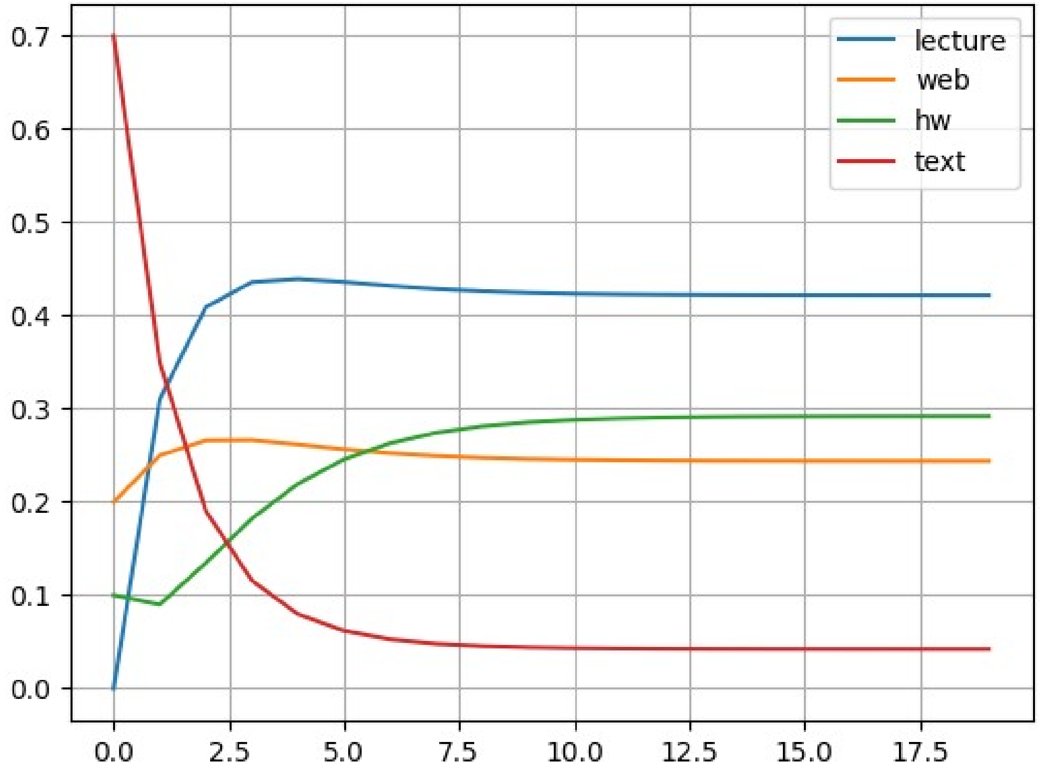


```
x0 = np.array([0,0,0,1])
x1 = A@x0
x2 = A@x1
x3 = A@x2
x3
```

```
array([0.4425, 0.2715, 0.133 , 0.153 ])
```

```
its = 20
allx = np.zeros((4,its))

allx[:,0] = np.array([0, 0.2, 0.1, 0.7])

for i in range(1,its):
    allx[:,i] = A@allx[:,i-1]

allx.T
```

```
array([[0.        , 0.2       , 0.1       , 0.7       ],
       [0.31      , 0.25      , 0.09      , 0.35      ],
       [0.409     , 0.266     , 0.1345    , 0.1905    ],
       [0.43585   , 0.26635   , 0.1821    , 0.1157    ],
       [0.43918   , 0.261695  , 0.2194825 , 0.0796425 ],
       [0.43597525, 0.25656025, 0.24568425, 0.06178025],
       [0.43188018, 0.25239965, 0.26303129, 0.05268889],
       [0.42850089, 0.24941677, 0.27414389, 0.04793845],
       [0.42607755, 0.24741064, 0.28111753, 0.04539427],
       [0.42445258, 0.24611144, 0.28543497, 0.04400101],
       [0.42340342, 0.24528993, 0.28808351, 0.04322314],
       [0.42274167, 0.24477863, 0.28969796, 0.04278174],
       [0.42233057, 0.24446379, 0.29067769, 0.04252795],
       [0.42207778, 0.24427137, 0.29127035, 0.0423805 ],
       [0.42192344, 0.24415438, 0.29162805, 0.04229414],
       [0.42182966, 0.24408351, 0.29184359, 0.04224324],
       [0.42177289, 0.24404069, 0.29197331, 0.0422131 ],
       [0.42173861, 0.24401488, 0.29205132, 0.0421952 ],
       [0.42171794, 0.24399933, 0.2920982 , 0.04218453],
       [0.42170549, 0.24398998, 0.29212637, 0.04217816]])

plt.plot(allx.T)
plt.legend(activity_names)
plt.grid()
```



```
lambdas, U = la.eig(A)
print(lambdas)
print(U[:,0])
```

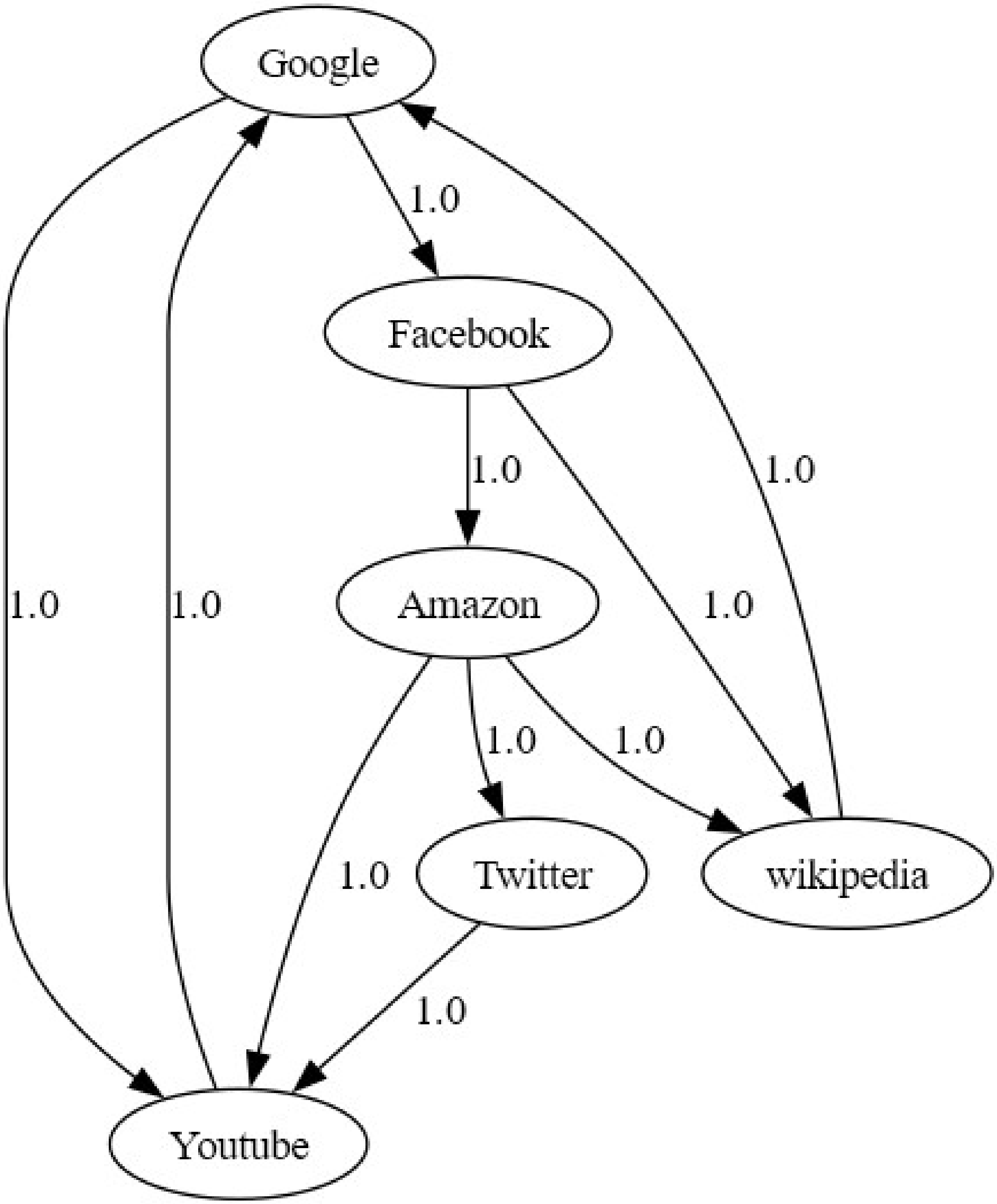
```
[1.         0.26339746 0.6        0.43660254]
[0.7402746  0.42830173 0.51290455 0.07402746]
```

```
U[:,0] / la.norm(U[:,0], 1)
```

```
array([0.42168675, 0.2439759 , 0.29216867, 0.04216867])
```

## Example 3

```
name_mapping={0:'Google', 1:'Facebook', 2:'Amazon', 3:'wikipedia',
4:'Twitter', 5:'Youtube'}

edges = [
    [0,1], [0,5],
    [1,2], [1,3],
    [2,3], [2,4], [2,5],
    [3,0],
    [4,5],
    [5,0]]

A = np.zeros((6,6))
for i,j in edges:
    A[j, i] = 1

graph_matrix(A, list(name_mapping.values()), show_weights=True)
```



```
A

array([[0., 0., 0., 1., 0., 1.],
       [1., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0.],
       [0., 1., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.],
       [1., 0., 1., 0., 1., 0.]])
```

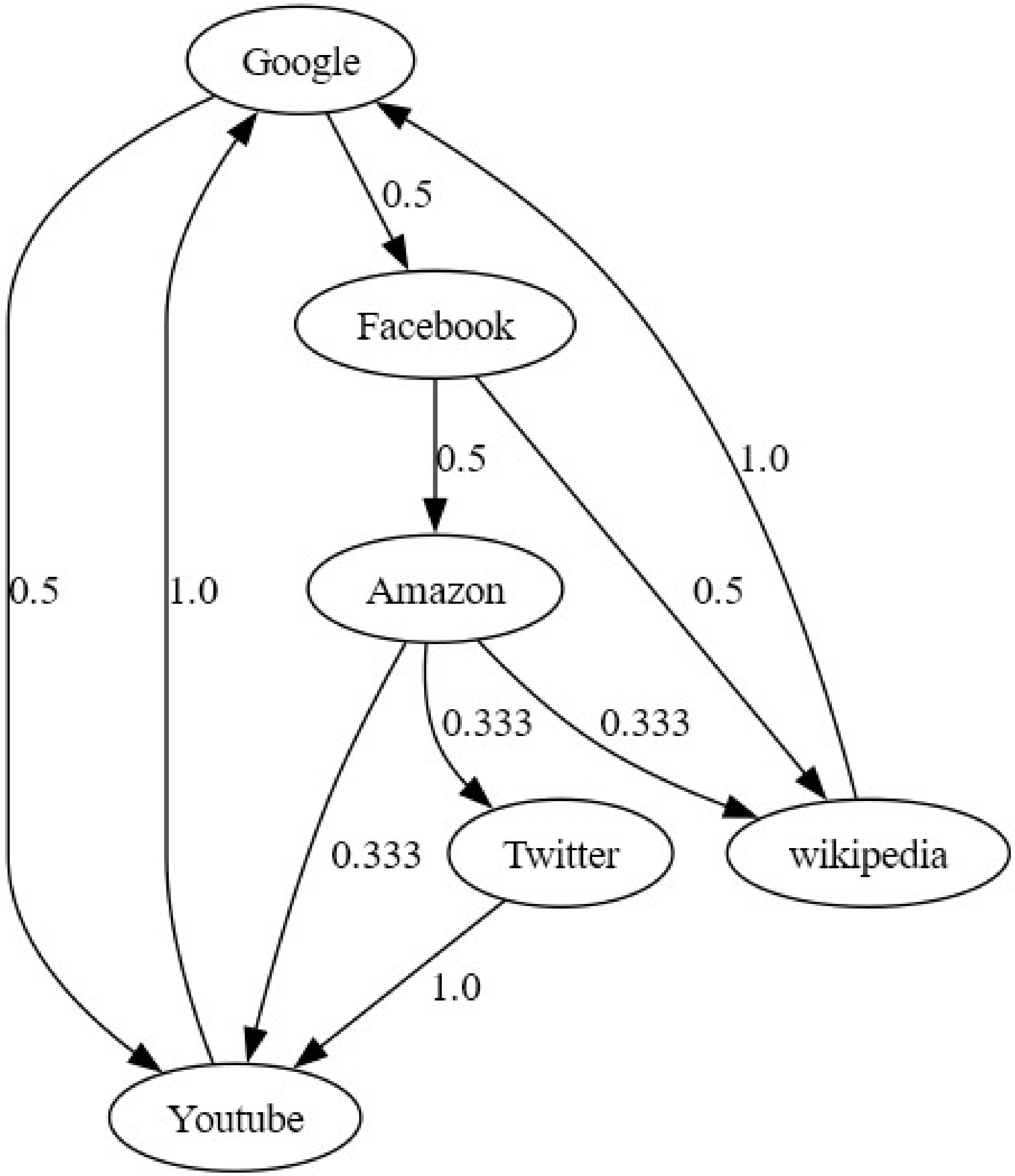
```
np.sum(A,axis=0)
```

```
array([2., 2., 3., 1., 1., 1.])
```

```
M = A/np.sum(A,axis=0)
M
```

```
array([[0.        , 0.        , 0.        , 1.        , 0.        ,
        1.        ],
       [0.5       , 0.        , 0.        , 0.        , 0.        ,
        0.        ],
       [0.        , 0.5       , 0.        , 0.        , 0.        ,
        0.        ],
       [0.        , 0.5       , 0.33333333, 0.        , 0.        ,
        0.        ],
       [0.        , 0.        , 0.33333333, 0.        , 0.        ,
        0.        ],
       [0.5       , 0.        , 0.33333333, 0.        , 1.        ,
        0.        ]])
```

```
graph_matrix(M, list(name_mapping.values()), show_weights=True)
```



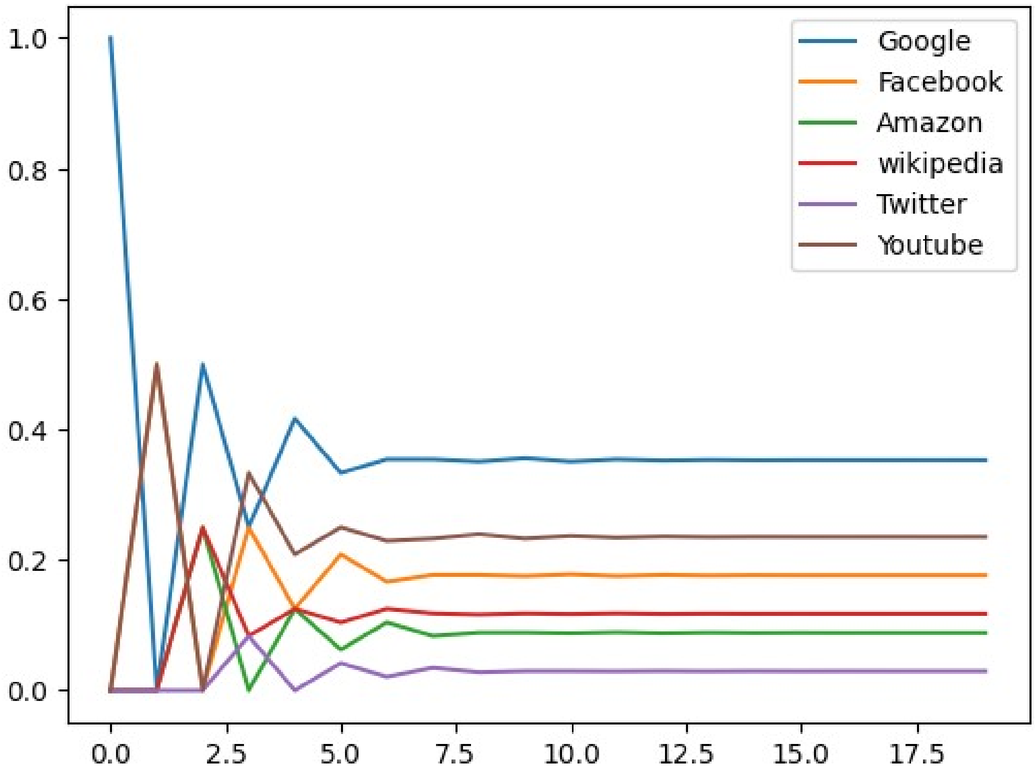
```
its = 20
allx = np.zeros((6,its))

allx[:,0] = np.array([1, 0, 0, 0, 0, 0])

for i in range(1,its):
    allx[:,i] = M@allx[:,i-1]
```

```
allx.T
```

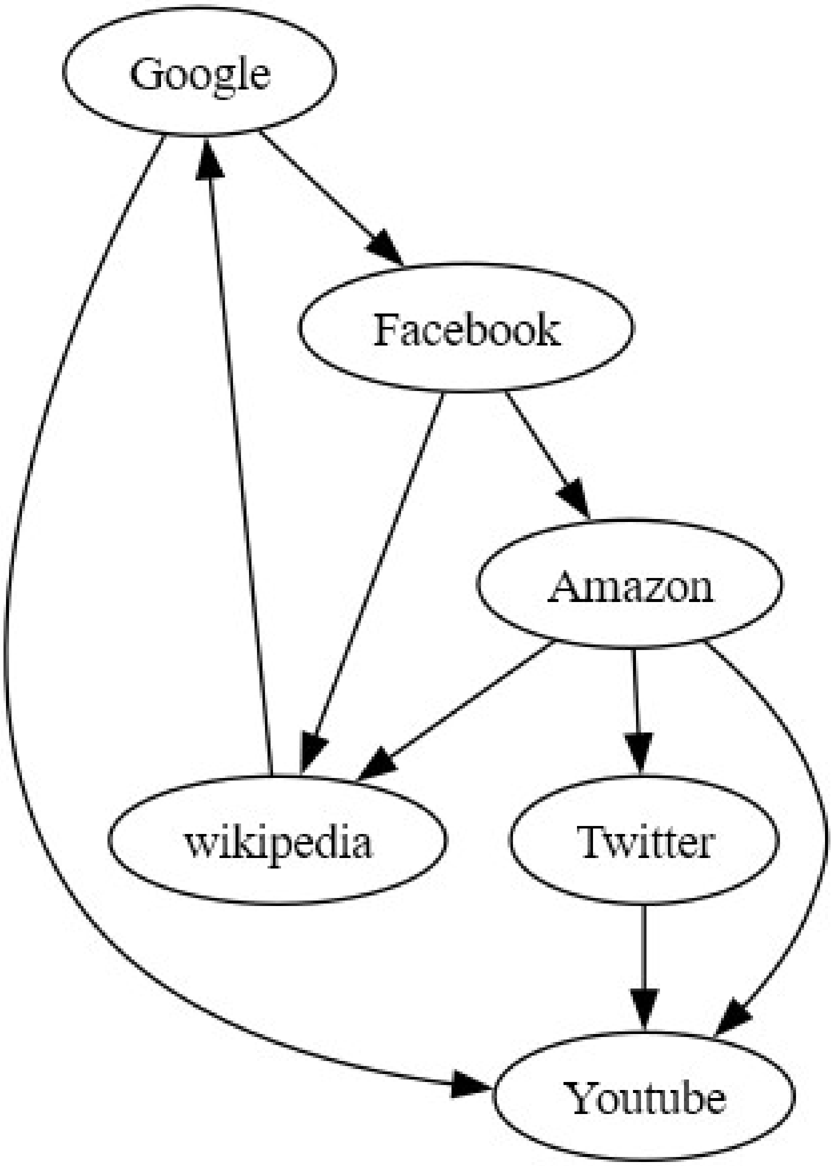
```
array([[1.        , 0.        , 0.        , 0.        , 0.        ,
        0.        ],
       [0.        , 0.5       , 0.        , 0.        , 0.        ,
        0.5       ],
       [0.5       , 0.        , 0.25      , 0.25      , 0.        ,
        0.        ],
       [0.25      , 0.25      , 0.        , 0.08333333, 0.08333333,
        0.33333333],
       [0.41666667, 0.125     , 0.125     , 0.125     , 0.        ,
        0.20833333],
       [0.33333333, 0.20833333, 0.0625    , 0.10416667, 0.04166667,
        0.25      ],
       [0.35416667, 0.16666667, 0.10416667, 0.125     , 0.02083333,
        0.22916667],
       [0.35416667, 0.17708333, 0.08333333, 0.11805556, 0.03472222,
        0.23263889],
       [0.35069444, 0.17708333, 0.08854167, 0.11631944, 0.02777778,
        0.23958333],
       [0.35590278, 0.17534722, 0.08854167, 0.11805556, 0.02951389,
        0.23263889],
       [0.35069444, 0.17795139, 0.08767361, 0.1171875 , 0.02951389,
        0.23697917],
       [0.35416667, 0.17534722, 0.08897569, 0.11820023, 0.02922454,
        0.23408565],
       [0.35228588, 0.17708333, 0.08767361, 0.11733218, 0.02965856,
        0.23596644],
       [0.35329861, 0.17614294, 0.08854167, 0.1177662 , 0.02922454,
        0.23502604],
       [0.35279225, 0.17664931, 0.08807147, 0.11758536, 0.02951389,
        0.23538773],
       [0.35297309, 0.17639612, 0.08832465, 0.11768181, 0.02935716,
        0.23526717],
       [0.35294898, 0.17648655, 0.08819806, 0.11763961, 0.02944155,
        0.23528525],
       [0.35292486, 0.17647449, 0.08824327, 0.11764263, 0.02939935,
        0.23531539],
       [0.35295802, 0.17646243, 0.08823724, 0.11765167, 0.02941442,
        0.23527621],
       [0.35292788, 0.17647901, 0.08823122, 0.11764363, 0.02941241,
        0.23530585]])
```

```
plt.plot(allx.T)
plt.legend(list(name_mapping.values()))
```

```
<matplotlib.legend.Legend at 0x1dbeda30770>
```



```
edges = [
    [0,1], [0,5],
    [1,2], [1,3],
    [2,3], [2,4], [2,5],
    [3,0],
    [4,5]]

A = np.zeros((6,6))
for i,j in edges:
    A[j, i] = 1

graph_matrix(A, list(name_mapping.values()), show_weights=False)
```



```
A
```

```
array([[0., 0., 0., 1., 0., 0.],
       [1., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0.],
       [0., 1., 1., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0.],
       [1., 0., 1., 0., 1., 0.]])
```

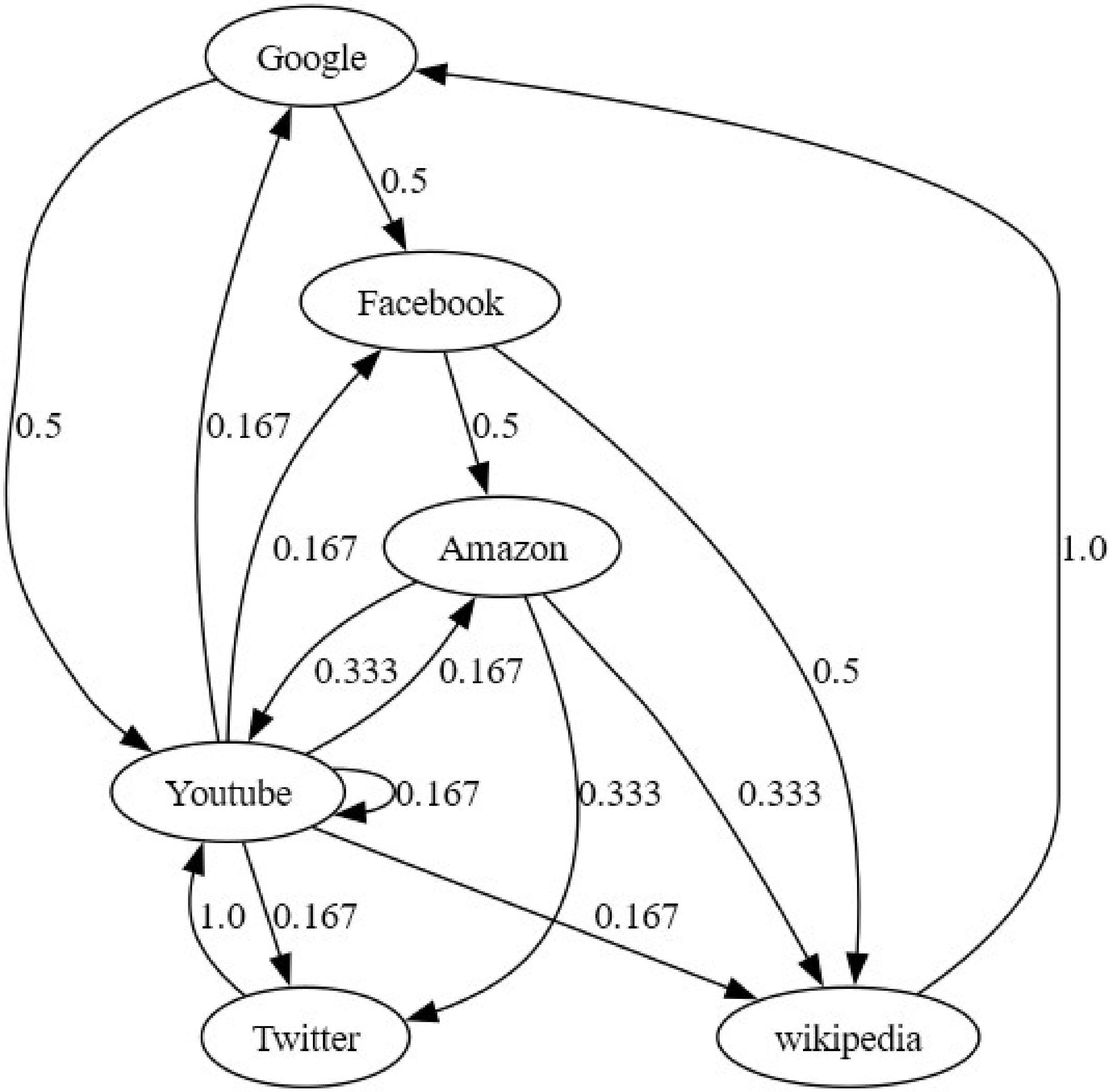
```
A[:, -1] = 1
A
```

```
array([[0., 0., 0., 1., 0., 1.],
       [1., 0., 0., 0., 0., 1.],
       [0., 1., 0., 0., 0., 1.],
       [0., 1., 1., 0., 0., 1.],
       [0., 0., 1., 0., 0., 1.],
       [1., 0., 1., 0., 1., 1.]])
```

```
M = A/np.sum(A,axis=0)
M
```

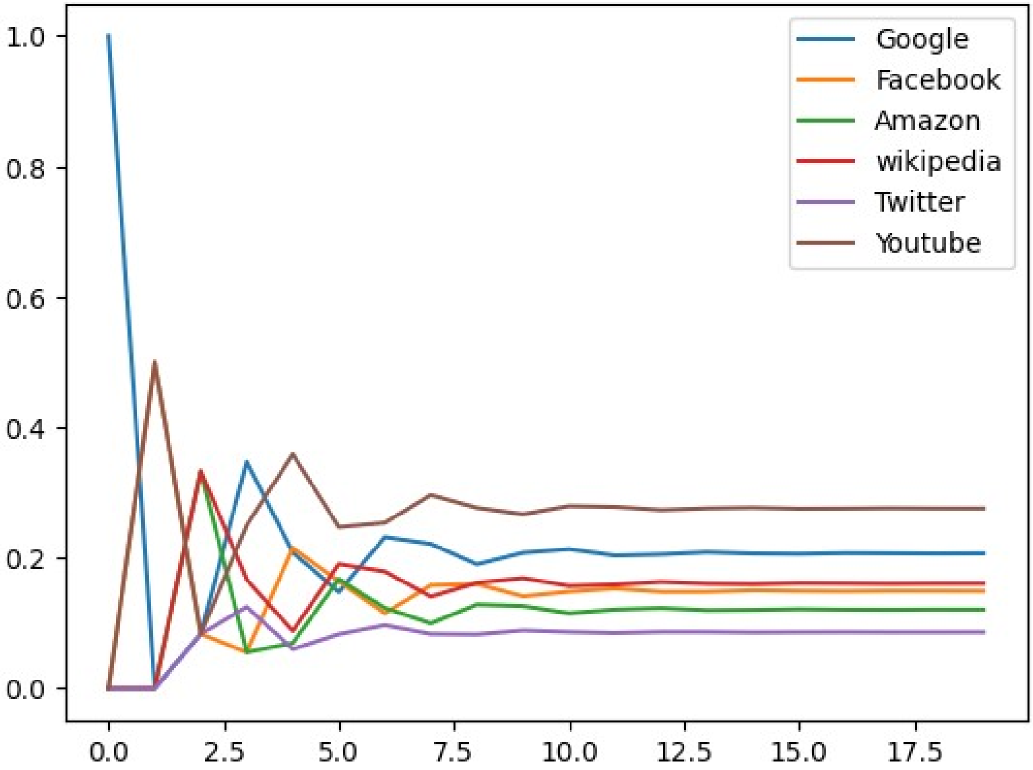
```
array([[0.        , 0.        , 0.        , 1.        , 0.        ,
        0.16666667],
       [0.5       , 0.        , 0.        , 0.        , 0.        ,
        0.16666667],
       [0.        , 0.5       , 0.        , 0.        , 0.        ,
        0.16666667],
       [0.        , 0.5       , 0.33333333, 0.        , 0.        ,
        0.16666667],
       [0.        , 0.        , 0.33333333, 0.        , 0.        ,
        0.16666667],
       [0.5       , 0.        , 0.33333333, 0.        , 1.        ,
        0.16666667]])
```

```
graph_matrix(M, list(name_mapping.values()), show_weights=True)
```

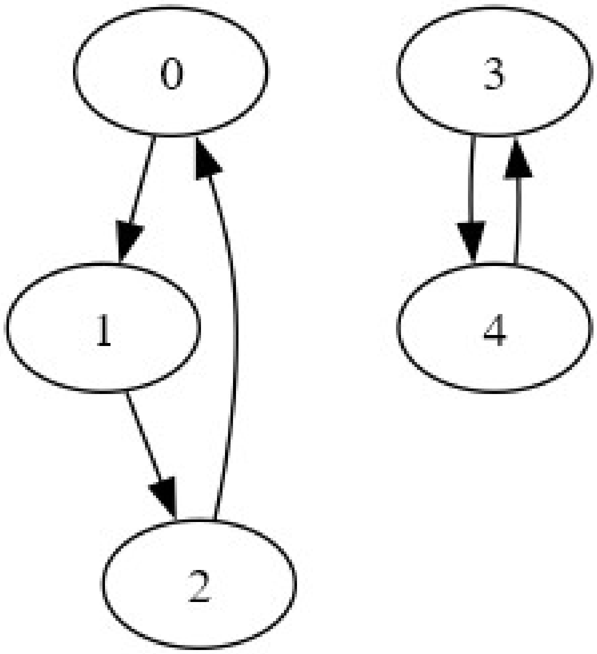


```
its = 20
allx = np.zeros((6,its))

allx[:,0] = np.array([1, 0, 0, 0, 0, 0])

for i in range(1,its):
    allx[:,i] = M@allx[:,i-1]

plt.plot(allx.T)
plt.legend(list(name_mapping.values()))
```

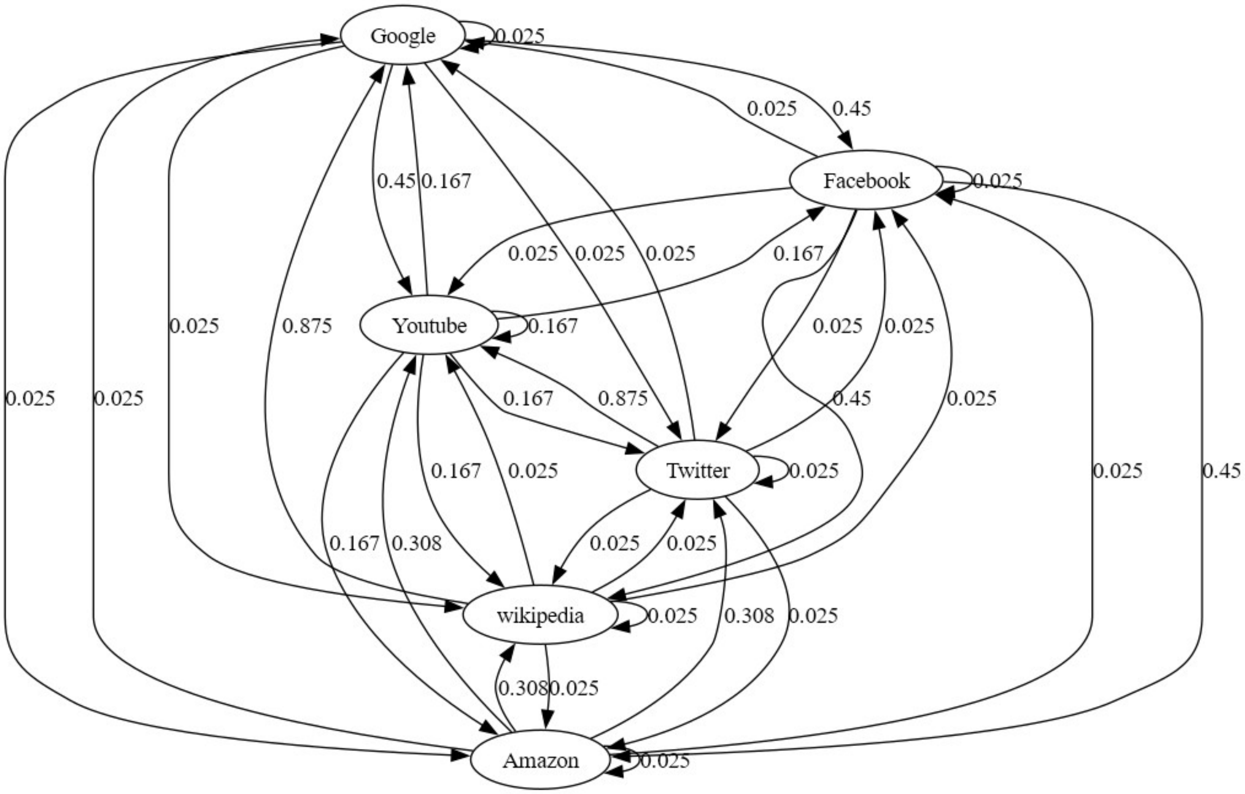
```
<matplotlib.legend.Legend at 0x1dbed8f3cb0>
```



```
B = np.array([[0,0,1,0,0],[1,0,0,0,0],[0,1,0,0,0],[0,0,0,0,1],
[0,0,0,1,0]])
graph_matrix(B, show_weights=False)
```

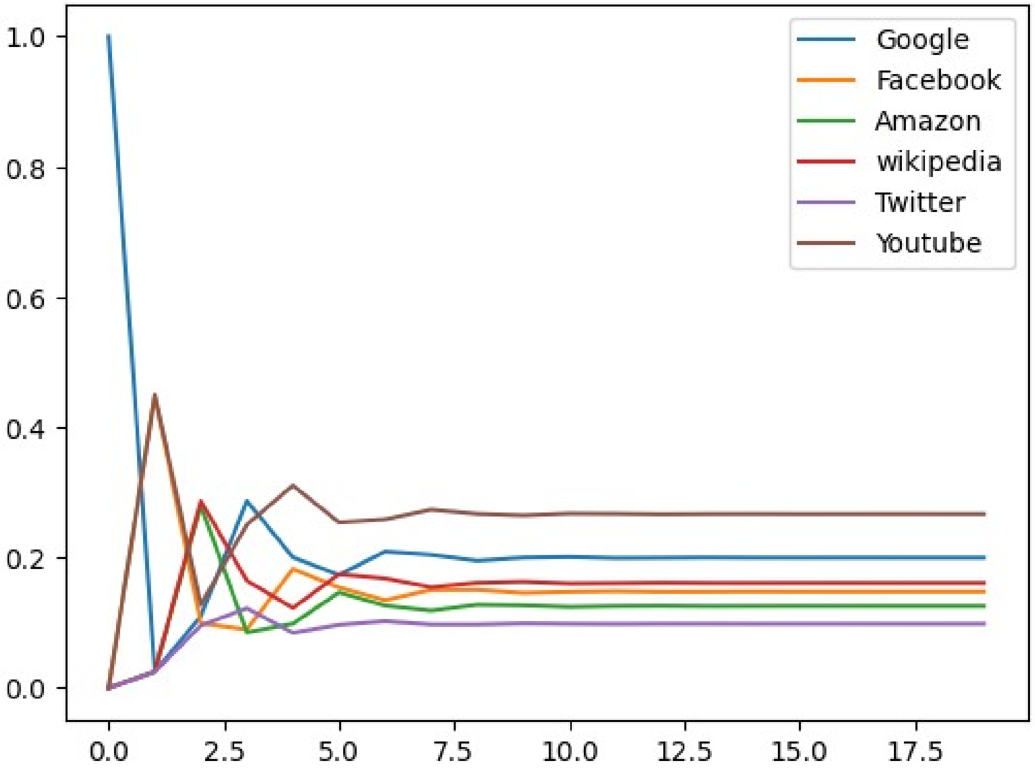


```
l,v = la.eig(B)
print('The eigenvalues are:')
print(l)

The eigenvalues are:
[-0.5+0.8660254j -0.5-0.8660254j  1. +0.j         1. +0.j
 -1. +0.j        ]

v[:,2]/la.norm(v[:,2],1)

array([-0.33333333+0.j, -0.33333333+0.j, -0.33333333+0.j,  0.
+0.j,
         0.        +0.j])

v[:,3]/la.norm(v[:,3],1)

array([0. +0.j, 0. +0.j, 0. +0.j, 0.5+0.j, 0.5+0.j])

BM = 0.85*M + 0.15/6*np.ones((6,6))
graph_matrix(BM, list(name_mapping.values()), show_weights=True)
```



```
its = 20
allx = np.zeros((6,its))

allx[:,0] = np.array([1, 0, 0, 0, 0, 0])
```

```
for i in range(1,its):
    allx[:,i] = BM@allx[:,i-1]

plt.plot(allx.T)
plt.legend(list(name_mapping.values()))

<matplotlib.legend.Legend at 0x1dbeda31b20>
```



```
allx.T

array([[1.        , 0.        , 0.        , 0.        , 0.        ,
         0.        ],
       [0.025     , 0.45      , 0.025     , 0.025     , 0.025     ,
         0.45      ],
       [0.11      , 0.099375  , 0.28      , 0.28708333, 0.09583333,
         0.12770833],
       [0.28711285, 0.08984201, 0.08532639, 0.16465972, 0.12242535,
         0.25063368],
       [0.2004672 , 0.1825294 , 0.09868929, 0.1228651 , 0.08468225,
         0.31076675],
       [0.17346063, 0.15422385, 0.14660028, 0.17456225, 0.09698726,
         0.25416573],
       [0.20938472, 0.13472758, 0.12655195, 0.1680887 , 0.10254356,
         0.25870349],
       [0.20452505, 0.15063817, 0.11890888, 0.15476527, 0.09750605,
        0.27365658],
       [0.19531849, 0.15069116, 0.12778924, 0.16148009, 0.09745887,
        0.26726215],
       [0.20012021, 0.1458725 , 0.12690588, 0.16311283, 0.09906909,
        0.26491948],
       [0.20117617, 0.14758135, 0.12452607, 0.16048274, 0.09848693,
        0.26774674],
       [0.19934112, 0.14843066, 0.12565286, 0.16093525, 0.09821318,
        0.26742694],
       [0.19968044, 0.14760546, 0.12596851, 0.16157016, 0.09848713,
        0.2666883 ],
       [0.20011548, 0.14764503, 0.12551316, 0.16120424, 0.09847192,
        0.26705017],
       [0.19985571, 0.14788118, 0.12558125, 0.16114331, 0.09839417,
        0.26714438],
       [0.19981727, 0.14778413, 0.12569496, 0.16127631, 0.09842681,
        0.26700053],
       [0.19990994, 0.14774741, 0.12563333, 0.1612469 , 0.09843865,
        0.26702377],
       [0.19988823, 0.14779009, 0.12562102, 0.16121713, 0.09842448,
        0.26705905],
       [0.19986792, 0.14778587, 0.12564415, 0.16123678, 0.09842599,
        0.26703929],
       [0.19988183, 0.14777443, 0.12563956, 0.16123874, 0.09842974,
        0.2670357 ]])
```