```
# Project 12: Matrix eigenvalues and the Google’s PageRank algorithm
import numpy as np
import matplotlib.pyplot as plt
import scipy.io
import networkx as nx
from scipy.sparse import csr_matrix
from matplotlib.pylab import eig


print("\nSubtask 1\n")
# Load the network data
# Load the adjacency matrix from the file 'AdjMatrix.mat'
data = scipy.io.loadmat("/content/AdjMatrix.mat")
AdjMatrix =csr_matrix(data['AdjMatrix'])
# Check the sparsity of the matrix
num_elements = AdjMatrix.shape[0] * AdjMatrix.shape[1]
num_non_zero_elements = AdjMatrix.nnz
nnzAdjMatrix = num_non_zero_elements / num_elements
print(f"Sparsity of AdjMatrix: {nnzAdjMatrix:.4f}")
```

```
Subtask 1

Sparsity of AdjMatrix: 0.0015
```

```
print("\nSubtask 2\n")
# Check the dimensions of the matrix
m, n = AdjMatrix.shape
print(f"Dimensions of AdjMatrix: {m} x {n}")
```

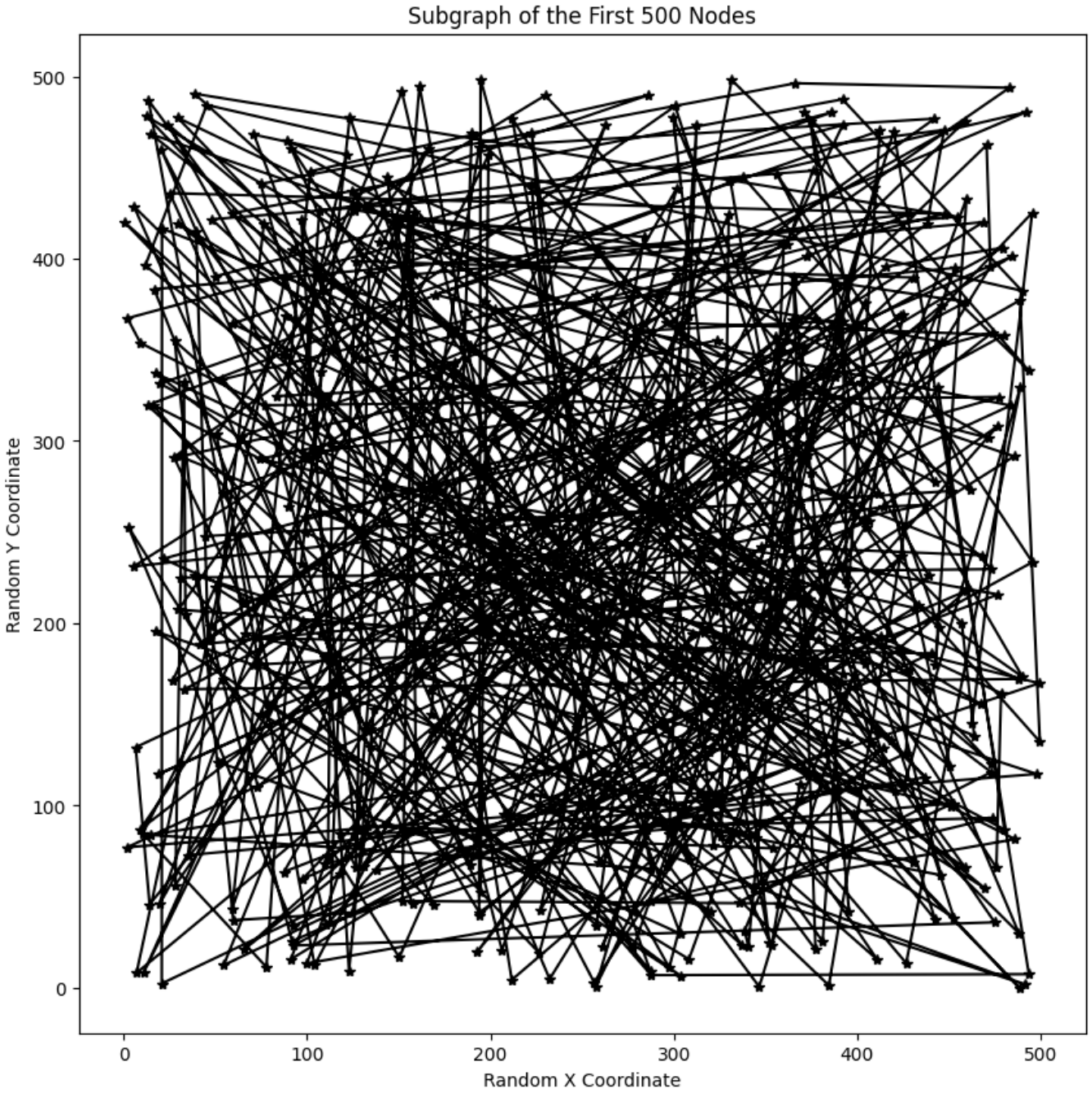
```
Subtask 2

Dimensions of AdjMatrix: 8297 x 8297
```

```
print("\nSubtask 3\n")
# Create a smaller submatrix and plot the network
NumNetwork = 500
AdjMatrixSmall = AdjMatrix[:NumNetwork, :NumNetwork].toarray() # Extract submatr
# Generate random coordinates for the nodes
#np.random.seed(0) # For reproducibility
coordinates = np.random.rand(NumNetwork, 2) * NumNetwork # Random coordinates
# Plot the graph
plt.figure(figsize=(10, 10))
plt.plot(coordinates[:, 0], coordinates[:, 1], 'k-*')
plt.title('Subgraph of the First 500 Nodes')
plt.xlabel('Random X Coordinate')
```

```
plt.ylabel('Random Y Coordinate')
plt.show()
# Variables
print(f"AdjMatrixSmall shape: {AdjMatrixSmall.shape}")
print(f"Coordinates shape: {coordinates.shape}")
print(f"NumNetwork: {NumNetwork}")
```

Subtask 3




```
AdjMatrixSmall shape: (500, 500)
Coordinates shape: (500, 2)
NumNetwork: 500
```

```
print("\nSubtask 4\n")
# Compute the Google Matrix
alpha = 0.15
GoogleMatrix = np.zeros((NumNetwork, NumNetwork))
# Check the amount of links originating from each webpage
NumLinks = np.sum(AdjMatrixSmall, axis=1)
for i in range(NumNetwork):
    if NumLinks[i] != 0:
        GoogleMatrix[i, :] = AdjMatrixSmall[i, :] / NumLinks[i]
    else:
        GoogleMatrix[i, :] = 1.0 / NumNetwork
GoogleMatrix = (1 - alpha) * GoogleMatrix + alpha * np.ones((NumNetwork,
NumNetwork)) / NumNetwork
# Compute the vectors w0, w1, w2, w3, w5, w10
w0 = np.ones(NumNetwork) / np.sqrt(NumNetwork)
w1 = w0 @ GoogleMatrix
w2 = w1 @ GoogleMatrix
w3 = w2 @ GoogleMatrix
w10 = w0 @ (GoogleMatrix ** 10)
w5 = w0 @ (GoogleMatrix ** 5)
deltaw = w10 - w5
print("Difference δw:", np.linalg.norm(deltaw))
```

```
Subtask 4

Difference δw: 0.27914257785554325
```

```
print("\nSubtask 5\n")
# Compute eigenvalues and eigenvectors
eigenvalues, right_eigenvectors = eig(GoogleMatrix)
# Find the index of the eigenvalue λ1 = 1
lambda_1_index = np.isclose(eigenvalues, 1)
# Get the right eigenvector corresponding to λ1
v1 = right_eigenvectors[:, lambda_1_index].flatten()
# Compute the left eigenvectors
left_eigenvalues, left_eigenvectors = eig(GoogleMatrix.T)
```

```
# Get the left eigenvector corresponding to λ1
u1 = left_eigenvectors[:, lambda_1_index].flatten()
print("Left Eigenvector (u1):", u1)
```

```
Left Eigenvector (u1): [0.04056307+0.j 0.02834097+0.j 0.08716923+0.j 0.00812904+0.j
 0.05257316+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.31456166+0.j 0.00812904+0.j
 0.03223344+0.j 0.03121822+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.04185401+0.j
 0.00812904+0.j 0.15345631+0.j 0.00812904+0.j 0.00812904+0.j
 0.02101443+0.j 0.05566927+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.04552237+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.03166808+0.j 0.25299032+0.j 0.038545  +0.j
 0.02080848+0.j 0.05265016+0.j 0.00812904+0.j 0.04523961+0.j
 0.00812904+0.j 0.02221731+0.j 0.00812904+0.j 0.0553448 +0.j
 0.00812904+0.j 0.06817385+0.j 0.03014009+0.j 0.00812904+0.j
 0.05189484+0.j 0.00841694+0.j 0.00812904+0.j 0.00812904+0.j
 0.04231846+0.j 0.00812904+0.j 0.03176089+0.j 0.00812904+0.j
 0.04441903+0.j 0.09339229+0.j 0.00812904+0.j 0.01631514+0.j
 0.06110101+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.02198936+0.j
 0.00812904+0.j 0.01548592+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.01466011+0.j 0.03454352+0.j 0.00812904+0.j
 0.01711427+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.01158557+0.j 0.00812904+0.j 0.04585962+0.j
 0.02638632+0.j 0.02185328+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.09702018+0.j
 0.00812904+0.j 0.02110607+0.j 0.01753219+0.j 0.00812904+0.j
 0.00812904+0.j 0.19978894+0.j 0.19614392+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.03980244+0.j
 0.00812904+0.j 0.11064736+0.j 0.0349661 +0.j 0.02656457+0.j
 0.04330746+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.03940024+0.j 0.01675149+0.j 0.0283924 +0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.02079878+0.j 0.01595535+0.j 0.05589886+0.j 0.01353273+0.j
 0.00812904+0.j 0.04648402+0.j 0.00812904+0.j 0.03937498+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.15207168+0.j 0.02145004+0.j 0.00812904+0.j
 0.08729022+0.j 0.03267737+0.j 0.08427301+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.0868351 +0.j 0.00812904+0.j
 0.02711911+0.j 0.00812904+0.j 0.01194664+0.j 0.00812904+0.j
 0.01294127+0.j 0.0217217 +0.j 0.00812904+0.j 0.00812904+0.j
 0.01668783+0.j 0.00812904+0.j 0.02841732+0.j 0.06915248+0.j
 0.03674081+0.j 0.00812904+0.j 0.02836281+0.j 0.05362505+0.j
 0.02385953+0.j 0.00812904+0.j 0.04731023+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.09484375+0.j 0.0275419 +0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.02948802+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j]
```

```
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.01717798+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.05014144+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.03207286+0.j
 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j 0.00812904+0.j
```

```
print("\nSubtask 6\n")
# Normalize u1 to have all positive components
u1 = np.abs(u1) / np.linalg.norm(u1, 1)
```

```
Subtask 6
```

```
print("\nSubtask 7\n")
MaxRank, PageMaxRank = np.max(u1), np.argmax(u1)
print(f"MaxRank: {MaxRank}, PageMaxRank: {PageMaxRank}")
```

```
Subtask 7

MaxRank: 0.033305110375255596, PageMaxRank: 27
```

```
print("\nSubtask 8\n")
MostLinks = np.sum(AdjMatrixSmall, axis=0) # Sum of columns
MaxLinks, PageMaxLinks = np.max(MostLinks), np.argmax(MostLinks)
print(f"MostLinks: {MostLinks}, MaxLinks: {MaxLinks}, PageMaxLinks: {PageMaxLir
```

```
Subtask 8

MostLinks: [  0.   0.  31.   0.   0.  20.   0.  40.   0.  15.   0.   0.   0
  35.   0.   0.   0.  22.   0.   0.   0.  22.   0.   0.   0.   0. 122.
  35.  23.   0.  14.  34.  20.  43.  24.   0.  14.  23.   0.   0.   0.
   0.   0.   0.   0.   0.   0.  19.  32.   0.   0.   0.  40.  56.  65.
   0.   0.   0.   0.  20.   0.   0.   1.   0.   0.   0.   0.   0.   0.
   0.  16.   0.   0.  64.   0.   0.   0.   0.  43.   0.   0.   0.   0.
   0.  55.   0.   0.  16.   0.   0.   0.  17.  27.  31.   0.   0.   0.
   0.   0.   0.   0.   0.   0.  21.   0.   0.  11.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
  19.   0.   0.  20.   0.  28.   0.   0.  35.   0.   0.   0.   0.  22.
   0.   0.   0.  29.   0.   0.  15.   0.   0.  20.  32.  44.  18.  18.
   1.   0.   0.   0.   0.   0.   0.   0.  14.   0.   0.   0.  12.  24.
   0.   0.  10.   0.   0.   0.  16.   0.   0.  35.   0.   0.   0.  31.
  27.   0.   0.   0.   0.   0.   0.   0.   0.  27.   0.   0.   0.   0.
   0.   0.   0.   0.   0.   0.   0.  28.   0.   0.   0.   0.   0.   0.
   0.   0.   0.  94.   0.  21.   0.   0.  18.   0.   0.   0.   0.  16.
   0.  21.  10.  36.   0.   7.   0.  22.   0.   0.   0.  20.   0.   0.
   0.   0.   0.   0.  52.   0.   0.  17.   0.   0.   0.  22.   0.  38.
   0.   0.  21.   0.  25.  13.  32.   0.  24.   0.   0.   0.   0.   0.
   0.   0.   0.   0. 122.   0.  13.  29.   0.   0.   0.   0.   0.  19.
   0.  70.   0.   0.   8.  19.   0.   0.   0.  20.   0.   0.   0.   0.
   0.   0.   0.  15. 108.  22.  13.  24.   0.  25.   0.  18.   0.  29.
   0.  34.  15.   0.  26.   1.   0.   0.  26.   0.  15.   0.  21.  41.
   0.   7.  39.   0.   0.   0.   0.   0.   0.   9.   0.  13.   0.   0.
   0.   7.  20.   0.  13.   0.   0.   0.   0.   3.   0.  35.  15.   8.
   0.   0.   0.   0.   0.  43.   0.  12.   7.   0.   0.  11.  18.   0.
   0.   0.   0.  15.   0.  45.  21.  11.  20.   0.   0.   0.  28.  17.
  21.   0.   0.   0.   0.   0.   0.   0.   0.   0.  10.  13.  33.   5.
   0.  19.   0.  24.   0.   0.   0.   0.   0.  50.   6.   0.  21.  14.
  45.   0.   0.   0.   0.   0.   0.   0.  36.   0.  15.   0.   3.   0.
  10.  16.   0.   0.   4.   0.  22.  22.  15.   0.  21.  24.  12.   0.
  18.   0.   0.   0.  43.  15.   0.   0.   0.   0.   0.   0.  18.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.  14.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
  23.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.  18.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.], MaxLinks: 122.0, PageM
```

```
print("\nSubtask 9\n")
are_equal = PageMaxRank == PageMaxLinks
print(f"Is the highest ranking webpage the same as the page with the most hyperl
# Q1: What is the number of hyperlinks pointing to the webpage MaxRank?
print(f"Number of hyperlinks pointing to the webpage MaxRank:{MostLinks[PageMaxR
```

```
Subtask 9

Is the highest ranking webpage the same as the page with the most hyperlink
Number of hyperlinks pointing to the webpage MaxRank:122.0
```