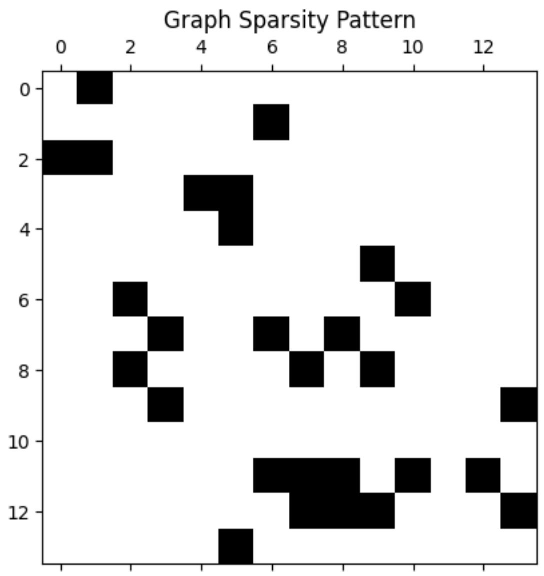
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.sparse import csr_matrix


def PageRank(G, alpha):
    it = 0
    delta = 1
    tol = 1e-8

    R = G.shape[0]
    p_k_1 = p_0 = np.ones(R) / R

    d = (np.sum(G, axis=0) == 0).astype(int)

    P = (G + (1 / R) * d.transpose())
    P_prime = P / np.sum(P, axis=0)

    M = alpha * P_prime + (1 - alpha) * (1 / R)

    while delta > tol:
        p_k = M @ p_k_1
        delta = np.linalg.norm(p_k - p_k_1, 1)
        p_k_1 = p_k
        it += 1

    return p_k, it


def PageRankSparse(G, alpha):
    it = 0
    delta = 1
    tol = 1e-8

    R = G.shape[0]
    p_k_1 = p_0 = np.ones(R) / R

    d = (np.array(G.sum(axis=0)).flatten() == 0).astype(int)

    P = G + d
    P_prime = P / P.sum(axis=0)

    while delta > tol:
        p_k = alpha * np.array(P_prime.dot(p_k_1)).flatten() + (1 - alpha) * (1 / R)
        delta = np.linalg.norm(p_k - p_k_1, 1)
        p_k_1 = p_k
        it += 1

    return p_k, it


G = np.array([[0, 1, 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],
              [1, 1, 0, 0, 0, 0, 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, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0],
              [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0],
              [0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
              [0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0],
              [0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0],
              [0, 0, 0, 1, 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, 0, 1, 1, 1, 0, 1, 0, 1, 0],
              [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1],
              [0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]])
alpha = 0.9


(p, it) = PageRank(G, alpha)
print(p, it)
```

```
[0.03885066 0.03993066 0.07381625 0.06248823 0.03288854 0.04002228
 0.06349601 0.09892554 0.10291725 0.06380141 0.02088186 0.2137178
 0.11537496 0.03288854] 24
```

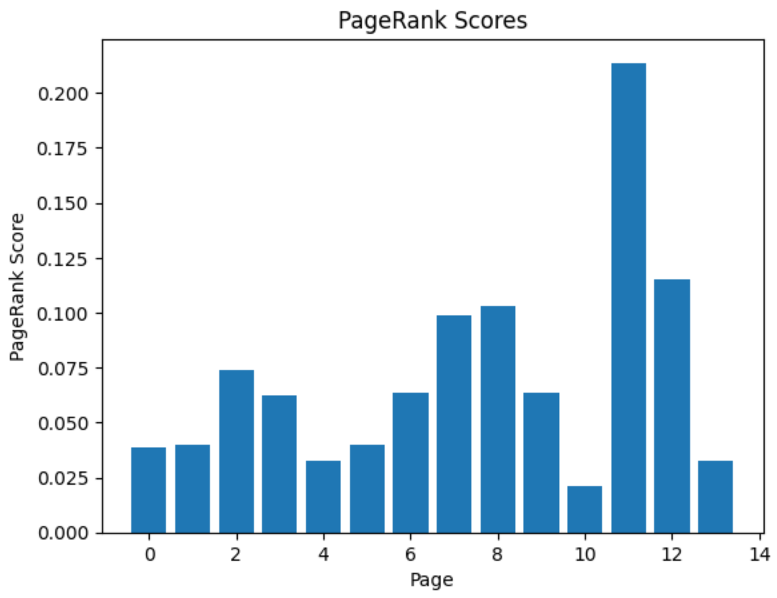
```
plt.spy(G)
plt.title("Graph Sparsity Pattern")
```

```
Text(0.5, 1.0, 'Graph Sparsity Pattern')
```



```
plt.bar(range(len(p)), p)
plt.title("PageRank Scores")
plt.xlabel("Page")
plt.ylabel("PageRank Score")
```

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

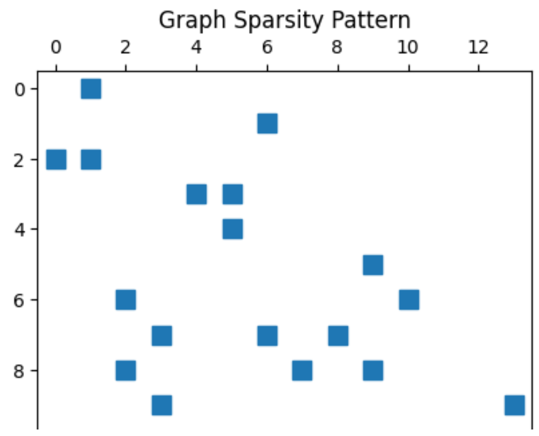


```
(p, it) = PageRankSparse(csr_matrix(G), alpha)
print(p, it)
```

```
[0.03885066 0.03993066 0.07381625 0.06248823 0.03288854 0.04002228
 0.06349601 0.09892554 0.10291725 0.06380141 0.02088186 0.2137178
 0.11537496 0.03288854] 24
```

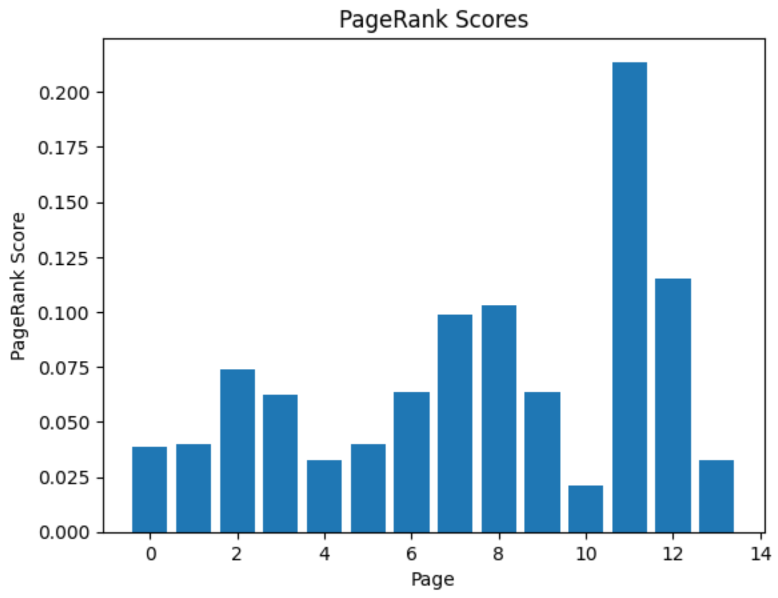
```
plt.spy(csr_matrix(G))
plt.title("Graph Sparsity Pattern")
```

```
Text(0.5, 1.0, 'Graph Sparsity Pattern')
```



```
plt.bar(range(len(p)), p)
plt.title("PageRank Scores")
plt.xlabel("Page")
plt.ylabel("PageRank Score")
```

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

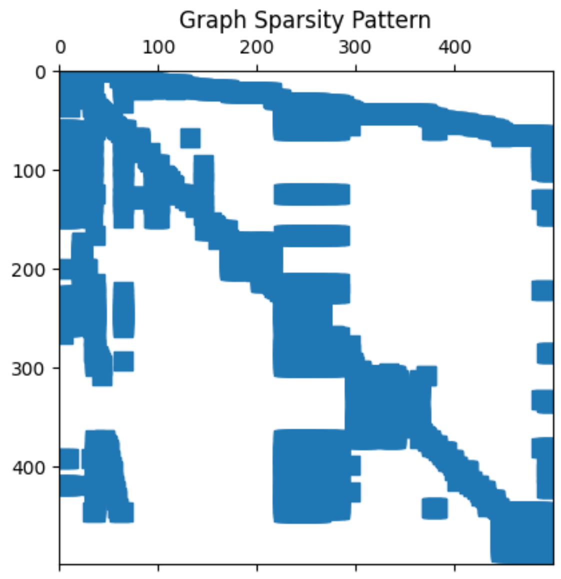


```
import scipy.io


data = scipy.io.loadmat('bbc.mat')


Gcsr = data['G']
Gcsr = Gcsr.transpose() #data uses the reverse adjacency matrix convention.
U = data['U']


(p, it) = PageRankSparse(Gcsr, alpha)
print(p, it)
```

```
 0.00067098 0.00067098 0.00068408 0.01788185 0.00232851 0.0006804
 0.00067098 0.0002     0.0006804  0.0002     0.0002     0.0006804
 0.00229142 0.0006804  0.0006804  0.0002     0.0002     0.0002
 0.0002     0.0002     0.0002     0.00020367 0.00236483 0.0052556
 0.0002     0.0002     0.0002     0.0002     0.0002     0.0002
 0.0002     0.00103441 0.00093645 0.00072575 0.00072575 0.00072575
 0.00072575 0.00072575 0.00184127 0.00467891 0.00028991 0.00103441
 0.00106866 0.00289893 0.00106866 0.00289893 0.00114862 0.00318235
 0.00075917 0.00161114 0.00388525 0.00093645 0.00106866 0.00289893
 0.0002     0.0002     0.0002     0.00020664 0.00020664 0.00020664
 0.00021031 0.0002     0.01006308 0.00060116 0.00030392 0.00029964
 0.00029964 0.00031634 0.00031634 0.00031634 0.00046544 0.00046544
 0.00046544 0.00056931 0.00046544 0.00046544 0.00046544 0.00039733
 0.00046544 0.00054519 0.00046544 0.0004204  0.00037216 0.00038663
 0.00115687 0.00046419 0.0002     0.00022385 0.0002     0.00020339
 0.00029492 0.00020339 0.00021384 0.0002     0.0002     0.0002
 0.0002     0.0002     0.00059238 0.00057516 0.00059238 0.00057516
 0.00059238 0.00059238 0.00057516 0.00059238 0.00059238 0.00057516
 0.00045191 0.00080977 0.0005448  0.00059238 0.00078941 0.00043418
 0.00079909 0.00045191 0.00055259 0.0002     0.0002     0.0002
 0.0005448  0.0005448  0.0005448  0.0005448  0.0005448  0.00042805
 0.00056275 0.00136982 0.00185796 0.00433401 0.0002     0.00118718
 0.00031872 0.00169117 0.00403076 0.00033269 0.00025267 0.00106866
 0.00289893 0.00428458 0.00874605 0.00088506 0.00114201 0.00093645
 0.0008921  0.00093645 0.00093645 0.00040706 0.00359294 0.00748853
 0.00069138 0.00093645 0.00093645 0.00102008 0.00170246 0.00416989
 0.00028991 0.00093645 0.00497874 0.00381034 0.00804193 0.00028991
 0.0002     0.00093645 0.0002     0.00025806 0.00084314 0.00113228
 0.00110674 0.00136266 0.00355206 0.0002     0.0043569  0.00229142
 0.00406087 0.00857651 0.00028991 0.00093645 0.00093645 0.00380288
 0.00800826 0.00085382 0.0015088  0.00110171 0.00104125 0.00085382
 0.00178077 0.00073423 0.0043363  0.00884009 0.0002     0.00193493
 0.00149957 0.00122965 0.0010452  0.00091082 0.00238812 0.00529794
 0.00093645 0.0002     0.00441948 0.00177791 0.00041416 0.0002
 0.00175262 0.00175262 0.00175262 0.00175262 0.00175262 0.00175262
 0.00175262 0.00175262 0.00175262 0.00175262 0.00175262 0.00176493
 0.00175262 0.00176493 0.00176493 0.00176493 0.00176493 0.00176493
 0.00176493 0.00175262 0.00175262 0.00175262 0.00175262 0.00176493
 0.00176493 0.00176493 0.00175262 0.00176493 0.00176493 0.00175262
 0.00176493 0.00175262 0.00176493 0.00176493 0.00175262 0.00175262
 0.00175262 0.00175262 0.00021637 0.00022018 0.00023249 0.00022018
 0.00020339 0.0002     0.0002     0.0002     0.0002     0.000212
 0.0002     0.0002    ] 109
```

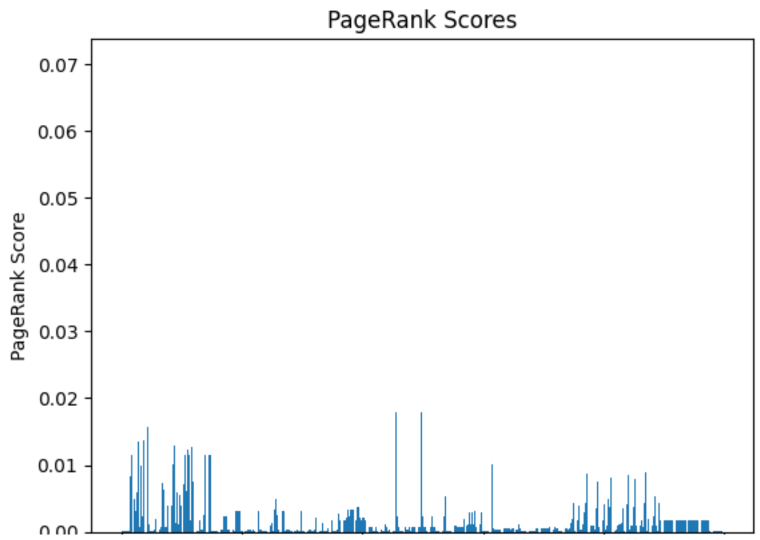
```
plt.spy(Gcsr)
plt.title("Graph Sparsity Pattern")
```

```
Text(0.5, 1.0, 'Graph Sparsity Pattern')
```



```
plt.bar(range(len(p)), p)
plt.title("PageRank Scores")
plt.xlabel("Page")
plt.ylabel("PageRank Score")
```

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



```
[val[0][0] for val in U[np.argsort(p)[-20:]]]
```

```
['https://www.bbc.com/news/world-europe-45902014',
 'https://www.bbc.com/news/world-latin-america-45944164',
 'http://bbc.in/2rAX810',
 'http://www.bbc.com/travel/columns/culinary-roots',
 'http://www.bbc.com/travel/columns/welcome-to-our-house',
 'http://www.bbc.com/travel/columns/to-the-ends-of-the-earth',
 'http://www.bbc.com/travel',
 'http://bbc.in/2s4bidh',
 'http://www.bbc.com/travel/columns/travel-journeys',
 'https://www.bbc.com/news/business-45961761',
 'https://www.bbc.com/news/uk-politics-45948282',
 'https://www.bbc.com/news/education-45979234',
 'https://www.bbc.com/news/business-45939984',
 'https://www.bbcgoodfood.com',
 'http://www.bbc.com/future/story/20181026-how-one-sided-objects-like-a-mobius-strip-work',
 'http://www.bbc.com/future/story/20181024-the-best-age-to-learn-a-foreign-language',
 'http://www.bbcamerica.com',
 'https://www.bbc.co.uk/news',
 'https://www.bbc.com/news',
 'http://www.bbc.com']
```