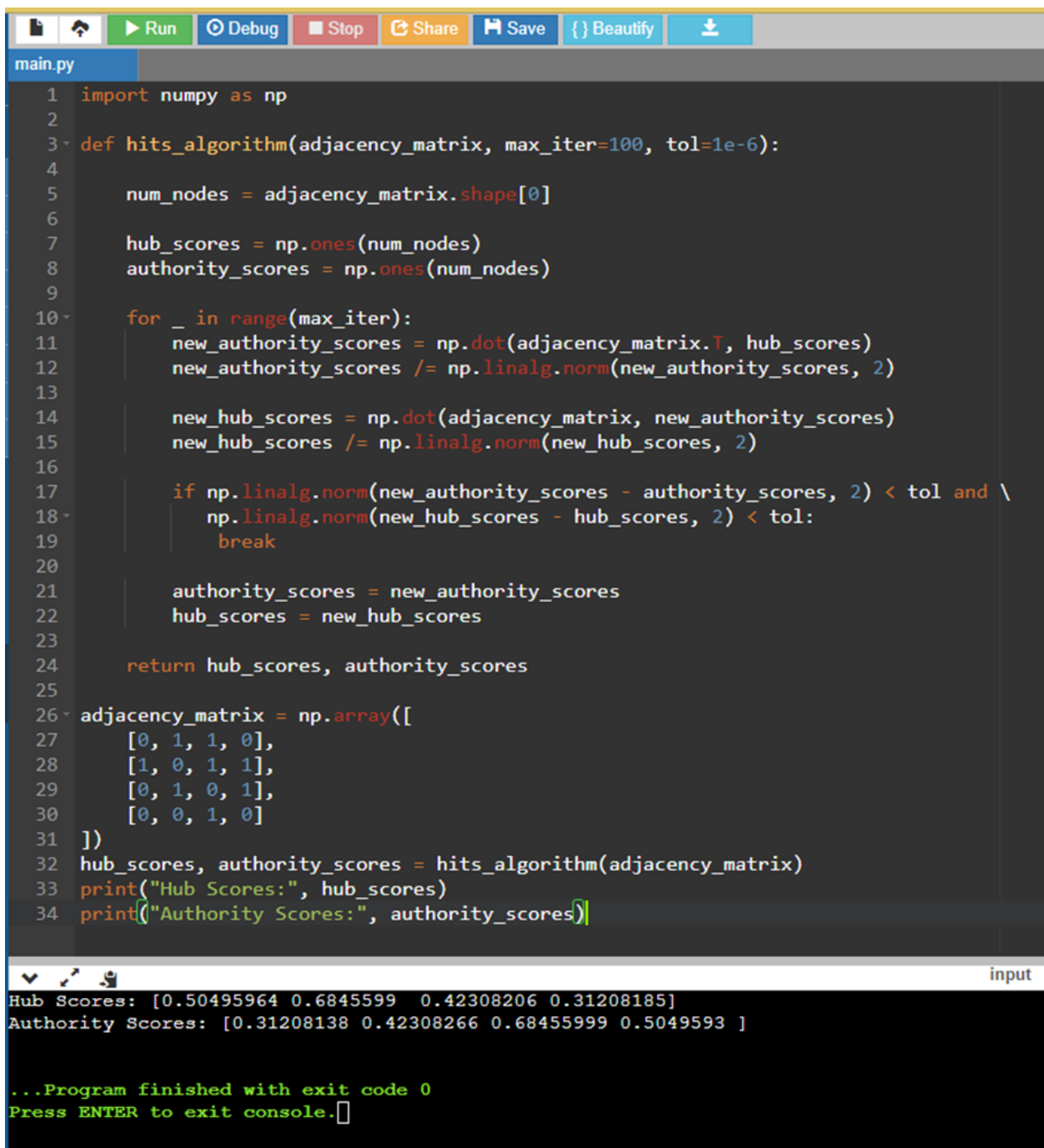


## Experiment 9



The image shows a Python IDE window with a toolbar at the top containing icons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. The file name 'main.py' is displayed in the top-left corner of the editor. The code defines a function 'hits\_algorithm' that takes an adjacency matrix and iterates to calculate hub and authority scores using NumPy and SciPy. It then applies this function to a specific 4x4 adjacency matrix and prints the results. The output at the bottom shows the calculated scores for each of the four nodes.

```
1 import numpy as np
2
3 def hits_algorithm(adjacency_matrix, max_iter=100, tol=1e-6):
4
5     num_nodes = adjacency_matrix.shape[0]
6
7     hub_scores = np.ones(num_nodes)
8     authority_scores = np.ones(num_nodes)
9
10    for _ in range(max_iter):
11        new_authority_scores = np.dot(adjacency_matrix.T, hub_scores)
12        new_authority_scores /= np.linalg.norm(new_authority_scores, 2)
13
14        new_hub_scores = np.dot(adjacency_matrix, new_authority_scores)
15        new_hub_scores /= np.linalg.norm(new_hub_scores, 2)
16
17        if np.linalg.norm(new_authority_scores - authority_scores, 2) < tol and \
18            np.linalg.norm(new_hub_scores - hub_scores, 2) < tol:
19            break
20
21        authority_scores = new_authority_scores
22        hub_scores = new_hub_scores
23
24    return hub_scores, authority_scores
25
26 adjacency_matrix = np.array([
27     [0, 1, 1, 0],
28     [1, 0, 1, 1],
29     [0, 1, 0, 1],
30     [0, 0, 1, 0]
31 ])
32 hub_scores, authority_scores = hits_algorithm(adjacency_matrix)
33 print("Hub Scores:", hub_scores)
34 print("Authority Scores:", authority_scores)
```

input

```
Hub Scores: [0.50495964 0.6845599  0.42308206 0.31208185]
Authority Scores: [0.31208138 0.42308266 0.68455999 0.5049593 ]

...Program finished with exit code 0
Press ENTER to exit console.
```