## **Experiment 8**

```
▶ IDLE Shell 3.9.13
                                                                                                                                                          RA.py - C:/Users/djsce.student/Desktop/PRA.py (3.9.13)
 File Edit Shell Debug Options Window Help
                                                                                                                                                         File Edit Format Run Options Window Help
   pageranks = np.linalg.solve(np.array(coefficients_list),np.array(constant_matrix))
                                                                                                                                                         def page_rank_algorithm(graph,damping_factor):
    outgoing = dict()
    incoming_nodes = dict()
    coefficients = dict()
    for i,rank in enumerate(pageranks):
print('Page Rank of {} is {:.4f}'.format(chr(65+i), rank))
                                                                                                                                                            # Outgoing Nodes
for i in range(len(graph)):
   n = int(input('Enter the number of nodes : '))
                                                                                                                                                                outgoing[i]=0
   d= float(input('Enter the damping factor : '))
                                                                                                                                                            for i,node in enumerate(graph):
                                                                                                                                                                for edge in node:
                                                                                                                                                                  if edge:
    print(Enter Adjacency Matrix with terms separated by a space : ')
                                                                                                                                                                      outgoing[i] += 1
    for i in range(n):

temp_list = input().split(' ')

graph.append(list(map(int,temp_list)))
                                                                                                                                                             # Incoming Nodes
                                                                                                                                                             for i in range(len(graph)):
                                                                                                                                                                temp=[]
for node in graph:
   page_rank_algorithm(graph,d)
                                                                                                                                                                  if node[i]:
                                                                                                                                                               temp.append(node)
incoming_nodes[i] = temp
main()
SyntaxError: multiple statements found while compiling a single statement
                                                                                                                                                            # Coefficient Matrix
                                                                                                                                                             for i,node in enumerate(graph):
Enter the number of nodes: 4
Enter the damping factor: 0.5
Enter Adjacency Matrix with terms separated by a space:
                                                                                                                                                                temp = \prod
                                                                                                                                                                temp = []
for j, other_node in enumerate(graph):
    if other_node in incoming_nodes[i]:
        temp.append(damping_factor*(1.0/outgoing[j]))
    elif i == j:
0010
                                                                                                                                                                      temp.append(-1)
0010
                                                                                                                                                                temp.append(0)
coefficients[i] = temp
Page Rank of A is 1.2308
Page Rank of B is 0.8077
Page Rank of C is 1.4615
Page Rank of D is 0.5000
                                                                                                                                                            coefficients_list = []
for key,value in coefficients.items():
                                                                                                                                                                coefficients_list.append(value)
                                                                                                                                                             constant_matrix = []
Enter the number of nodes : 3
Enter the damping factor : 0.5
Enter Adjacency Matrix with terms separated by a space :
                                                                                                                                                             for i in range(len(graph)):
                                                                                                                                                                constant_matrix.append(damping_factor-1)
                                                                                                                                                            pageranks = np.linalg.solve(np.array(coefficients_list),np.array(constant_matrix))
\begin{smallmatrix}0&0&1\\1&0&0\end{smallmatrix}
                                                                                                                                                            print()
                                                                                                                                                             for i,rank in enumerate(pageranks):
print('Page Rank of {} is {:.4f}'.format(chr(65+i), rank))
Page Rank of A is 1.0769
Page Rank of B is 0.7692
Page Rank of C is 1.1538
                                                                                                                                                          def main();
n = int(innut('Enter the number of nodes : '))
```

```
IDLE Shell 3.9.13
                                                                                                                                              RA.py - C:/Users/djsce.student/Desktop/PRA.py (3.9.13)
File Edit Shell Debug Options Window Help
                                                                                                                                              File Edit Format Run Options Window Help
   pageranks = np.linalg.solve(np.array(coefficients_list),np.array(constant_matrix))
                                                                                                                                                 # Incoming Nodes
                                                                                                                                                 for i in range(len(graph)):
   for i,rank in enumerate(pageranks):
print('Page Rank of {} is {:.4f}'.format(chr(65+i), rank))
                                                                                                                                                   temp=[]
for node in graph:
if node[i]:
                                                                                                                                                    temp.append(node)
incoming_nodes[i] = temp
 def main():
    n = int(input('Enter the number of nodes : '))
   d= float(input('Enter the damping factor : '))
                                                                                                                                                 # Coefficient Matrix
                                                                                                                                                 for i,node in enumerate(graph):
                                                                                                                                                    temp = []
                                                                                                                                                    for j,other_node in enumerate(graph):
   graph = []
print('Enter Adjacency Matrix with terms separated by a space : ')
                                                                                                                                                       if other node in incoming nodes[i]:
temp.append(damping_factor*(1.0/outgoing[j]))
elif i == j:
   for i in range(n):
    temp_list = input().split(' ')
    graph.append(list(map(int,temp_list)))
                                                                                                                                                         temp.append(-1)
   page_rank_algorithm(graph,d)
                                                                                                                                                    temp.append(0)
coefficients[i] = temp
main()
                                                                                                                                                coefficients_list = []
for key,value in coefficients.items():
    coefficients_list.append(value)
 SyntaxError: multiple statements found while compiling a single statement
Enter the number of nodes: 4
Enter the damping factor: 0.5
Enter Adjacency Matrix with terms separated by a space: 0110
                                                                                                                                                 constant matrix = []
                                                                                                                                                 for i in range(len(graph)):
                                                                                                                                                    constant_matrix.append(damping_factor-1)
0010
                                                                                                                                                 pageranks = np.linalg.solve(np.array(coefficients list),np.array(constant matrix))
1000
                                                                                                                                                 print()
0010
                                                                                                                                                 for i,rank in enumerate(pageranks):
Page Rank of A is 1.2308
                                                                                                                                                    print('Page Rank of {} is {:.4f}'.format(chr(65+i), rank))
Page Rank of B is 0.8077
Page Rank of C is 1.4615
Page Rank of D is 0.5000
                                                                                                                                                 n = int(input('Enter the number of nodes : '))
                                                                                                                                                 d= float(input('Enter the damping factor : '))
Enter the number of nodes: 3
Enter the damping factor: 0.5
Enter Adjacency Matrix with terms separated by a space: 011
                                                                                                                                                graph = []
                                                                                                                                                print(Enter Adjacency Matrix with terms separated by a space : ')
for i in range(n):
                                                                                                                                                   temp_list = input().split(' ')
graph.append(list(map(int,temp_list)))
001
100
Page Rank of A is 1.0769
Page Rank of B is 0.7692
Page Rank of C is 1.1538
                                                                                                                                                page_rank_algorithm(graph,d)
```