|  |  |
| --- | --- |
| Name: | Prerna Sunil Jadhav |
| Sap  Id: | 60004220127 |
| Class: | T. Y. B.Tech (Computer Engineering) |
| Course: | Data Mining and Warehouse Laboratory |
| Course Code: | DJ19CEL501 |
| Experiment  No.: | 09 |

**AIM:** Implementation of HITS Algorithm

CODE:

from math import sqrt

def hits\_algorithm(num\_nodes, graph, iterations): authority\_scores = dict()

hub\_scores = dict()

for i in range(len(graph)): authority\_scores[i] = 1

hub\_scores[i] = 1 incoming\_nodes = dict() for i in range(len(graph)):

temp=[]

for node in graph: if node[i]:

temp.append(node) incoming\_nodes[i] = temp

outgoing\_nodes = dict()

for i,node in enumerate(graph): temp = []

for j,edge in enumerate(node): if edge:

temp.append(graph[j]) outgoing\_nodes[i] = temp

print()

for k in range(iterations): print('Iteration : ',k+1) print('Authority Score') normalization\_value = 0

for i,node in enumerate(graph): authority\_scores[i]=0

for j,other\_node in enumerate(graph): if other\_node in incoming\_nodes[i]:

authority\_scores[i] += hub\_scores[j] normalization\_value += (authority\_scores[i]\*\*2)

normalization\_value = sqrt(normalization\_value) for i in range(num\_nodes):

authority\_scores[i] /= normalization\_value

print('{} :{:.2f}'.format(chr(65+i),authority\_scores[i]),end=' | ') print()

print('Hub Score') normalization\_value = 0

for i,node in enumerate(graph):

hub\_scores[i]=0

for j,other\_node in enumerate(graph): if other\_node in outgoing\_nodes[i]:

hub\_scores[i] += authority\_scores[j] normalization\_value += (hub\_scores[i]\*\*2)

normalization\_value = sqrt(normalization\_value) for i in range(num\_nodes):

hub\_scores[i] /= normalization\_value

print('{} :{:.2f}'.format(chr(65+i),hub\_scores[i]),end=' | ') print("\n\n")

def main():

n = int(input('Enter the no of nodes : ')) graph = []

print('Enter Adjacency Matrix : ') for i in range(n):

temp = input()

temp\_list = temp.split(' ') graph.append(list(map(int,temp\_list)))

k = int(input('Enter No of Iterations to be performed : ')) hits\_algorithm(n, graph, k)

main()

A diagram of a number of squares

Description automatically generated with medium confidenceGRAPH:

OUTPUT:

Enter the no of nodes : 3 Enter Adjacency Matrix : 0 0 1

0 0 1

0 0 0

Enter No of Iterations to be performed : 3

Iteration : 1 Authority Score

A :0.00 | B :0.00 | C :1.00 |

Hub Score

A :0.71 | B :0.71 | C :0.00 |

Iteration : 2 Authority Score

A :0.00 | B :0.00 | C :1.00 |

Hub Score

A :0.71 | B :0.71 | C :0.00 |

Iteration : 3 Authority Score

A :0.00 | B :0.00 | C :1.00 |

Hub Score

A :0.71 | B :0.71 | C :0.00 |