Name: Siddharth Unny Batch: A4 Sap Id: 60004200080

Aim: Implementation of HITS Algorithm

Theory:
Hyperlink Induced Topic Search (HITS) Algorithm is a Link Analysis Algorithm that rates webpages, developed by Jon Kleinberg. This algorithm is used to the web link-structures to discover and rank the webpages relevant for a particular search.
HITS uses hubs and authorities to define a recursive relationship between webpages. Before understanding the HITS Algorithm, we first need to know about Hubs and Authorities.

Given a query to a Search Engine, the set of highly relevant web pages are called Roots. Theyare potential Authorities.
Pages that are not very relevant but point to pages in the Root are called Hubs. Thus, an Authority is a page that many hubs link to whereas a Hub is a page that links to many authorities.

Working:
In the HITS algorithm, the first step is to retrieve the most relevant pages to the search query. This set is called the root set and can be obtained by taking the top pages returned by a text-based search algorithm. A base set is generated by augmenting the root set with all the web pages that are linked from it and some of the pages that link to it. The web pages in the base set and all hyperlinks among those pages form a focused subgraph. The HITS computation is performed only on this focused subgraph. According to Kleinberg the reason for constructing a base set is to ensure that most (or many) of the strongest authorities are included.

Authority and hub values are defined in terms of one another in a mutual recursion. An authority value is computed as the sum of the scaled hub values that point to that page. A hub value is the sum of the scaled authority values of the pages it points to. Some implementations also consider the relevance of the linked pages.

The algorithm performs a series of iterations, each consisting of two basic steps:

- **Authority update**: Update each node's authority score to be equal to the sum of thehub scores of each node that points to it. That is, a node is given a high authority score by being linked from pages that are recognized as Hubs for information.
- **Hub update**: Update each node's hub score to be equal to the sum of the authorityscores of each node that it points to. That is, a node is given a high hub score bylinking to nodes that are considered to be authorities on the subject.

The Hub score and Authority score for a node is calculated with the following algorithm:

➔ Start with each node having a hub score and authority score of 1.
➔ Run the authority update rule
➔ Run the hub update rule
➔ Normalize the values by dividing each Hub score by the square root of the sum of the squares of all Hub scores, and dividing each Authority score by the square root of the sum of the squares of all Authority scores.
➔ Repeat from the second step as necessary.

Comparison:
HITS, like Page and Brin's PageRank, is an iterative algorithm based on the linkage of the documents on the web. However it does have some major differences:

- It is query dependent, that is, the (Hubs and Authority) scores resulting from the link analysis are influenced by the search terms;
- As a corollary, it is executed at query time, not at indexing time, with the associated hit on performance that accompanies query-time processing.
- It is not commonly used by search engines. (Though a similar algorithm was said to be used by Teoma, which was acquired by Ask Jeeves/Ask.com.)
- It computes two scores per document, hub and authority, as opposed to a single score;
- It is processed on a small subset of ‘relevant’ documents (a 'focused subgraph' or base set), not all documents as was the case with PageRank.

Advantages:
1. HITS scores due to its ability to rank pages according to the query string, resulting in relevant authority and hub pages.
2. HITS is sensitive to user queries (as compared to PageRank).
3. Important pages are obtained on the basis of calculated authority and hub value.

Disadvantages:

1. Since HITS is a query dependent algorithm the query time evaluation is expensive.
2. The rating or scores of authorities and hubs could rise due to flaws done by the web page designer. HITS assumes that when a user creates a web page he links a hyperlink from his page to another authority page, as he honestly believes that the authority page is in some way related to his page (hub).
3. Topic drift occurs when there are irrelevant pages in the root set and they are strongly connected. Since the root set itself contains non-relevant pages, this will reflect on the pages in the base set.

Algorithm

```
G := set of pages
for each page p in G do
   p.auth = 1 // p.auth is the authority score of the page p
   p.hub = 1 // p.hub is the hub score of the page p
for step from 1 to k do // run the algorithm for k stepsnorm
   = 0
   for each page p in G do // update all authority values first
      p.auth = 0
      for each page q in p.incomingNeighbors do // p.incomingNeighbors is the set of pages that link
to p
         p.auth += q.hub
      norm += square(p.auth) // calculate the sum of the squared auth values to normalise norm
   = sqrt(norm)
   for each page p in G do // update the auth scores p.auth
      = p.auth / norm // normalise the auth values
   norm = 0
   for each page p in G do // then update all hub valuesp.hub
      = 0
      for each page r in p.outgoingNeighbors do // p.outgoingNeighbors is the set of pages that p links
to
         p.hub += r.auth
      norm += square(p.hub) // calculate the sum of the squared hub values to normalise norm =
   sqrt(norm)
   for each page p in G do // then update all hub valuesp.hub =
      p.hub / norm // normalise the hub values
```

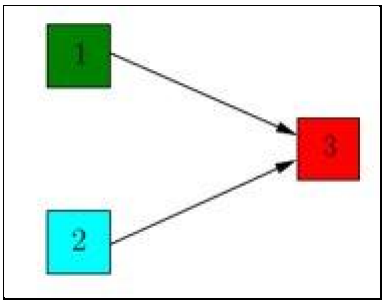
Name: Siddharth Unny Batch: A4 Sap Id: 60004200080

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()
```

Graph:



Output:

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

Name: Siddharth Unny Batch: A4 Sap Id: 60004200080

```
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 |
```

Conclusion:

Hyperlink-Induced Topic Search (HITS) is a link analysis algorithm that rates Web pages. HITS, like Page and Brin's PageRank, is an iterative algorithm based on the linkage of the documents on the web. However its disadvantages outweigh its advantages and thus it is not commonly used in search engines as compared to the PageRank algorithm which provesto be more efficient on large datasets.