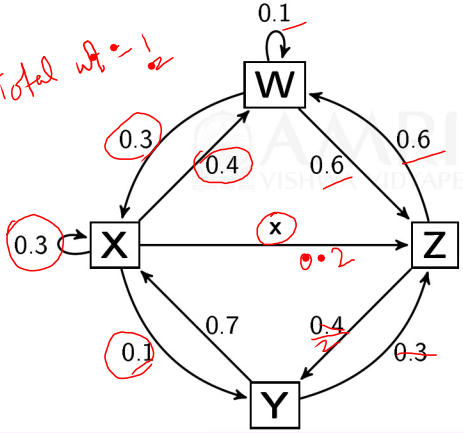
GRAPHS WITH WEIGHTED LINKS

- Assume that deportation probability α=0.2.
- Note that probability of clicking on a link is given in the graph.



$$T_{WY} = .05 + .8 * 0 = .05$$

$$T_{YZ} = .05 + .8 * .4 = .37$$

$$\begin{array}{c} \\ W \\ X \\ Y \\ Z \end{array} \begin{bmatrix} .13 & .37 & .05 & .53 \\ .29 & .29 & .61 & .05 \\ .05 & .13 & .05 & .37 \\ .53 & .21 & .29 & .05 \end{bmatrix} \begin{bmatrix} .25 \\ .25 \\ .25 \\ .25 \end{bmatrix}$$

GRAPHS WITH WEIGHTED LINKS ..

$$R_0 = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}, R_1 = \begin{bmatrix} 0.2700 \\ 0.3100 \\ 0.1500 \\ 0.2700 \end{bmatrix}, R_2 = \begin{bmatrix} 0.3004 \\ 0.2732 \\ 0.1612 \\ 0.2652 \end{bmatrix}, R_3 = \begin{bmatrix} 0.2888 \\ 0.2779 \\ 0.1567 \\ 0.2766 \end{bmatrix}, R_4 = \begin{bmatrix} 0.2948 \\ 0.2738 \\ 0.1607 \\ 0.2707 \end{bmatrix},$$

$$R_5 = \begin{bmatrix} 0.2911 \\ 0.2765 \\ 0.1585 \\ 0.2739 \end{bmatrix}, R_6 = \begin{bmatrix} 0.2932 \\ 0.2750 \\ 0.1598 \\ 0.2720 \end{bmatrix}, ..., R_{11} = \begin{bmatrix} 0.2924 \\ 0.2756 \\ 0.1593 \\ 0.2727 \end{bmatrix}, R_{12} = \begin{bmatrix} 0.2925 \\ 0.2755 \\ 0.1593 \\ 0.2727 \end{bmatrix}, R_{13} = \begin{bmatrix} 0.2924 \\ 0.2755 \\ 0.1593 \\ 0.2727 \end{bmatrix}$$

OBJECTIVES

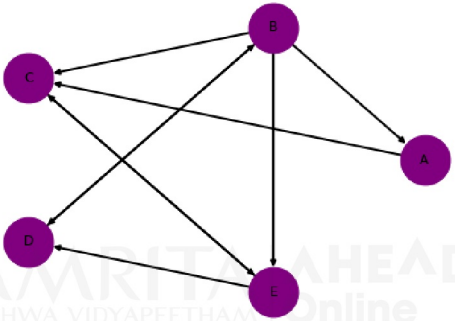
Examples on PageRank Algorithm

- NetworkX

PageRank in NetworkX

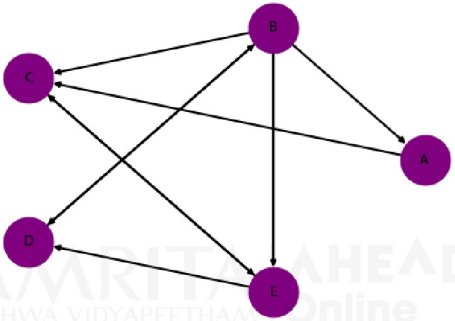
```
import networkx as nx
import matplotlib . pyplot as plt
edge_widths = [2]
G2 = nx.DiGraph([('B', 'A'),('A', 'C'),('B', 'C'),
('B', 'D'), ('B', 'E'), ('C', 'E')('D','B'),
('E', 'C'), ('E', 'D') ])
pos=nx.circular_layout(G2)
nx.draw (G2 , with_labels = True, node_size=2000, node_color="purple",
         width=edge_widths, pos=pos)
plt.savefig("/home/sajeev/Dropbox/Amrita/Courses/AHEAD/
Lecture/images/CN/pagerank_ex_1.png")
```

PageRank in NetworkX




```
pr = nx.pagerank(G2)
{'A': 0.07129426630386904,
'B': 0.19432595907703074,
'C': 0.25392478409597197,
'D': 0.19332465773768306,
'E': 0.28713033278544525}
```

Transition Matrix




```
google_matrix(G2)
matrix([[0.03   , 0.2425, 0.03   , 0.03   , 0.03   ],
        [0.03   , 0.03  , 0.03   , 0.88   , 0.03   ],
        [0.88   , 0.2425, 0.03   , 0.03   , 0.455  ],
        [0.03   , 0.2425, 0.03   , 0.03   , 0.455  ],
        [0.03   , 0.2425, 0.88   , 0.03   , 0.03   ]])
```

OBJECTIVES

HITS Algorithm

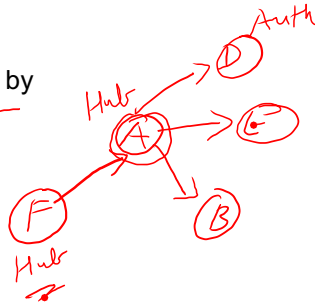
- Drawbacks of PageRank
- Introduction to HITS

PageRank

- Richer-get-richer phenomenon
  - May be difficult for new pages with few inlinks to compete with older, highly linked pages with high PageRank
  - Could promote small fraction of new pages at random or add decay factor to links
- Study (Amento et al., Does authority mean quality?) showed just counting number of inlinks gives similar ranking as PageRank

Emphasize on mutual reinforcement between authority and hub webpages
PageRank emphasizes hyperlink weight normalization and web surfing based on random walk models.

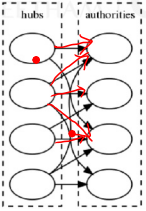
An ideal website should link to other relevant sites and also being linked by other important sites.



Authority: A node is high-quality if many high-quality nodes link to it
Hub: A node is high-quality if it links to many high-quality nodes

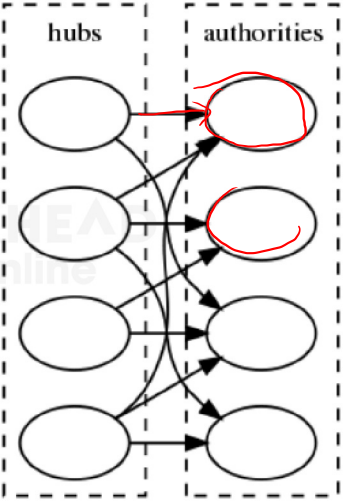
HITS Algorithm

- Hyperlink-induced topic search (HITS) by Jon Kleinberg
  - Hub: page with outlinks to informative web pages
  - Authority: informative/authoritative page with many inlinks
  - Recursive definition:
    - Good hubs point to good authorities Good authorities are pointed to by good hubs



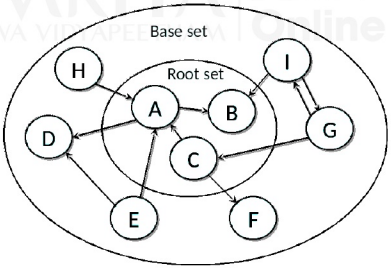
HITS Algorithm: Basics

- HITS is applied on a subgraph after a search is done on the complete graph
- Uses hubs and authorities to define a recursive relationship between web pages
- An authority is a page that many hubs link to
- A hub is a page that links to many authorities



HITS Algorithm

- HITS Algorithm Simplified
  - Retrieve pages most relevant to search query: root set
  - Retrieve all pages linked to/from the root set: base set
  - Perform authority and hub calculations iteratively on all nodes in the subgraph
  - When finished, every node has an authority score and hub score
- Resulting subgraph between 1K - 5K pages



HITS Algorithm

Calculate H and A

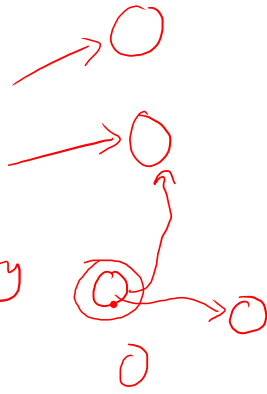
E is set of all directed edges in subgraph
e_{qp} is edge from page *q* to *p*

$$A(p) = \sum_{q:e_{qp} \in E} H(q)$$

inlinks

$$H(p) = \sum_{q:e_{pq} \in E} A(q)$$

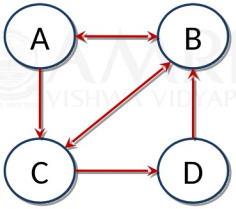
outlinks



- H and A scores computed repetitively until they converge, about 10-15 iterations
- Can also be calculated efficiently using matrix multiplication

HITS: EXAMPLE

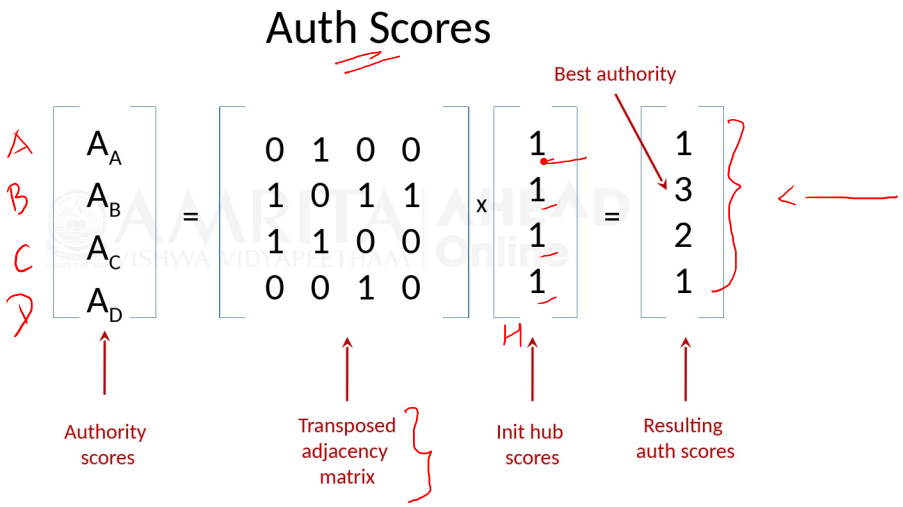
Example Subgraph



	A	B	C	D	
A	0	1	1	0	A
B	1	0	1	0	B
C	0	1	0	1	C
D	0	1	0	0	D

Adjacency matrix

HITS: EXAMPLE ..



HITS: EXAMPLE ..

Hub Scores

Best hub

$$\begin{bmatrix} H_A \\ H_B \\ H_C \\ H_D \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 4 \\ 3 \end{bmatrix}$$

Hub scores

Adjacency matrix

Init auth scores

Resulting hub scores

OBJECTIVES

More on HITS Algorithm

- HITS Examples

HITS Algorithm

Calculate H and A

E is set of all directed edges in subgraph
e_{qp} is edge from page q to p

$$A(p) = \sum_{q:e_{qp} \in E} H(q)$$

inlinks

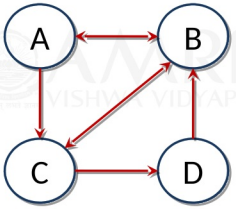
$$H(p) = \sum_{q:e_{pq} \in E} A(q)$$

outlinks

- H and A scores computed repetitively until they converge, about 10-15 iterations
- Can also be calculated efficiently using matrix multiplication

HITS: EXAMPLE

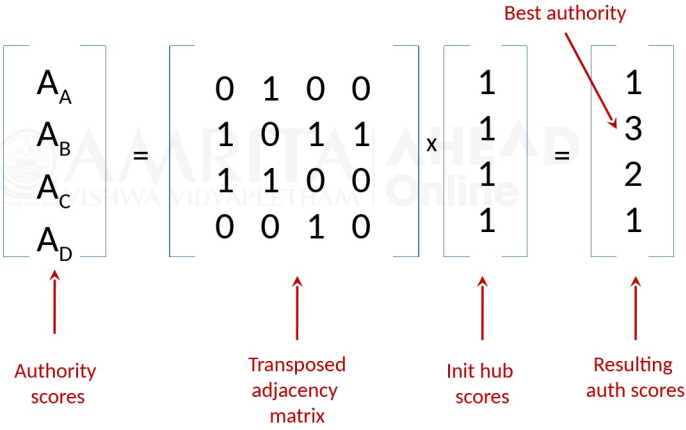
Example Subgraph



	A	B	C	D	
	0	1	1	0	A
	1	0	1	0	B
	0	1	0	1	C
	0	1	0	0	D

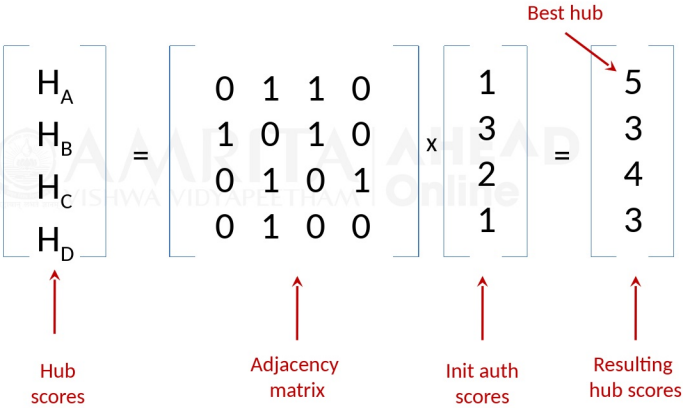
Adjacency matrix

HITS: EXAMPLE ..



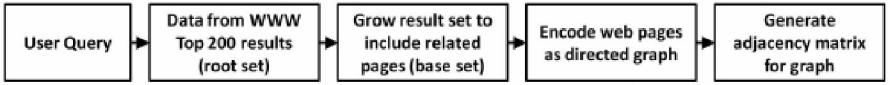
HITS: EXAMPLE ..

Hub Scores

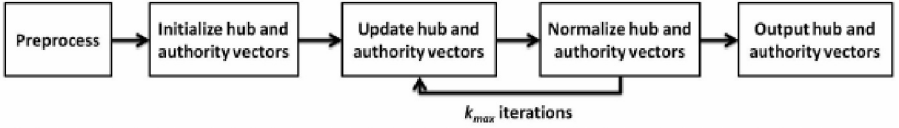


HITS Workflow

HITS Preprocessing



HITS Algorithm



HITS ALGORITHM

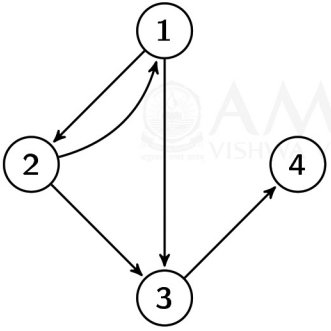
STEPS INVOLVED

- **Select initial set of web pages relevant to the user's query.** An initial set of web pages relevant to the user's query are selected in order to reduce the problem to a manageable size.
- **Initialize vectors.** HITS returns two vectors: one that scores each vertex's importance as a hub and the other that scores each vertex's importance as an authority. Each element of these vectors is initialized to 1.
- **Iteratively update vectors.** The algorithm iterates, kmax times, updating the two score vectors during each iteration.
- **Normalize vector scores.** In order to ensure vectors are scaled appropriately and also reach convergence, they are normalized after each iteration of the algorithm.
- **Output vectors.** The final step of the algorithm is to output the final vectors after k_{max} iterations.

HITS EXAMPLE (1/2)

Transpose of adj. matrix

$$a^{(k)} = L^T h^{(k-1)}, \quad h^{(k)} = L\, a^{(k)}$$



$$L = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad h^{(0)} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Auth. Score

$$a^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

$$h^{(1)} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 1 \\ 0 \end{bmatrix}$$

HITS Example (2/2)

$$\sqrt{3^2+3^2+1^2} = \sqrt{19}$$

$$\Rightarrow \frac{1}{\sqrt{19}} \begin{bmatrix} 3 \\ 3 \\ 1 \\ 0 \end{bmatrix}$$

Normalized value

$$a^{(2)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \frac{1}{\sqrt{19}} \begin{bmatrix} 3 \\ 3 \\ 1 \\ 0 \end{bmatrix}$$

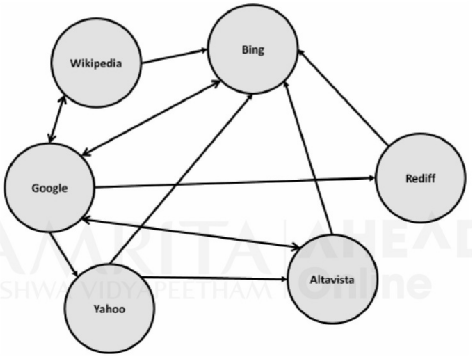
$$= \frac{1}{\sqrt{19}} \begin{bmatrix} 3 \\ 3 \\ 6 \\ 1 \end{bmatrix} \Rightarrow \frac{1}{\sqrt{55 \times 19}} \begin{bmatrix} 3 \\ 3 \\ 6 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.092 \\ 0.092 \\ 0.185 \\ 0.039 \end{bmatrix}$$

$$h^{(2)} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \frac{1}{\sqrt{55 \times 19}} \begin{bmatrix} 3 \\ 3 \\ 6 \\ 1 \end{bmatrix}$$

$$= \frac{1}{\sqrt{163 \times 55 \times 19}} \begin{bmatrix} 9 \\ 9 \\ 1 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.021 \\ 0.021 \\ 0.002 \\ 0 \end{bmatrix}$$

Node 3 is authority node, nodes 1 & 2 are hubs.

HITS ALGORITHM: EXAMPLE..



	Wiki	Google	Bing	Yahoo	Altavista	Rediff
Wikipedia	0	1	1	0	0	0
Google	1	0	1	1	1	1
Bing	0	1	0	0	0	0
Yahoo	0	0	1	0	1	0
Altavista	0	1	1	0	0	0
Rediffmail	0	0	1	0	0	0

$$\mathbf{a}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{h}^{(0)} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix},$$

HITS Algorithm: Example.. (1/4)

- Update authority scores:

$$a^{(k)} = A^T . h^{(k-1)}$$

$$a^{(1)} = A^T . h^{(0)}$$

$$= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} . \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 5 \\ 1 \\ 2 \\ 1 \end{bmatrix}$$

HITS Algorithm: Example.. (2/4)

- Normalize the vector after each iteration by dividing each element by the square root of the sum of squares of the elements.

$$\begin{bmatrix} \frac{1}{\sqrt{41}} \\ \frac{3}{\sqrt{41}} \\ \frac{5}{\sqrt{41}} \\ \frac{1}{\sqrt{41}} \\ \frac{2}{\sqrt{41}} \\ \frac{1}{\sqrt{41}} \end{bmatrix} = \begin{bmatrix} 0.15617 \\ 0.46852 \\ 0.78087 \\ 0.15617 \\ 0.312348 \\ 0.15617 \end{bmatrix}$$

- Update hub scores:

$$h^{(k)} = A.a^{(k)}$$

$$h^{(1)} = A.a^{(1)}$$

HITS ALGORITHM: EXAMPLE.. (3/4)

$$= \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.15617 \\ 0.46852 \\ 0.78087 \\ 0.15617 \\ 0.312348 \\ 0.15617 \end{bmatrix} = \ldots \begin{bmatrix} 0.454 \\ 0.567 \\ 0.170 \\ 0.397 \\ 0.454 \\ 0.284 \end{bmatrix}$$

- Convergence of authority scores

Iteration (k)	0	1	2	3	4	5	6
Wikipedia	1	0.156	0.204	0.224	0.232	0.236	0.237
Google	1	0.469	0.388	0.350	0.332	0.324	0.320
Bing	1	0.781	0.777	0.769	0.765	0.762	**0.761**
Yahoo	1	0.156	0.204	0.224	0.232	0.236	0.238
Altavista	1	0.312	0.347	0.369	0.378	0.383	0.385
Rediffmail	1	0.156	0.204	0.224	0.232	0.236	0.238

HITS Algorithm: Example.. (4/4)

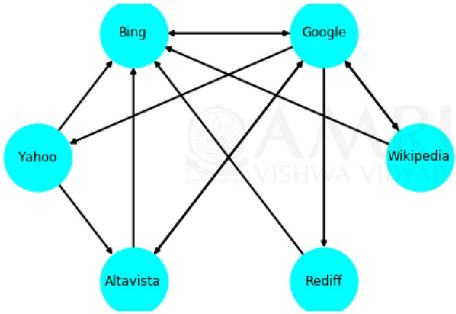
- Convergence of hub scores

Iteration (k)	0	1	2	3	4	5	6
Wikipedia	1	0.454	0.418	0.401	0.393	0.389	0.378
Google	1	0.567	0.624	0.648	0.659	0.664	**0.666**
Bing	1	0.170	0.139	0.126	0.119	0.116	0.115
Yahoo	1	0.397	0.404	0.408	0.409	0.410	0.411
Altavista	1	0.454	0.418	0.401	0.393	0.389	0.387
Rediffmail	1	0.284	0.279	0.276	0.274	0.273	0.273

USING NETWORKX (1/2)

```
import networkx as nx
import matplotlib . pyplot as plt
G2=nx.DiGraph()
G2.add_nodes_from(['Wikipedia',..... 'Rediff'])
G2.add_edges_from([('Wikipedia', 'Google'),...('Rediff', 'Bing')])
pos=nx.circular_layout(G2)
nx.draw (G2 , with_labels = True, node_size=2000,
node_color="cyan", width=edge_widths, pos=pos)
plt.savefig("/home/sajeev/Dropbox/Amrita/Courses/
AHEAD/Lecture/images/CN/hits_ex_1.png")
```

USING NETWORKX (2/2)




```
h,a=nx.hits(G2)
Hubs:
{'Wikipedia': 0.1725885065086705,
 'Google': 0.2985796600753021,
 'Bing': 0.05080519287130723,
 'Yahoo': 0.1836548203986865,
 'Altavista': 0.1725885065086705,
 'Rediff': 0.12178331363736325}

Authority:
{'Wikipedia': 0.10964493614741574,
 'Google': 0.14541326689703665,
 'Bing': 0.3485649495399657,
 'Yahoo': 0.10964493614741574,
 'Altavista': 0.1770869751207503,
 'Rediff': 0.10964493614741574}
```

References

Hagberg, A., Swart, P., and S Chult, D.
Exploring network structure, dynamics, and function using networkx.
Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Newman, M.
Networks.
Oxford university press, 2018.

Zinoviev, D.
Complex network analysis in Python: Recognize-construct-visualize-analyze-interpret.
Pragmatic Bookshelf, 2018.