ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Lesson 5:
Social Network Analysis

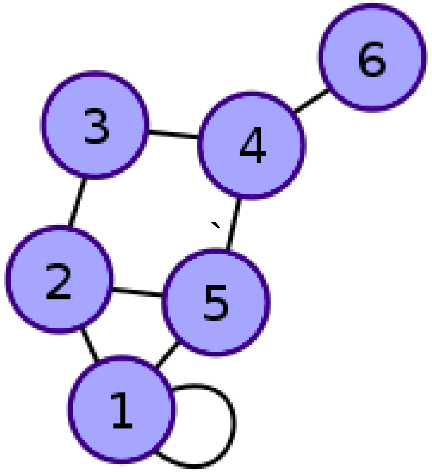
Big problems in Social Network Analysis

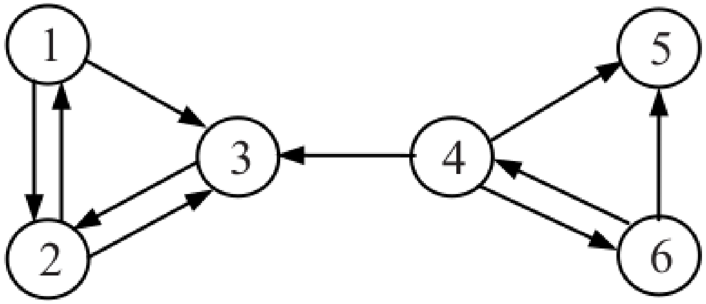
- Graph Ranking: Analyze the role of nodes in graph
- Community detection: Detect communities consisting of members of similar nature
- Link prediction: Predicting the evolution of a graph over time
- Graph classification: Classify the vertices and edges of the graph into given classes

Agenda

1. Graph Ranking
2. Community Detection
3. Graph Representation

1. Graph Ranking
1.1 Basic concepts of graphs



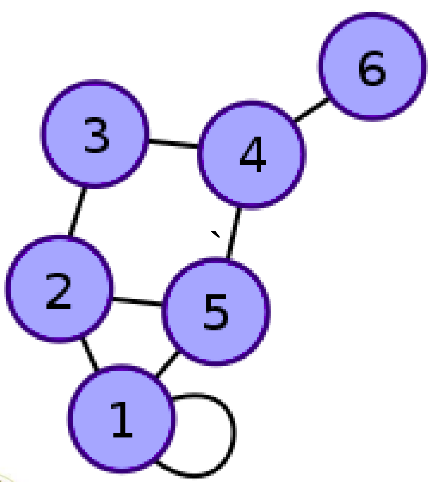


a) Undirected graph

Directed graph

Adjacent Matrix

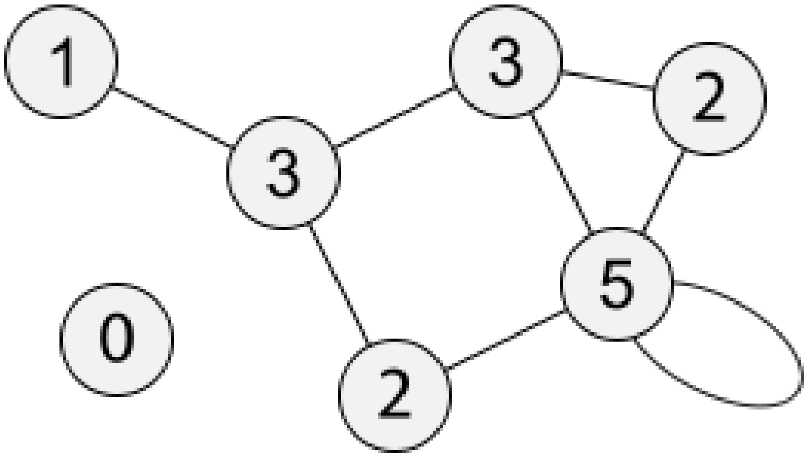
$$a[i, j] = \begin{cases} 1 & \text{if there is edge } (i,j) \\ 2 & \text{if there is a edge from a node to itself} \\ 0 & \text{otherwise} \end{cases}$$

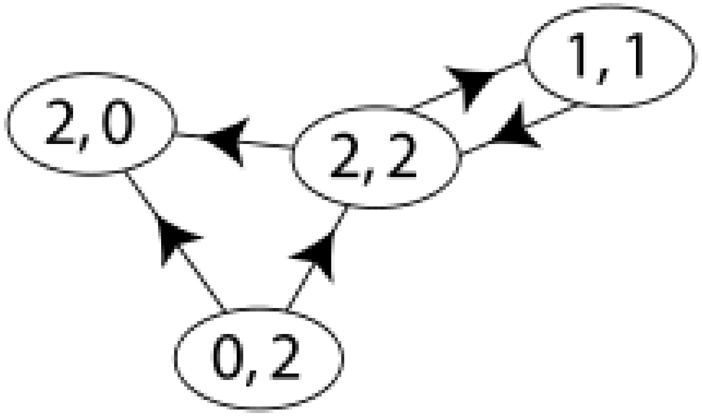


$$\begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Degree of a node

- $d_i(i)$ = number of in-edge of node *i*
- $d_o(i)$ = number of out-edge of node *i*





1.2 Dijkstra algorithm

- Find shortest path from source node *s* to the other nodes of graph
- *d(v)*: Distance from node s to node *v*

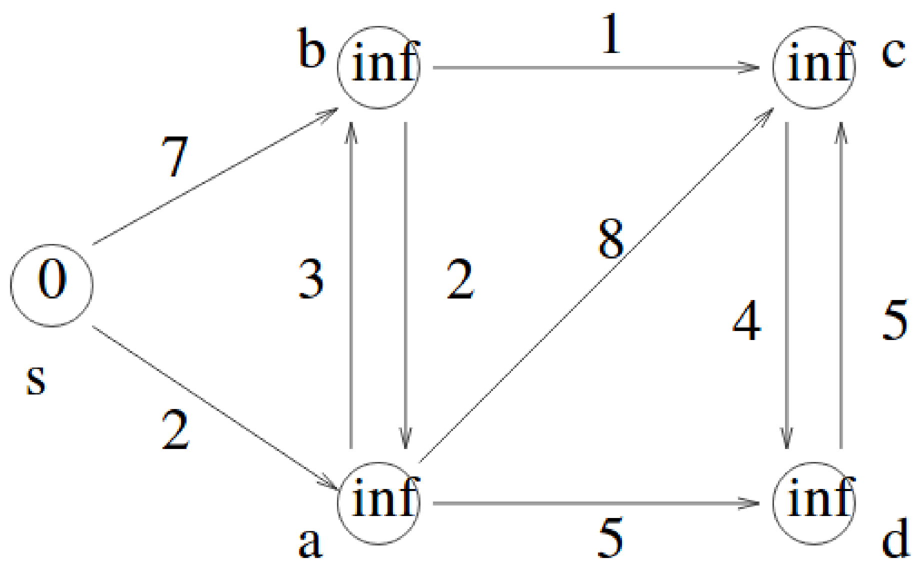
  S1: Initialize $d(s) = 0$; $d(v) = \infty$

  S2: Arrange the nodes in a specific order in a queue Q

  S3: Get node *u* from queue Q then update distance *d(v)* (if needed) of every node *v* adjacent to node *u*

  Go back to step **S2** until every node is computed

Example

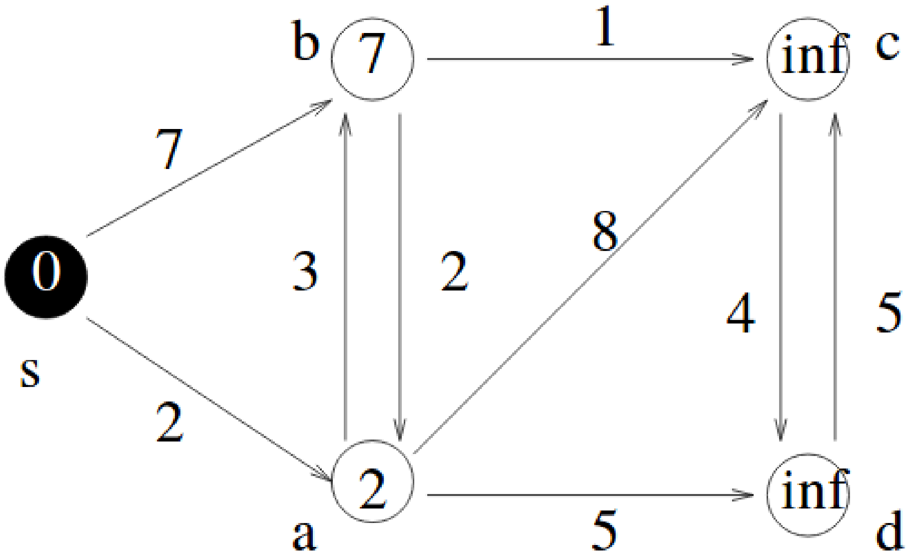


Example (cont.)

v	s	a	b	c	d
$d[v]$	0	∞	∞	∞	∞
$pred[v]$	nil	nil	nil	nil	nil
$color[v]$	W	W	W	W	W

v	s	a	b	c	d
$d[v]$	0	∞	∞	∞	∞

Example (cont.)

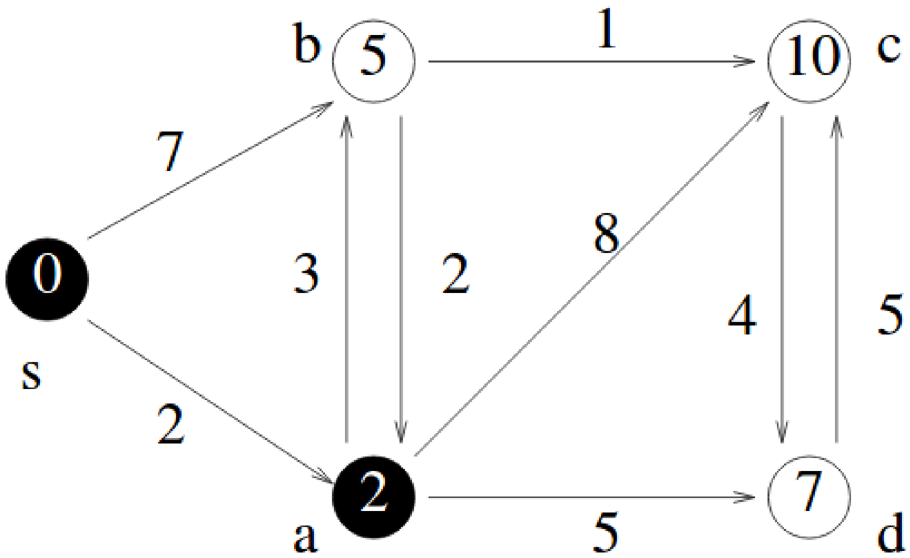


Example (cont.)

v	s	a	b	c	d
$d[v]$	0	2	7	∞	∞
$pred[v]$	nil	s	s	nil	nil
$color[v]$	B	W	W	W	W

v	a	b	c	d
$d[v]$	2	7	∞	∞

Example (cont.)

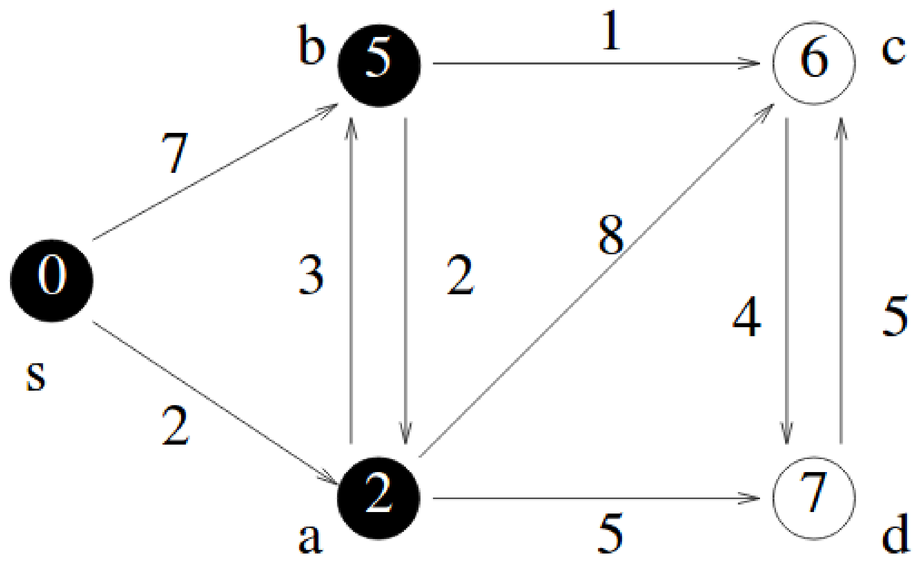


Example (cont.)

v	s	a	b	c	d
$d[v]$	0	2	5	10	7
$pred[v]$	nil	s	a	a	a
$color[v]$	B	B	W	W	W

v	b	c	d
$d[v]$	5	10	7

Example (cont.)

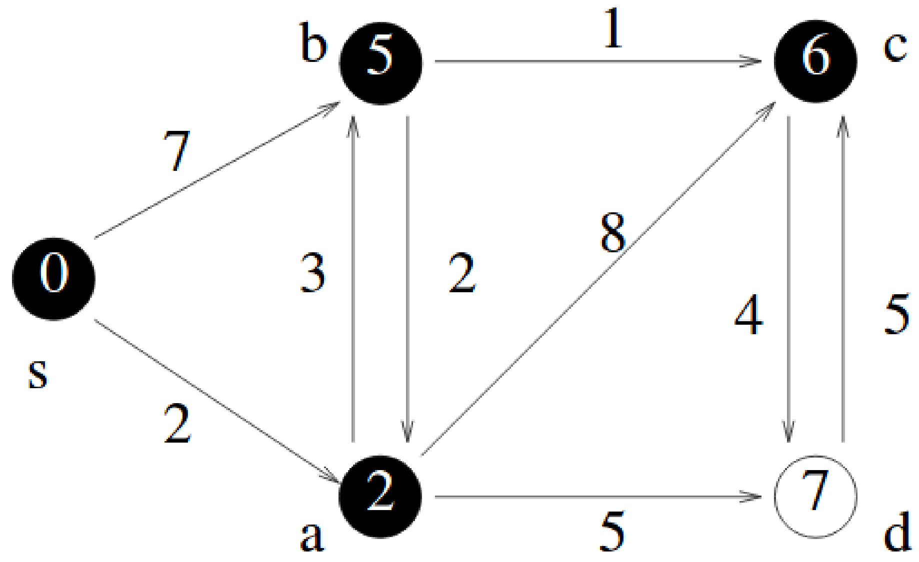


Example (cont.)

v	s	a	b	c	d
$d[v]$	0	2	5	6	7
$pred[v]$	nil	s	a	b	a
$color[v]$	B	B	B	W	W

v	c	d
$d[v]$	6	7

Example (cont.)

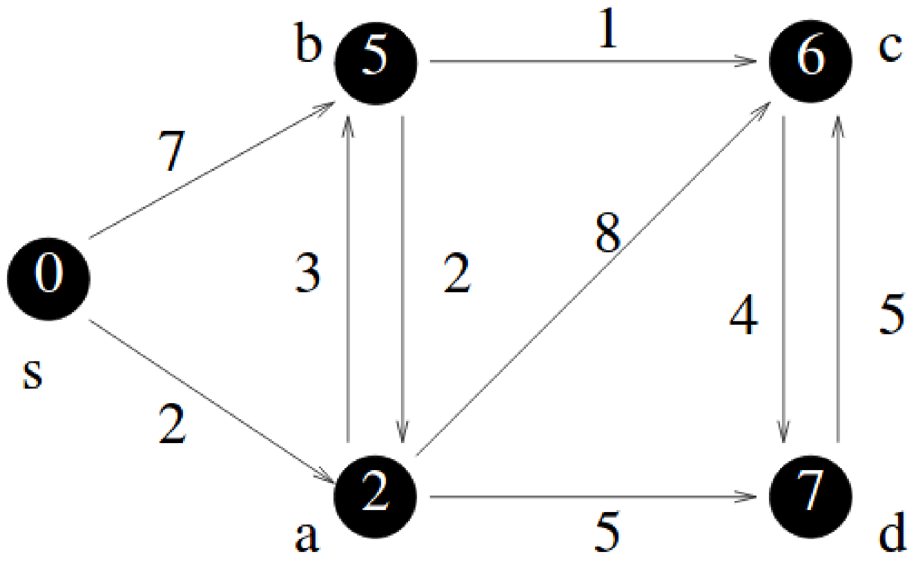


Example (cont.)

v	s	a	b	c	d
$d[v]$	0	2	5	6	7
$pred[v]$	nil	s	a	b	a
$color[v]$	B	B	B	B	W

v	d
$d[v]$	7

Example (cont.)



Example (cont.)

v	s	a	b	c	d
$d[v]$	0	2	5	6	7
$pred[v]$	nil	s	a	b	a
$color[v]$	B	B	B	B	B

$$Q = \emptyset.$$

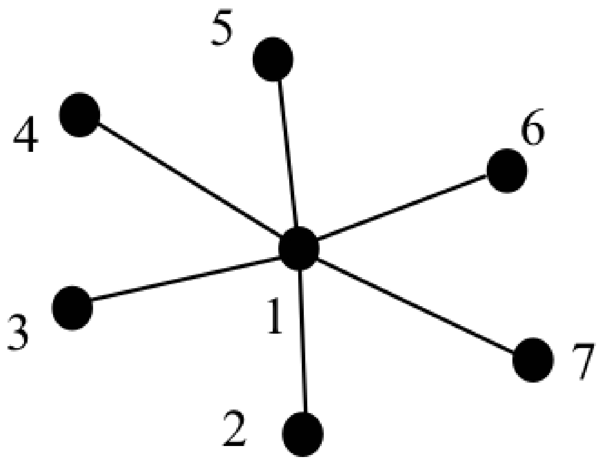
1.3 Degree Centrality
Closeness Centrality

$$C_C(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)}.$$

d(i, j): shortest distance from node i node j

Betweeness Centrality

$$C_B(i) = \sum_{j<k} \frac{p_{jk}(i)}{p_{jk}}.$$



$p_{jk}(i)$: Number of shortest path from node ***j*** to node ***k*** the pass node i ***i***

$C_B(1) = 15$, $C_B(2) = C_B(3) = C_B(4) = C_B(5) = C_B(6) = C_B(7) = 0$

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

1.4 Prestige
Degree Prestige

$$P_D(i) = \frac{d_I(i)}{n-1},$$

$d_i(i)$: in-degree of node i

Proximity Degree

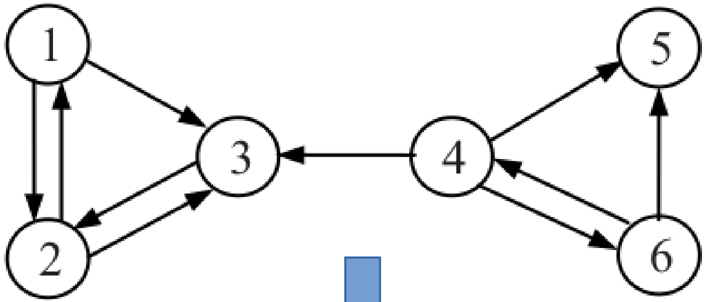
$$P_P(i) = \frac{|I_i| / (n-1)}{\sum_{j \in I_i} d(j,i) \Big/ |I_i|},$$

I_i: Set of nodes that can reach node i

1.5 PageRank Algorithm

- Rank graphs based on general structure
- For large graphs, the rank is approximated by an iterative algorithm based on the 'random walk’
- Important applications in web search engines
- Cons: Doesn't depend on the query

Hyperlink graph



$$A = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}.$$

Hyperlink graph (cont.)

Standardize:

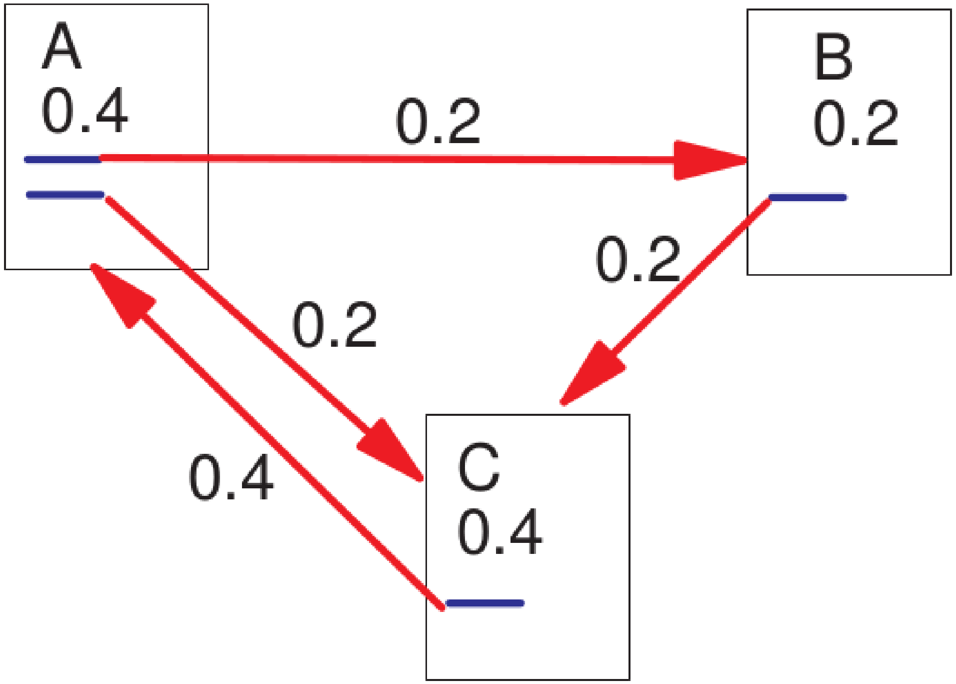
$$A = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}. \Rightarrow \overline{A} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}.$$

Formula

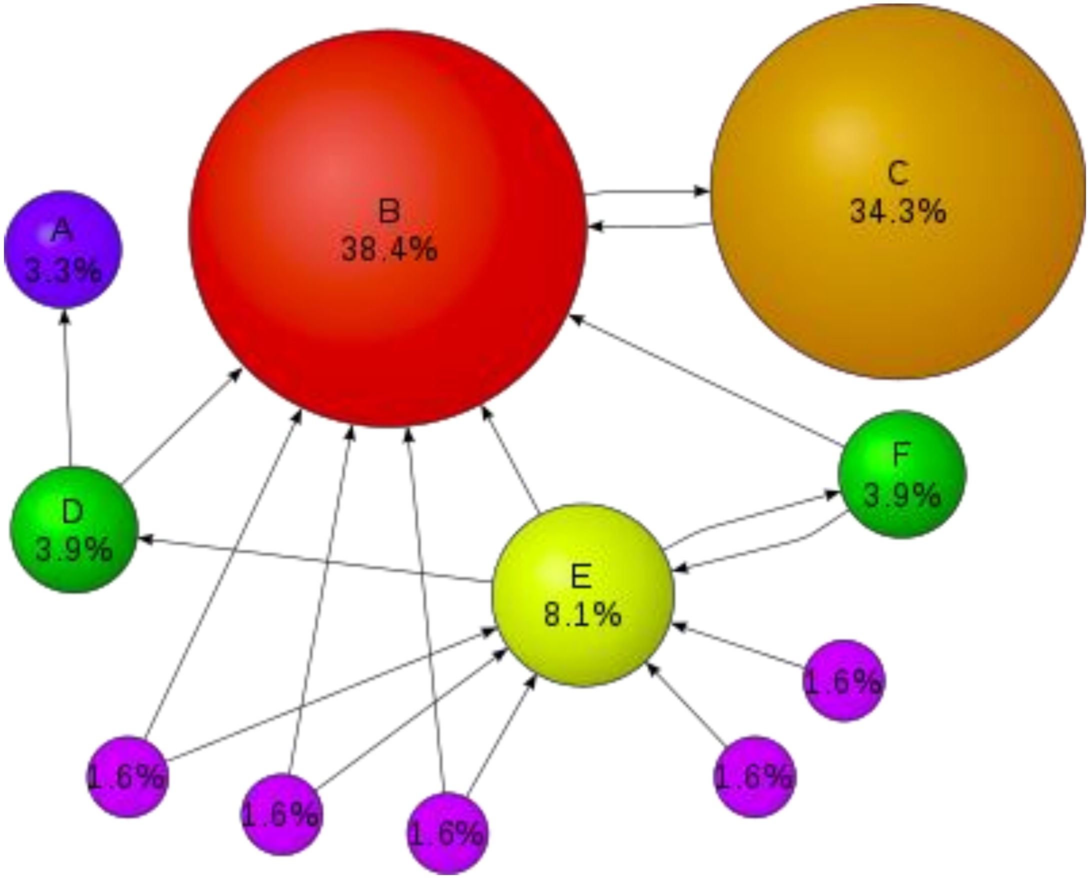
$R(A) = (1 - d) / N + d * \Sigma_{B:(B,A) \in E} R(B) / d_o(B)$

R(A): Thứ hạng của đỉnh A
d: damping factor
N: số đỉnh của đồ thị
(B,A) cạnh của đồ thị
d_o(B) bậc ra của đỉnh B

Example (d = 1)



Example (d = 0.85)

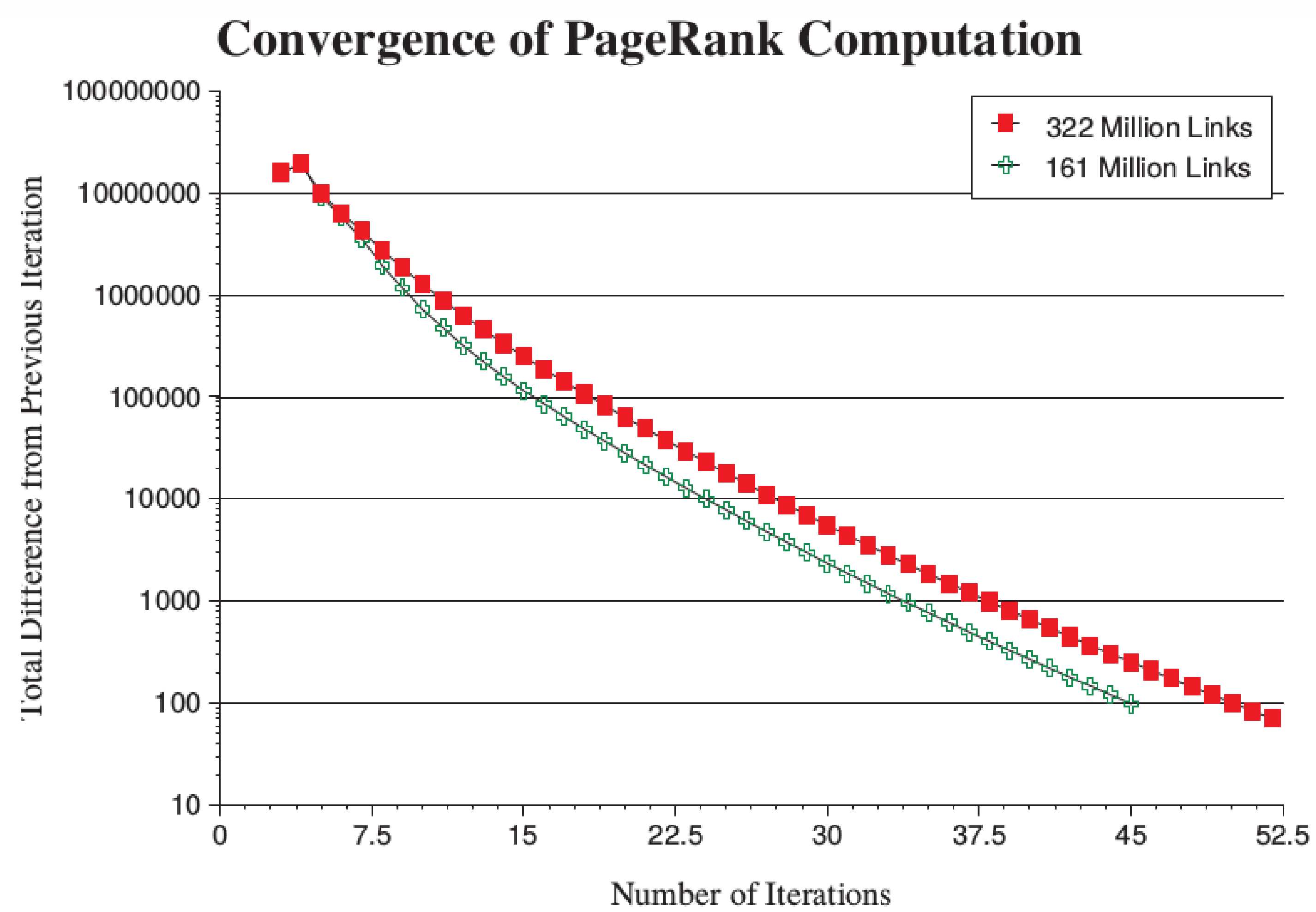


Algorithm

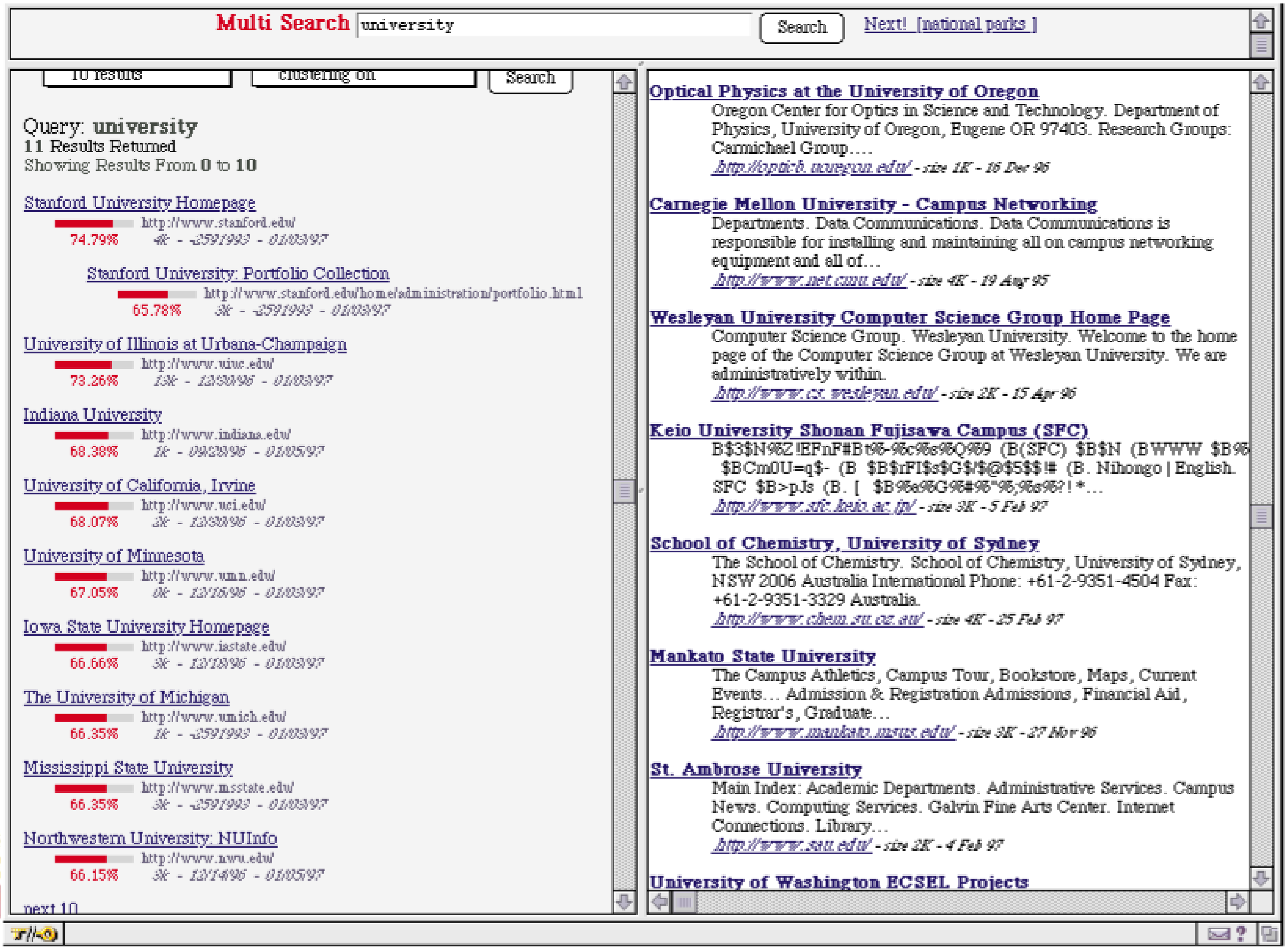
Algorithm PageRank(d, E)

1. Init page ranks $R^{(0)}$;
2. $i = 1$;
3. **repeat**
4. **for** each page A **do**
5. $R^{(i)}(A) = (1 – d) \ / N + d * \Sigma_{B:(B,A) \in E} R^{(i-1)}(B) / d_o(B)$;
6. **endfor**
7. i++;
8. **until** converged

Convergence speed

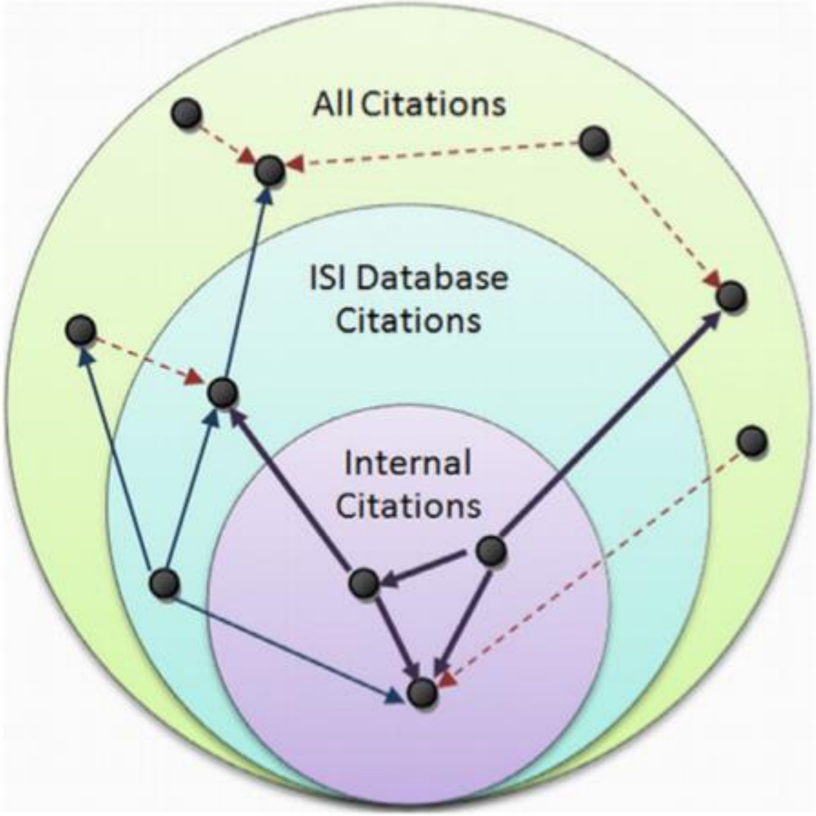


Application: Web Search

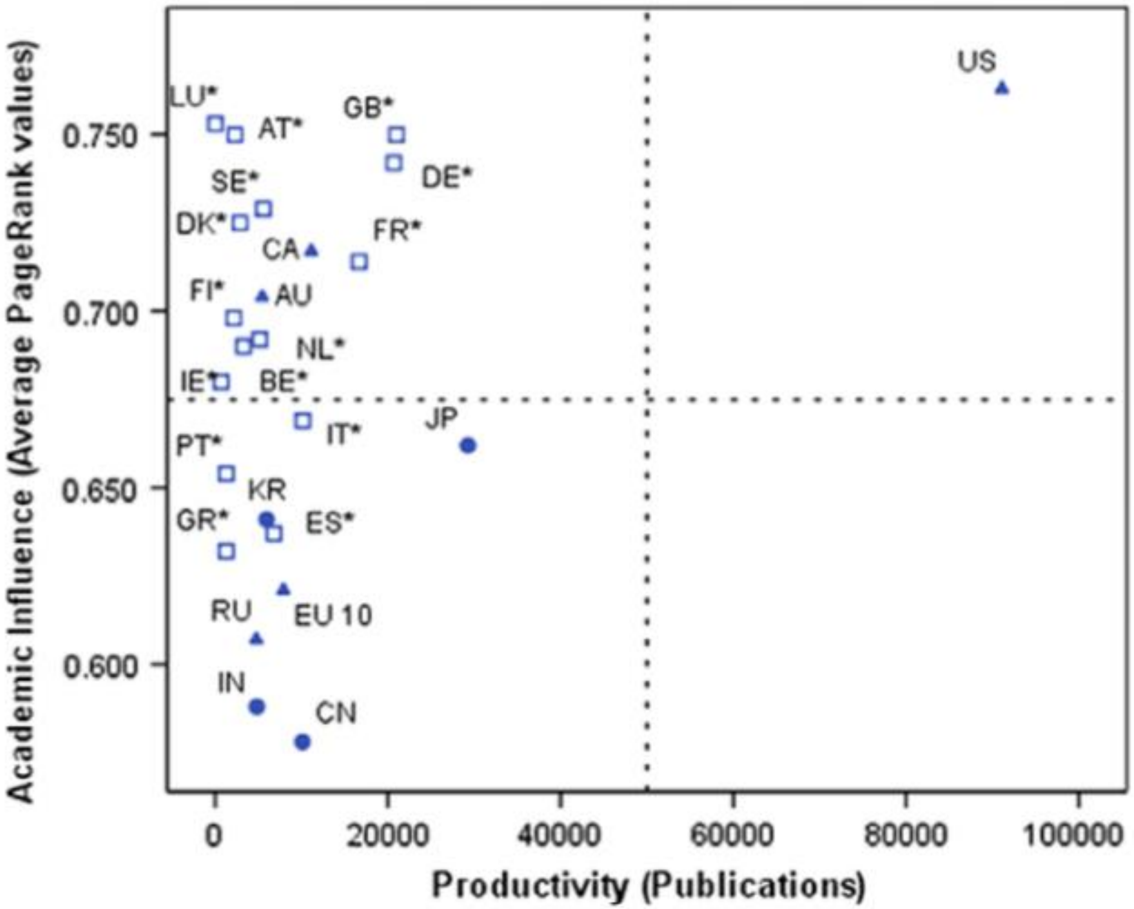


Application: Citation analysis

Guan et al. 2008. “*Bringing Page-Rank to the Citation Analysis*”



Application: Citation analysis (cont.)



1.6 HITS Algorithm

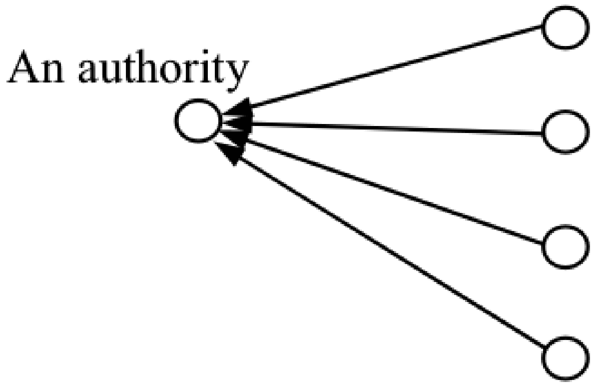
- Hypertext Induced Topic Search
- J. Kleinberg. "*Authoritative Sources in a Hyperlinked Environment.*" In Proc. of the 9th ACM SIAM Symposium on Discrete Algorithms (SODA'98), pp. 668–677, 1998.

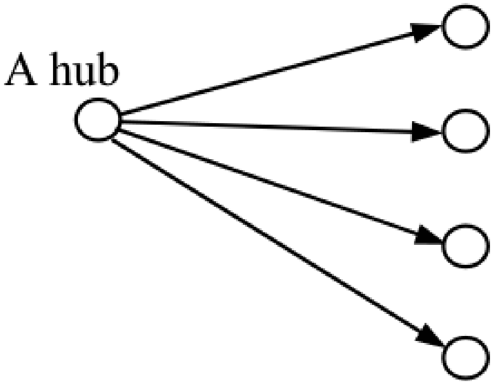
	Spam filtering	**Query relevance**	**Execution**
HIST	✗	✓	Online
PageRank	✓	✗	Offline

Authority/Hub

Authority: pages with many in-links

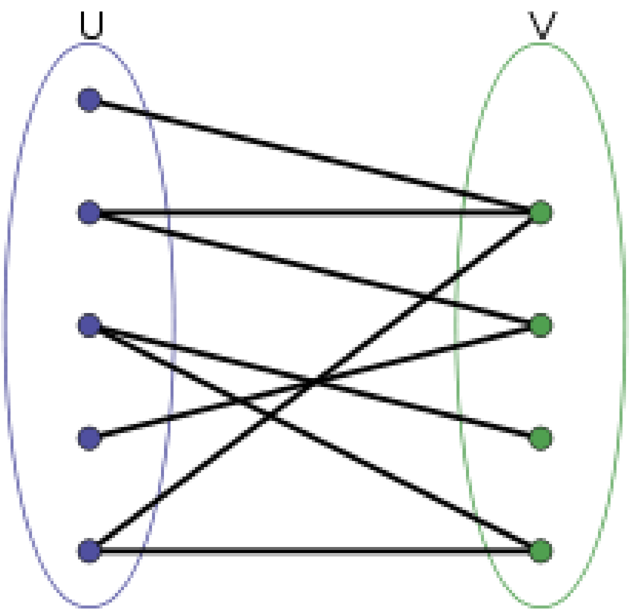
Hub: pages with many out-links

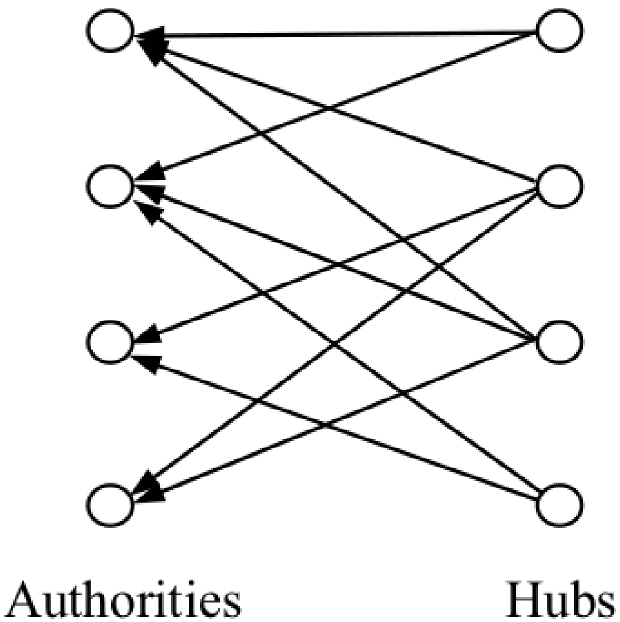




Bigraph

- Graph divided into 2 separated set of node such that every edge connects 2 node of different s





Algorithm

Input: *Query q*

Output: authority score and hub score of relevant pages of query *q*

Algorithm:

1 – Retrieve information

2 – Expand graph

3 - Compute rank

1-Retrieve information

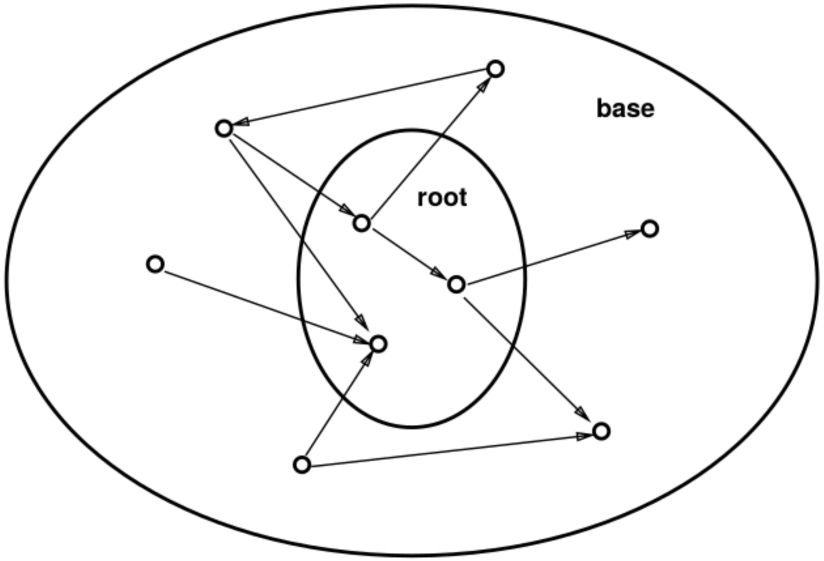
Requires a search engine has relevant documents of query q

- Input query q and a root set **W** of top k pages relevant to q

2- Expand graph

From root set W, expand to base set **S**

- For each page *p* in ***W***
  - Insert pages that p links to
  - Insert pages that links to p



3- Compute rank

Authority score (a)

Hub score (h)

G = (V,E)

$$L_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

$$a(i) = \sum_{(j,i)\in E} h(j) \qquad \sum_{i=1}^{n} a(i) = 1$$

$$h(i) = \sum_{(i,j)\in E} a(j) \qquad \sum_{i=1}^{n} h(i) = 1$$

3- Compute rank (cont.)

$$a = L^T h$$

$$h = La$$

HITS-Iterate(G)

$a_0 \leftarrow h_0 \leftarrow (1, 1, \ldots, 1)$;
$k \leftarrow 1$
Repeat
$a_k \leftarrow L^T L a_{k-1}$;
$h_k \leftarrow L L^T h_{k-1}$;
$a_k \leftarrow a_k / ||a_k||_1$; // normalization
$h_k \leftarrow h_k / ||h_k||_1$; // normalization
$k \leftarrow k + 1$;
until $||a_k - a_{k-1}||_1 < \varepsilon_a$ and $||h_k - h_{k-1}||_1 < \varepsilon_h$;
return a_k and h_k

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!