Ahmedabad University

# Link Analysis

Amit A. Nanavati
Ahmedabad University

*ACM Winter School on Network Science, 2023, Ahmedabad University*

*Thanks to*
*Easley, David, and Jon Kleinberg. "Networks, crowds, and markets." Cambridge Books (2012).*

# Searching the Web: The Problem of Ranking

- Why is this a hard problem?
  - How do you search newspaper articles, papers, patents, legal documents..?
  - formats? (text, spreadsheet, pdf,..)
  - synonyms
    - ladyfinger, okra,...
  - polysemy
    - jaguar (car or animal?)

# Searching the Web is *hard*

- harder to rank documents according to a common criterion
  - diversity in authoring styles
  - pages written by experts, novices, children,...
  - dynamic nature of web content
  - "world trade center" query returned "unexpected" results on the day (why?)
  - News search features were added (how?)
  - whatsapp, twitter fill the gap between static content and real-time awareness

# Searching the Web is *hard*

- Web has shifted much of the information retrieval question
  - from a problem of scarcity to a problem of abundance.
- Which few should the search engine recommend?

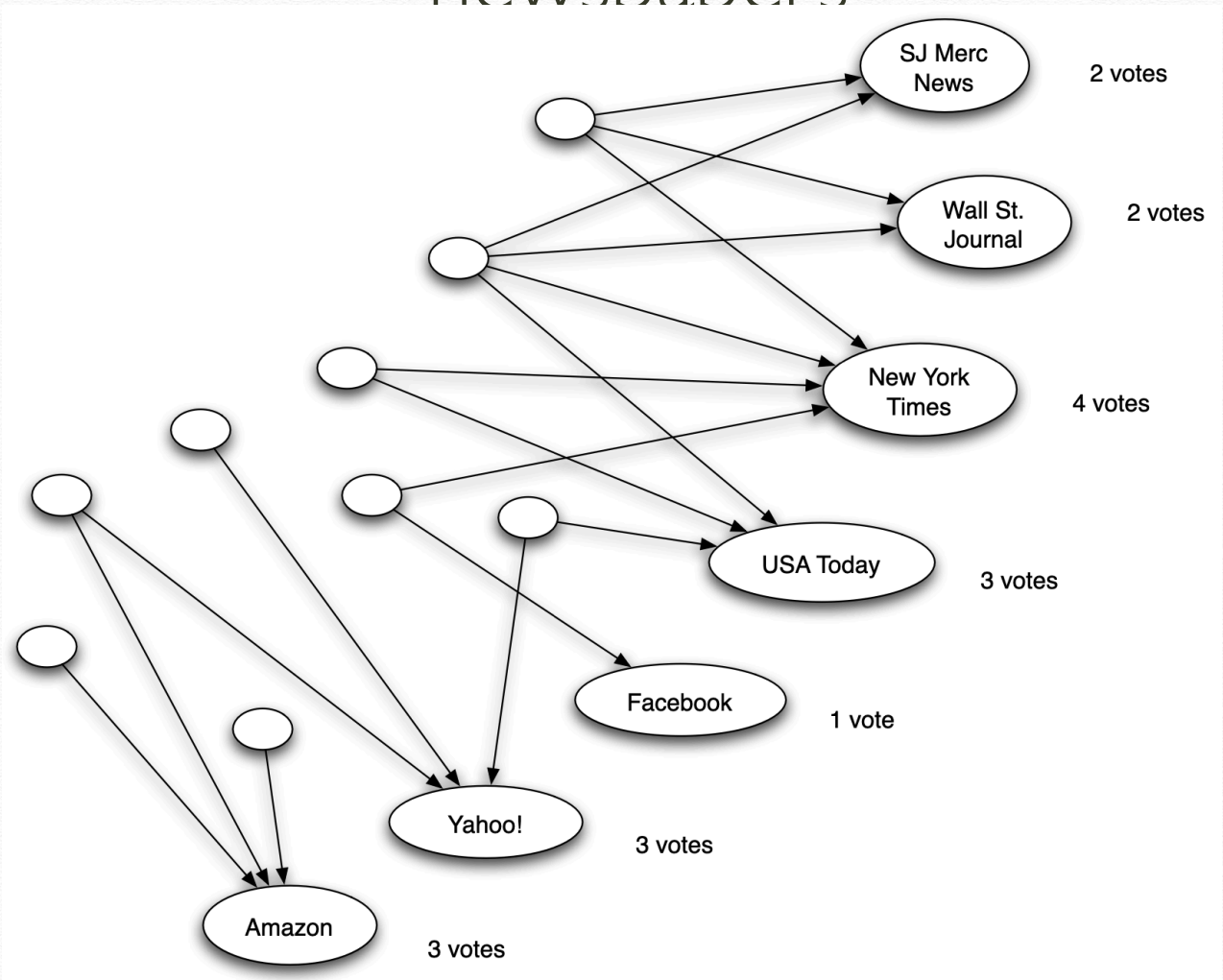
# Voting by in-links

- Which few should the search engine recommend?
- The perspective is to note that there is not really any way to use features purely *internal* to the page to solve this problem.
  - for example, iisc.ac.in does not have the word "IISc" very frequently
- It stands out because of other web pages..
  - when other pages have "iisc" in them, they point to iisc.ac.in
- Step 1: In the case of the query "IISc," we could first collecting a large sample of pages that have "IISc" in them — as determined by a classical, text-only, information retrieval approach.

# Voting by in-links

- This is the first part of the argument that:
  *links are essential to ranking*
  - We can use them to assess the authority of a page on a topic
  - ..through the implicit endorsements that other pages on the topic confer through their links to it.

# Counting in-links to pages for the query "newspapers"



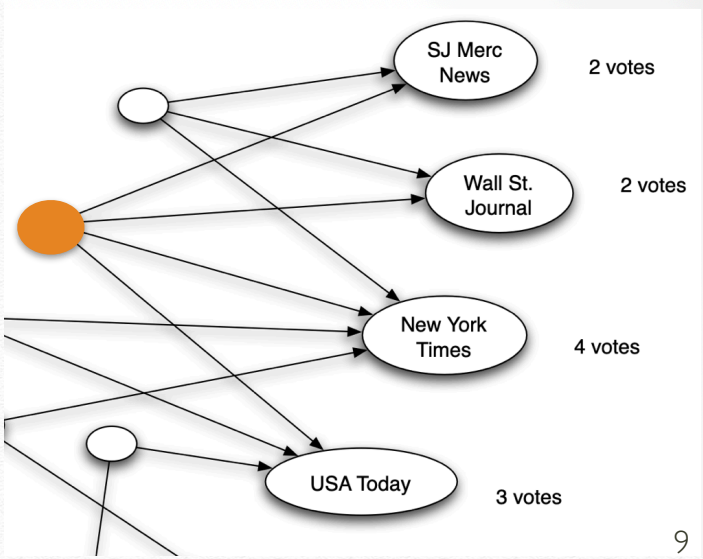
# Voting by in-links

- Suppose, we enter the one-word query "newspapers."
- High scores for a mix of prominent newspapers (good)
- A lot of in-links no matter what the query is — pages like Yahoo!, Facebook, Amazon.. (not so good)
- Top 4 highest vote getters have 2 newspapers and 2 non-newspapers.
- Votes are only a very simple kind of measure.
- In addition to the newspapers themselves, there is another kind of useful answer to our query:
  - Pages that compile lists of newspapers relevant to the topic.

# Voting by in-links

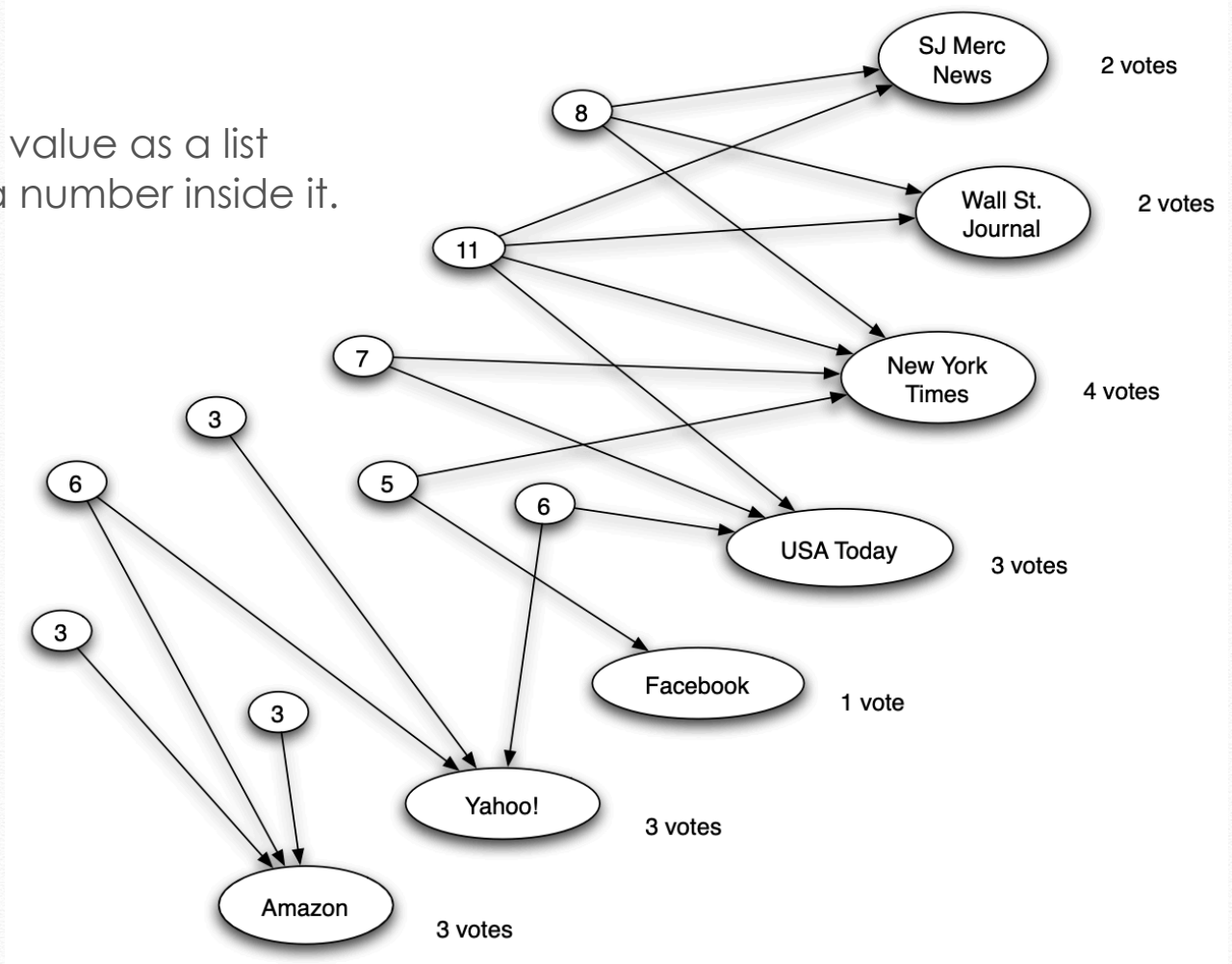
- Pages that compile *lists* of newspapers
  - Suggests a useful technique for *finding* good lists.
  - Among the pages casting votes, a few of them in fact voted for *many* of the pages that received a lot of votes.

- These pages have some sense where the good answers are,
  - So we score them highly as *lists.*



# Counting in-links to pages for the query "newspapers"

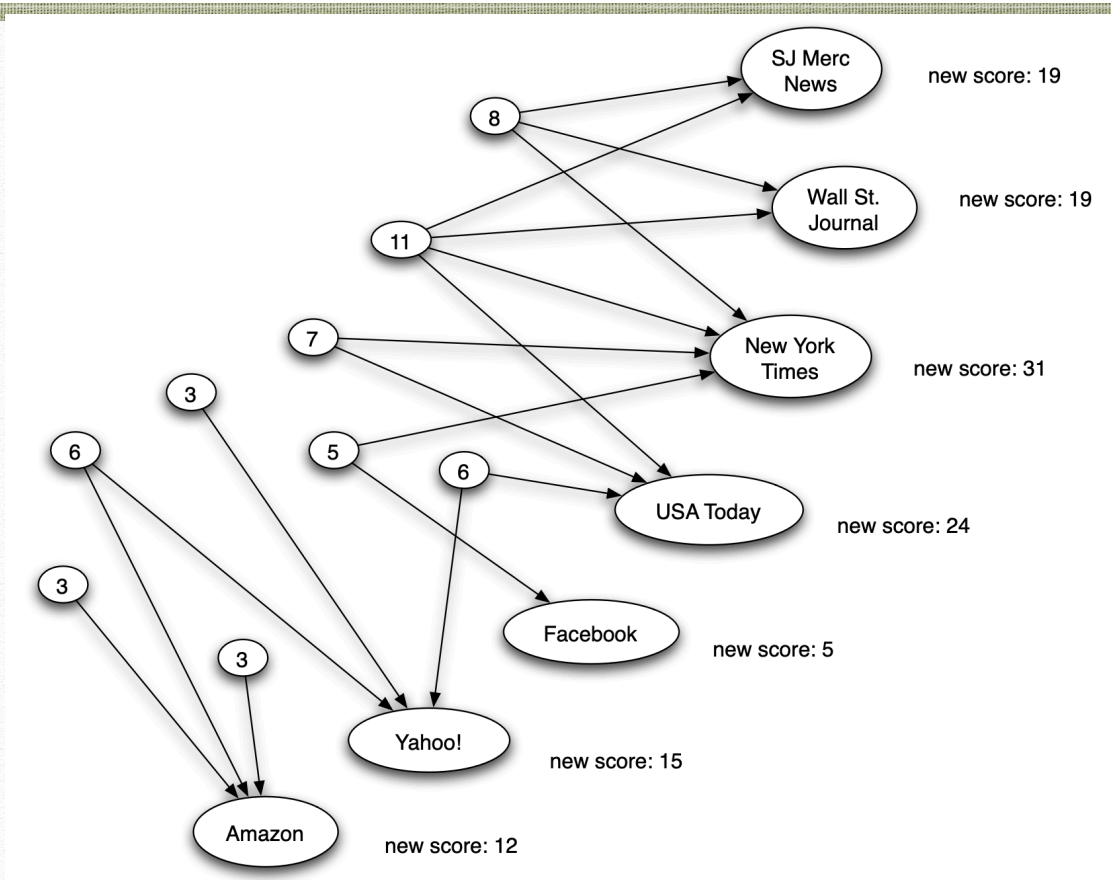
Each page's value as a list is written as a number inside it.



# Voting by in-links

Each of the labeled page's new score is equal to the sum of the values of all lists that point to it.

Now, the real newspapers are doing better than Yahoo!, Facebook and Amazon.




- Pages scoring well as lists actually give better results
  - Weight their votes more heavily.
- Tabulate the votes again,
  - Each page's vote has a weight equal to its value as a list.

# Voting by in-links

- The final part of the argument for link analysis
  - Why stop here?
- We have better votes on the right-hand-side of the figure,
  - Use these to get still more refined values for the quality of the lists.
- Refined estimates for the high-value lists,
  - Re-weight the votes of the right-hand-side once again.
- Principle of *Repeated Improvement*.

# Hubs and Authorities

- This suggests a ranking procedure
  - We will call the kinds of pages we were originally seeking: authorities.
  - We will call the high-value lists: hubs.

- For each page $p$ we are trying to estimate its value as a potential authority and as a potential hub:
  - Assign it two numerical scores: $auth(p)$ and $hub(p)$.
  - Initialise both to be 1.

# Hubs and Authorities

- Hubs and Authorities -- Authority Update Rule
  - For each page $p$, update $auth(p)$ to be the sum of the hub scores of all pages that point to it.
- Hubs and Authorities -- Hub Update Rule
  - For each page $p$, update $hub(p)$ to be the sum of the authority scores of all pages that it points to.

# Principle of Repeated Improvement

1. We start with all hub scores and all authority scores equal to 1.

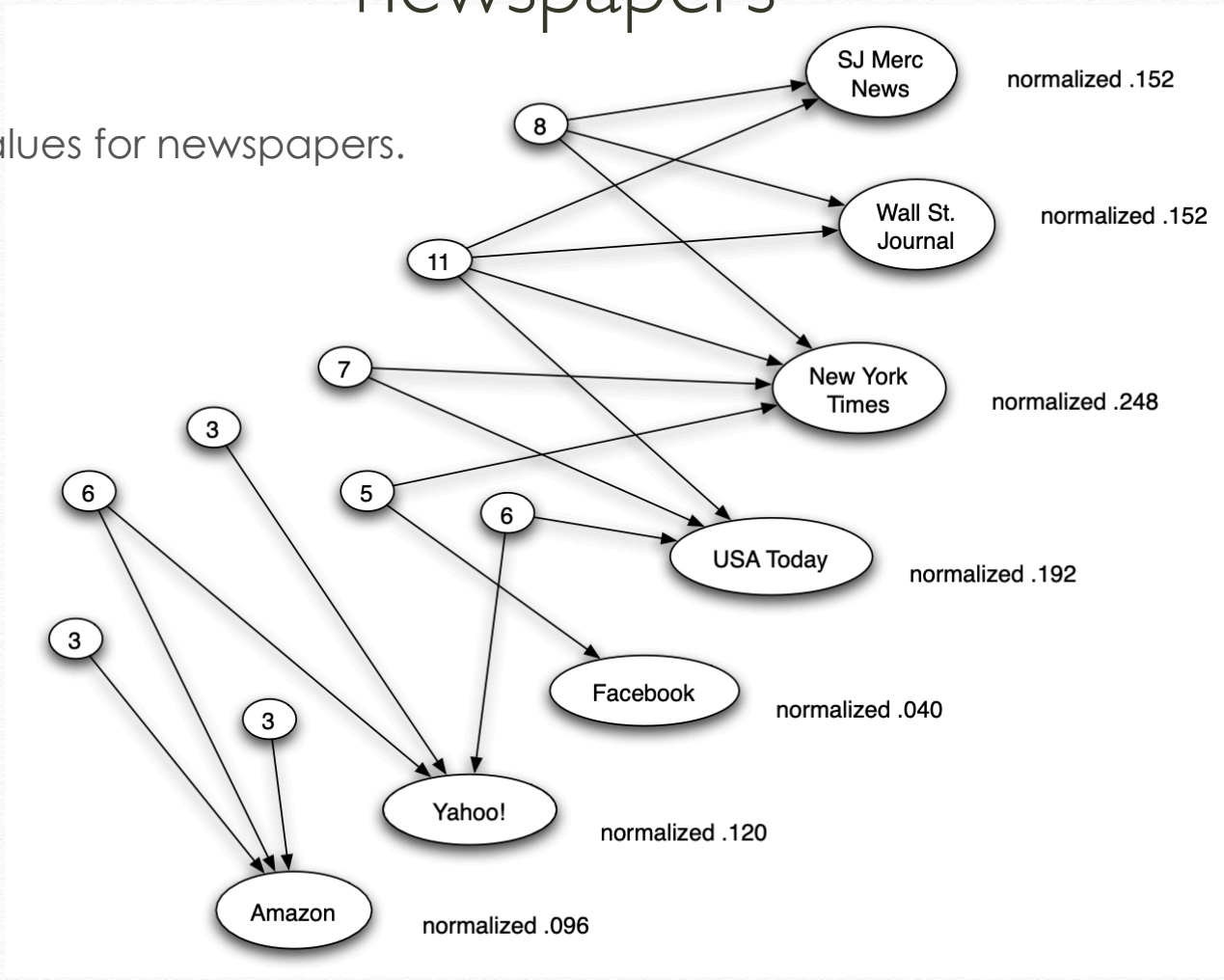
2. We choose a number of steps $k$.

3. We then perform a sequence of $k$ hub-authority updates.
   - First apply the Authority Update Rule to the current set of scores.
   - Then apply the Hub Update Rule to the resulting set of scores.

4. The hub and authority scores may involve numbers that are very large.
   - We only care about their relative sizes, so we can normalize.
     - Divide down each authority score by the sum of all authority scores, and
     - Divide down each hub score by the sum of all hub scores.

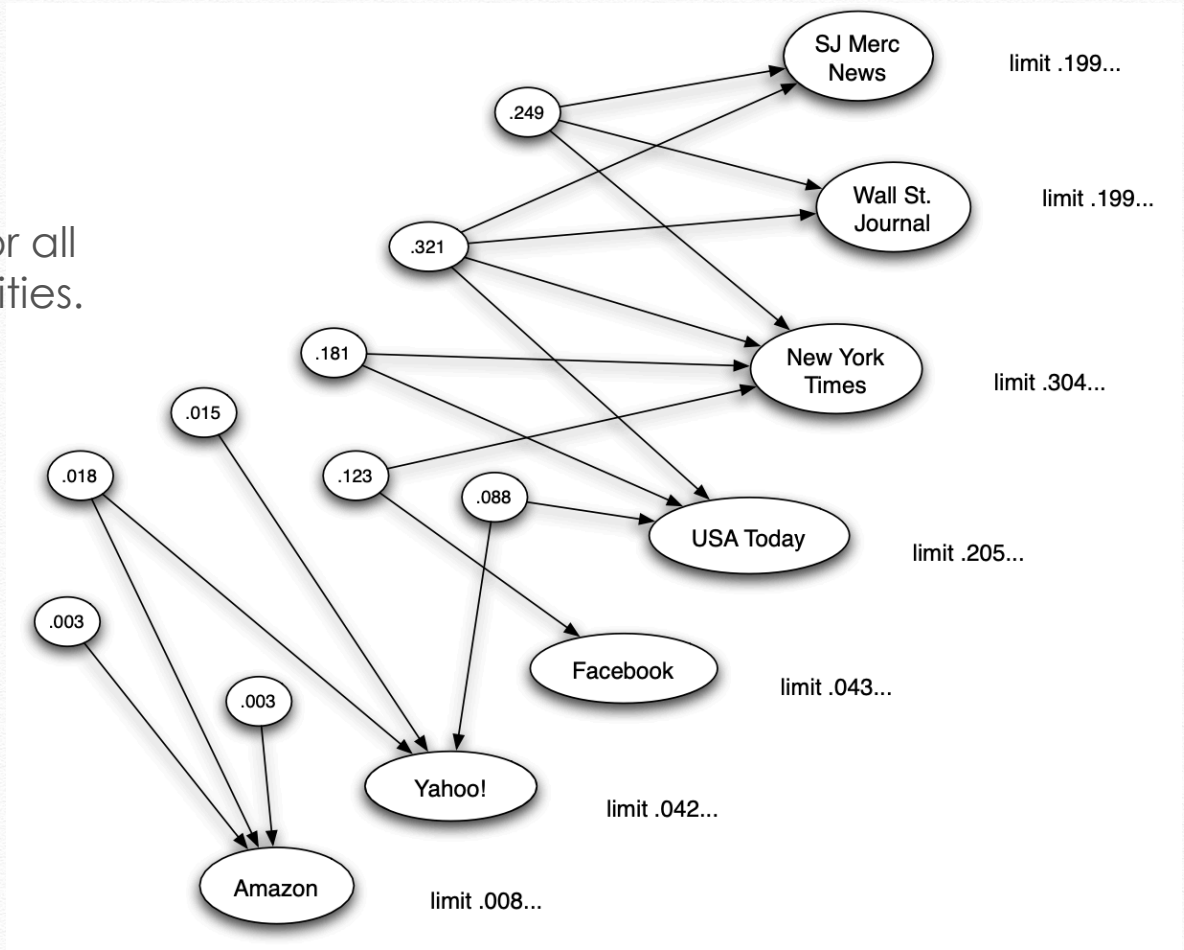
# Counting in-links to pages for the query "newspapers"

Normalised values for newspapers.



# Hubs and Authority Values

Limiting values for all hubs and authorities.



# Hubs and Authorities

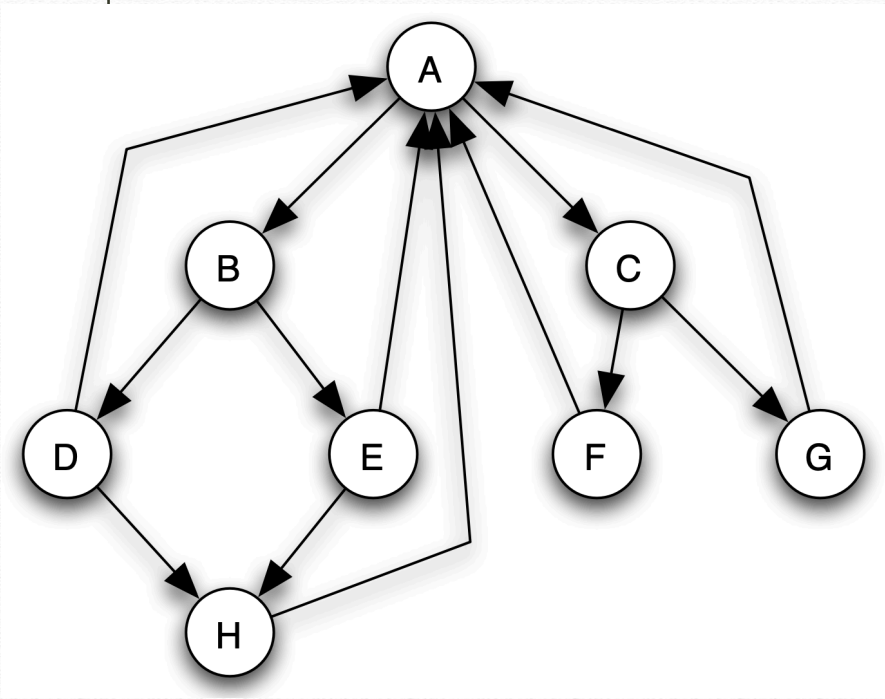
- The normalized values actually converge to limits as $k$ goes to infinity.
- We reach the same limiting values no matter what we choose as the initial hub and authority values.
  - provided only that all of them are positive.

# PageRank

- A page is important if it is cited by other important pages.
- This is often the dominant mode of endorsement
  - academic or governmental pages
  - scientific literature
- Principle of Repeated Improvement
  - The Principle is applied here by having nodes repeatedly pass endorsements across their out-going links
  - with the weight of a node's endorsement based on the current estimate of its PageRank.
- Nodes that are currently viewed as more important get to make stronger endorsements.

# PageRank

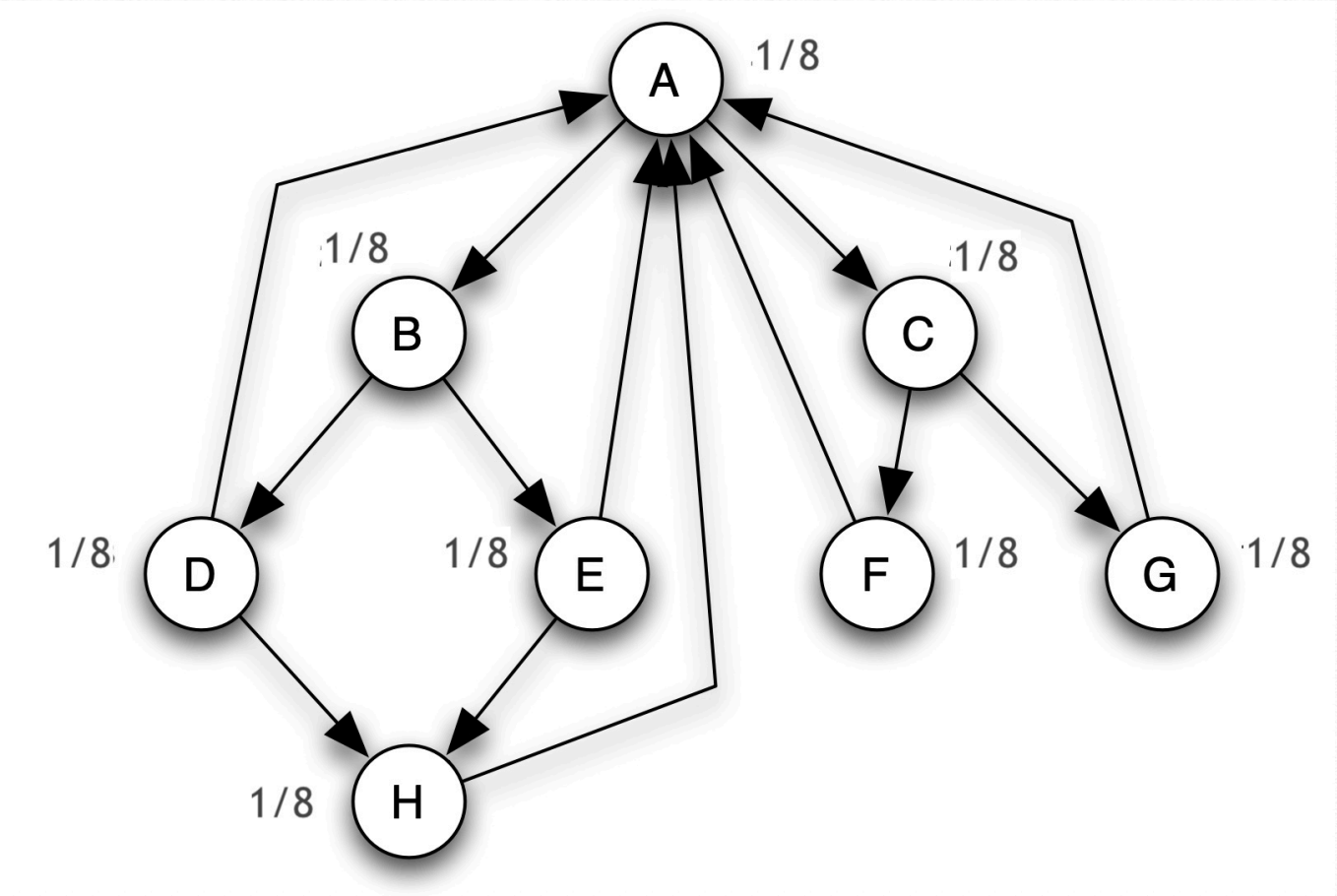
- Intuitively, PageRank as a kind of "fluid" that circulates through the network,
  - Passing from node to node across edges, and pooling at the nodes that are the most important.



# PageRank

- In a network with $n$ nodes, we assign all nodes the same initial PageRank, set to be $1/n$.
- We choose a number of steps $k$.
- We then perform a sequence of $k$ updates to the PageRank values
  - Each page divides its current PageRank equally across its out-going links, and passes these equal shares to the pages it points to.
  - If a page has no out-going links, it passes all its current PageRank to itself.
  - Each page updates its new PageRank to be the sum of the shares it receives.

# PageRank

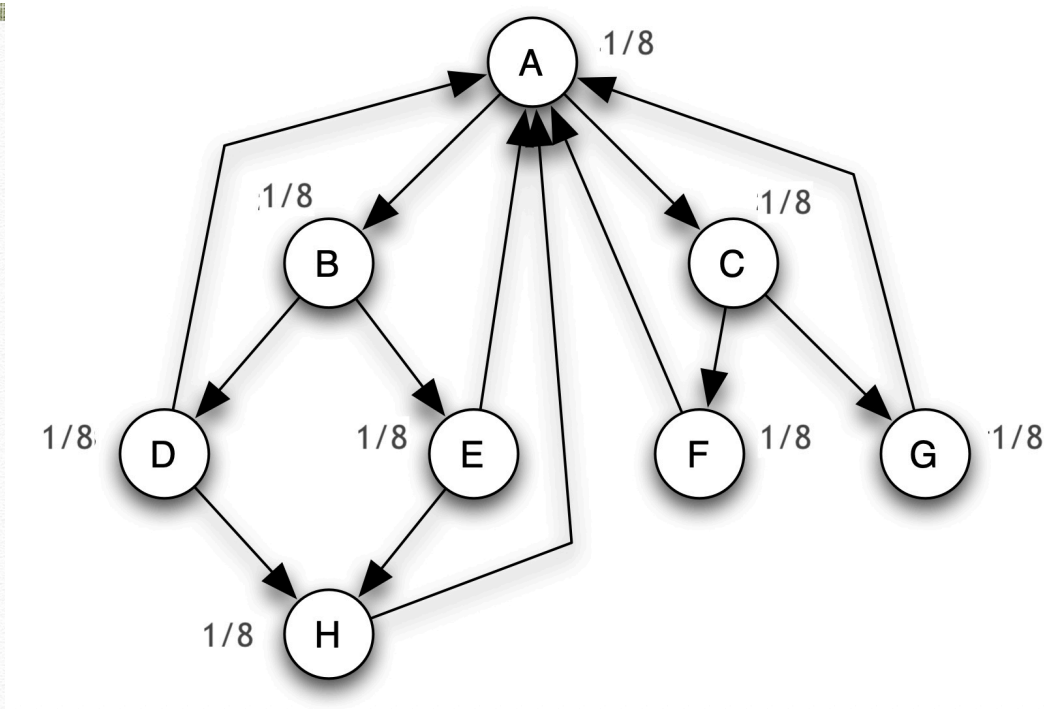



$n = 8$

$1/n = 1/8$

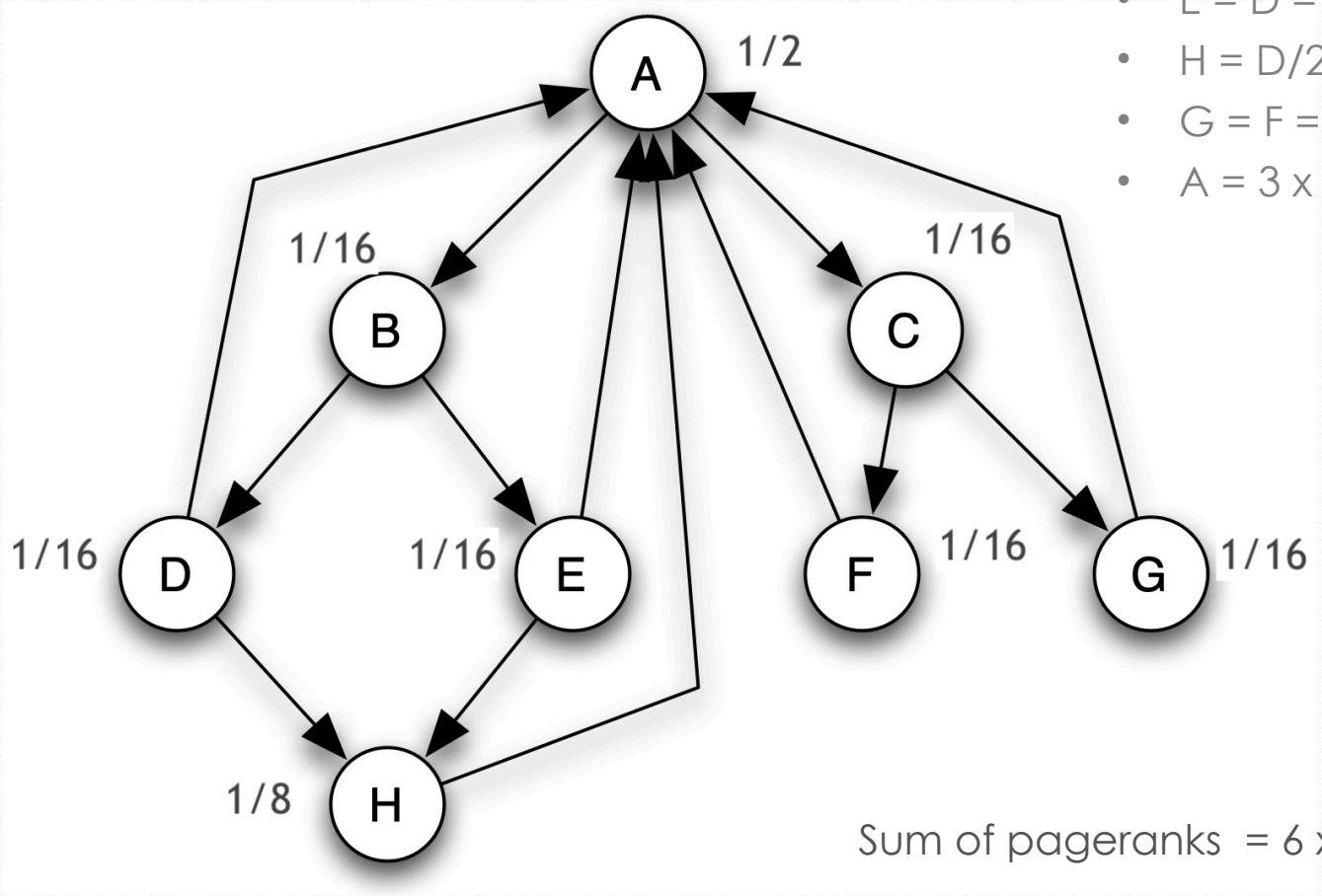
Sum = 1

- B = A/2
- C = A/2
- E = D = B/2
- H = D/2 + E/2
- F = C/2
- G = C/2
- A = F + G + H + D/2 + E/2




- B = 1/8 x 1/2 = 1/16
- C = 1/8 x 1/2 = 1/16
- E = D = 1/8 x 1/2 = 1/16
- H = D/2 + E/2 = 1/16+1/16 = 1/8
- G = F = C/2 = 1/8 x 1/2 = 1/16
- A = 3 x 1/8 + 2 x 1/16 = 8/16 = 1/2

# PageRank

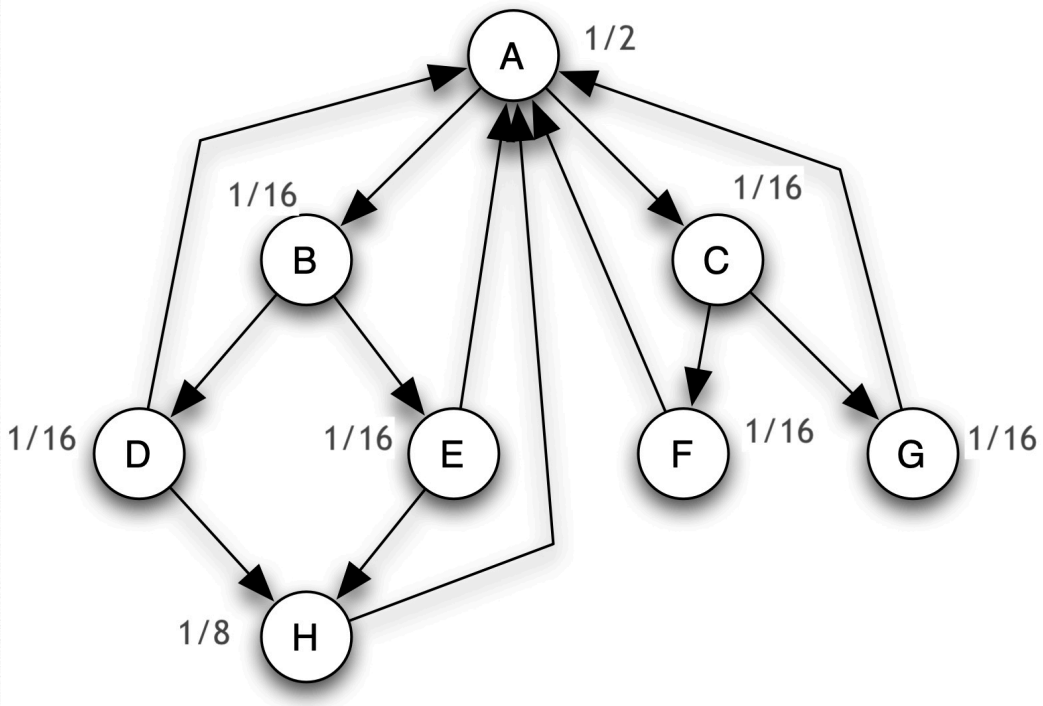



- B = 1/8 x 1/2 = 1/16
- C = 1/8 x 1/2 = 1/16
- E = D = 1/8 x 1/2 = 1/16
- H = D/2 + E/2 = 1/16+1/16 = 1/8
- G = F = C/2 = 1/8 x 1/2 = 1/16
- A = 3 x 1/8 + 2 x 1/16 = 8/16 = 1/2

After 1 iteration:

Sum of pageranks = 6 x 1/16 + 1/8 + 1/2 = 1

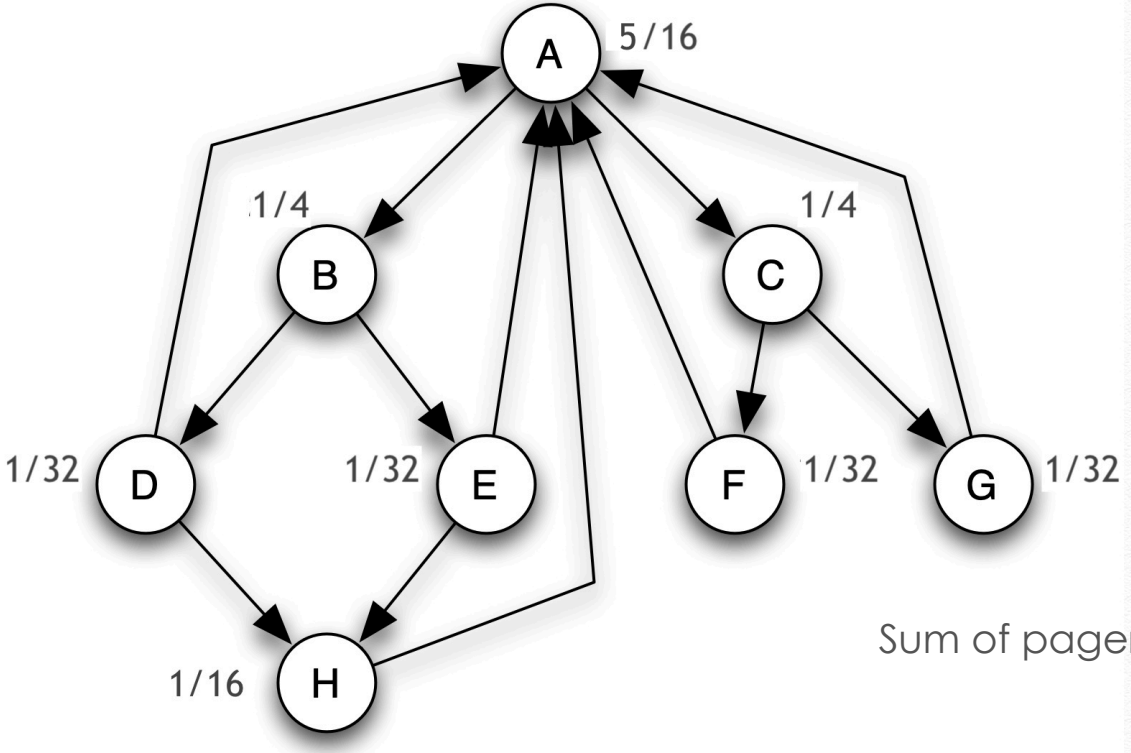
- B = A/2
- C = A/2
- E = D = B/2
- H = D/2 + E/2
- F = C/2
- G = C/2
- A = F + G + H + D/2 + E/2




- B = 1/2 x 1/2 = 1/4
- C = 1/2 x 1/2 = 1/4
- E = D = 1/16 x 1/2 = 1/32
- H = D/2 + E/2 = 1/32+1/32 = 1/16
- G = F = C/2 = 1/16 x 1/2 = 1/32
- A = 1/16 + 1/16 + 1/8 + 1/32 + 1/32 = 5/16

# PageRank

- B = 1/2 x 1/2 = 1/4
- C = 1/2 x 1/2 = 1/4
- E = D = 1/16 x 1/2 = 1/32
- H = D/2 + E/2 = 1/32+1/32 = 1/16
- G = F = C/2 = 1/16 x 1/2 = 1/32
- A = 1/16 + 1/16 + 1/8 + 1/32 + 1/32 = 5/16

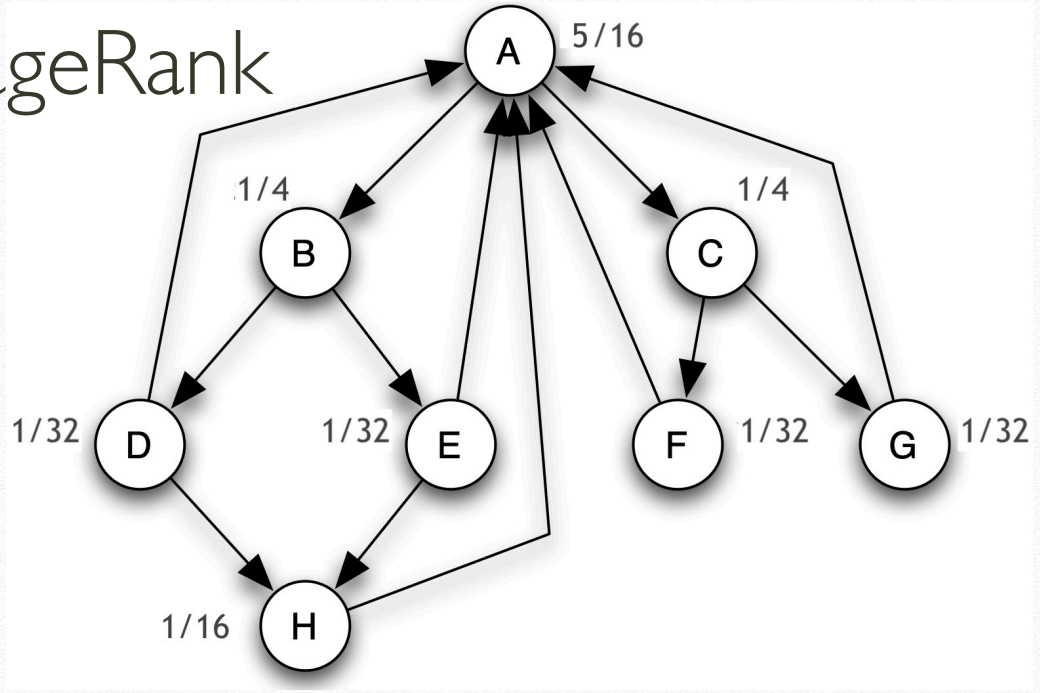



After 2nd iteration:

Sum of pageranks = 4 x 1/32 + 1/16 + 2 x 1/4 + 5/16
= 1/8 + 1/16 + 1/2 + 5/16
= 1/8 + 6/16 + 1/2
= 1

# PageRank

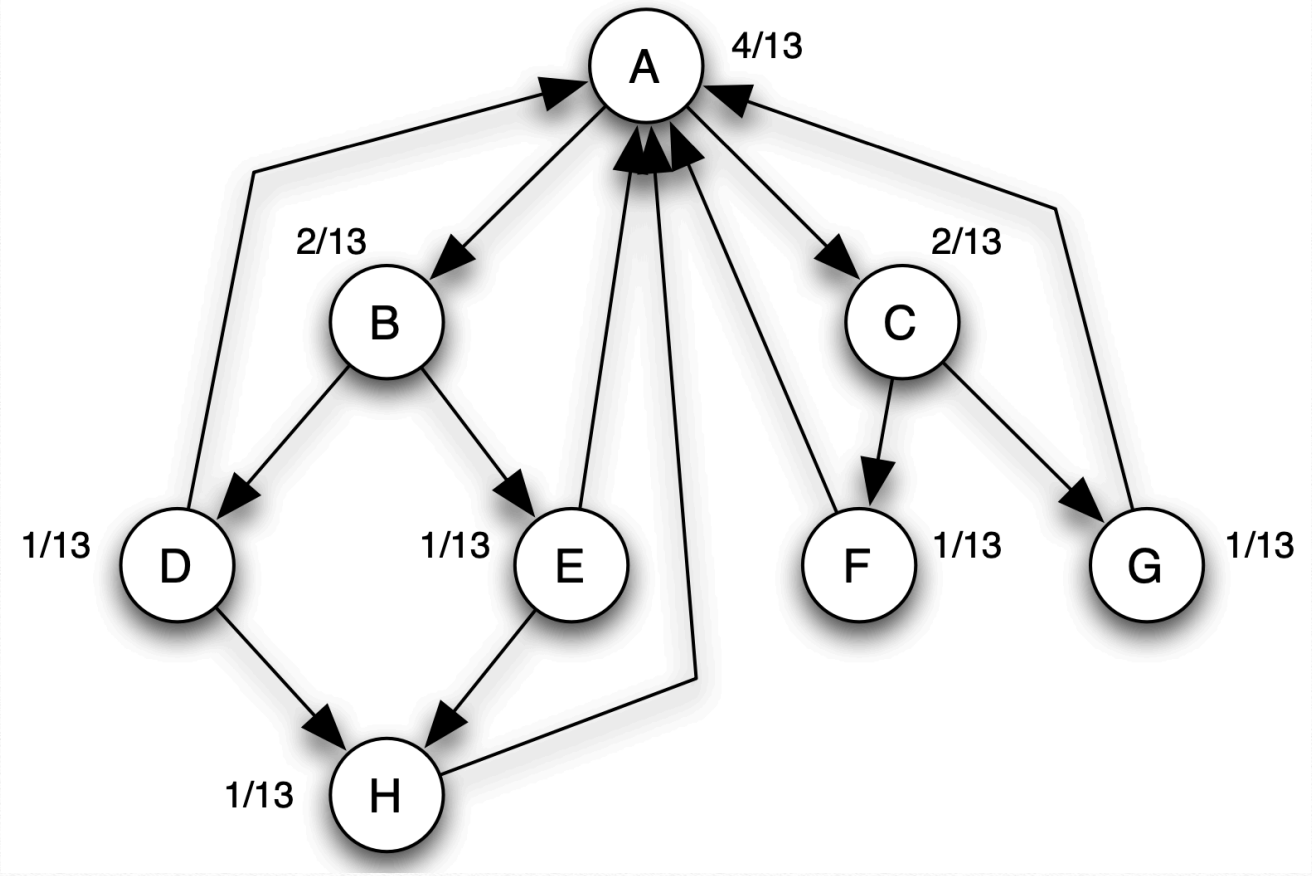
- B = A/2
- C = A/2
- E = D = B/2
- H = D/2 + E/2
- F = C/2
- G = C/2
- A = F + G + H + D/2 + E/2



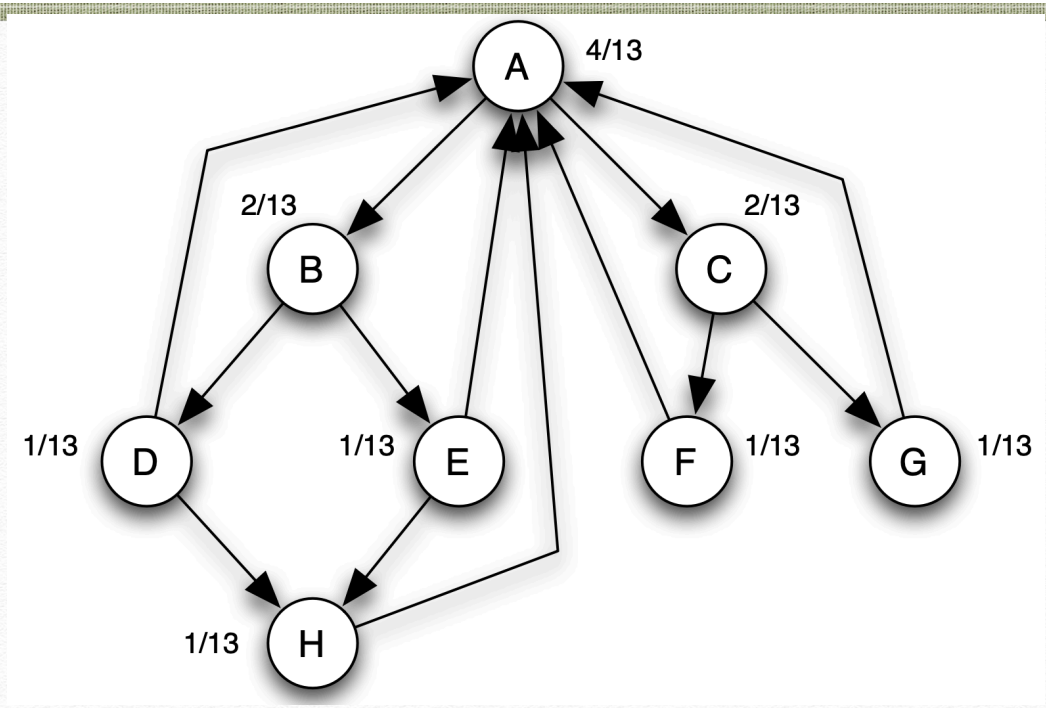

- C = B = 5/16 x 1/2 = 5/32
- E = D = 1/4 x 1/2 = 1/8
- H = D/2 + E/2 = 1/64+1/64 = 1/32
- G = F = C/2 = 1/4 x 1/2 = 1/8
- A = 1/32 + 1/32 + 1/16 + 1/64 + 1/64 = 5/32

Sum of pageranks = 5/32 x 3 + 4 x 1/8 + 1/32 = 1

# PageRank




Equilibrium Values




- B = A/2
- C = A/2
- E = D = B/2
- H = D/2 + E/2
- F = C/2
- G = C/2
- A = F + G + H + D/2 + E/2

- C = B = 4/13 x 1/2 = 2/13
- E = D = 2/13 x 1/2 = 1/13
- H = D/2 + E/2 = 1/26+1/26 = 1/13
- G = F = C/2 = 2/13 x 1/2 = 1/13
- A = 1/13 + 1/13 + 1/13 + 1/26 + 1/26 = 4/13

Sum of pageranks = 4/13 + 5 x 1/13 + 2 x 2/13 = 1

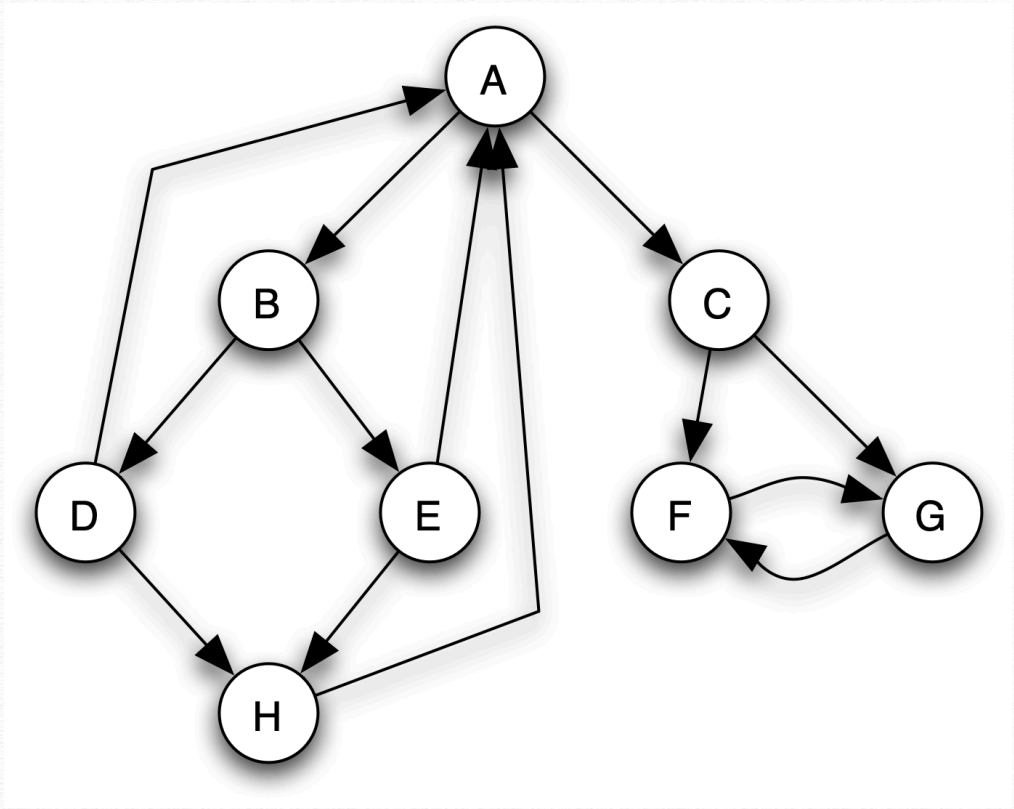
# Equilibrium Values of PageRank

- PageRank values of all nodes converge to limiting values as the number of update steps $k \to \infty$
  - *except in certain degenerate special cases*
    *(we won't get into this now!)*
- PageRank is never created nor destroyed, just moved around from one node to another.
  - PageRank is conserved throughout the computation.
- Equilibrium: if we take the limiting PageRank values and do another iteration, the values do not change.

# Equilibrium Values of PageRank

- *If the network is strongly connected,*
  - *then there is a unique set of equilibrium values,*
  - *and so whenever the limiting PageRank values exist,*
  - *they are the only values that satisfy this equilibrium.*

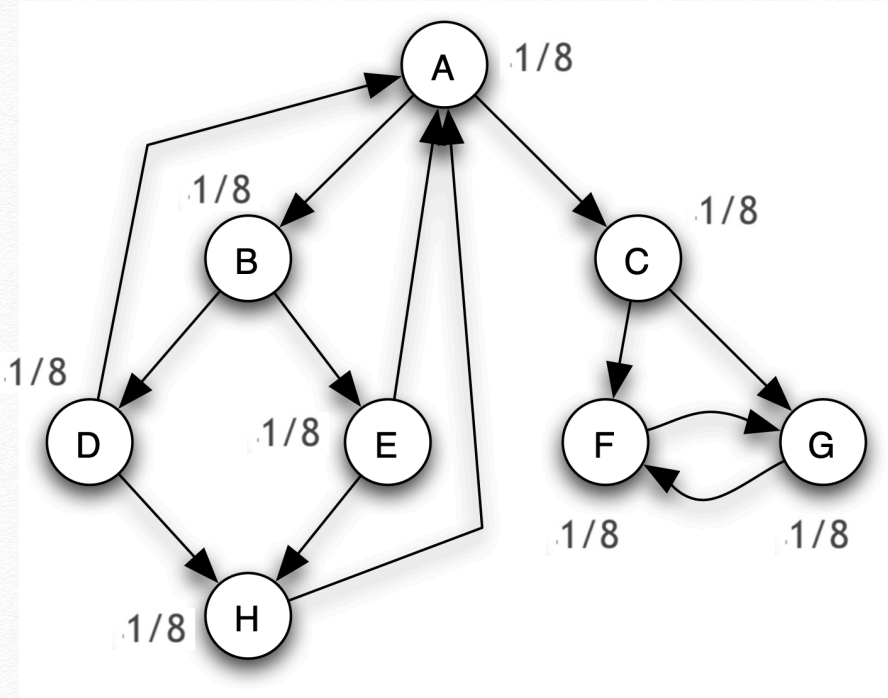
# Scaling PageRank




- *Suppose we have a network like this, then what happens?*

# PageRank

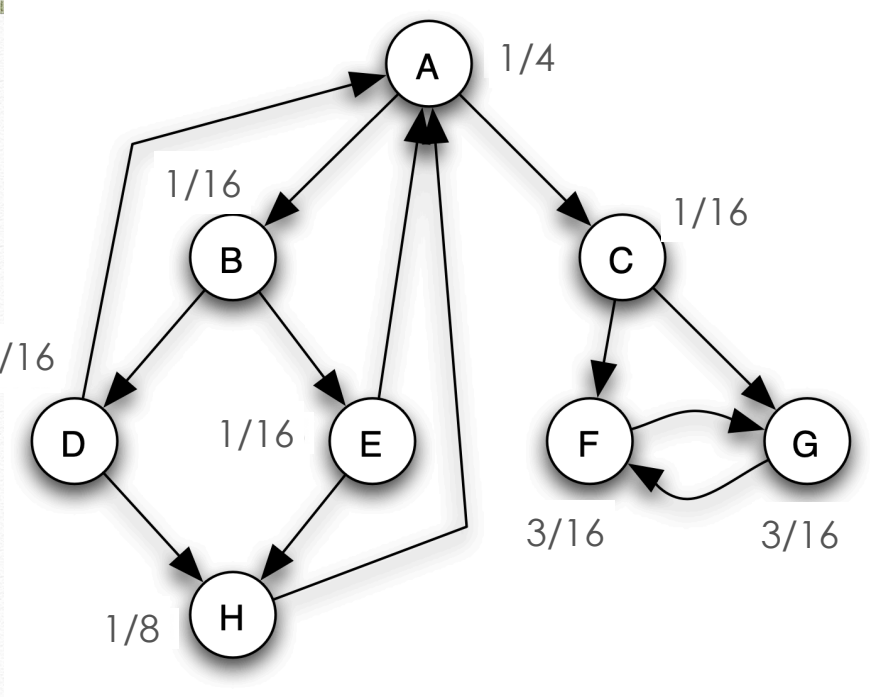
- B = A/2
- C = A/2
- E = D = B/2
- H = D/2 + E/2
- F = C/2 + G
- G = C/2 + F
- A = H + D/2 + E/2




- B = 1/8 x 1/2 = 1/16
- C = 1/8 x 1/2 = 1/16
- E = D = 1/8 x 1/2 = 1/16
- H = D/2 + E/2 = 1/16+1/16 = 1/8
- G = C/2 + F = 1/8 x 1/2 + 1/8 = 3/16
- F = C/2 + G = 1/8 x 1/2 + 1/8 = 3/16
- A = 1/8 + 2 x 1/16 = 2/8 = 1/4

# PageRank

- B = A/2
- C = A/2
- E = D = B/2
- H = D/2 + E/2
- F = C/2 + G
- G = C/2 + F
- A = H + D/2 + E/2




- C = B = 1/4 x 1/2 = 1/8
- E = D = 1/16 x 1/2 = 1/32
- H = D/2 + E/2 = 1/32+1/32 = 1/16
- G = C/2 + F = 1/32 + 3/16 = 7/32
- F = C/2 + G = 1/32 + 3/16 = 7/32
- A = 1/8 + 2 x 1/32 = 3/16

# Scaling PageRank

- Eventually, all the pagerank values will flow to F and G, and the rest of them will become zero.
- This is a problem for real-world networks which have a set of nodes from where there are no paths going out of the set.
  (Where have you seen this?)
- How do we get around this problem?

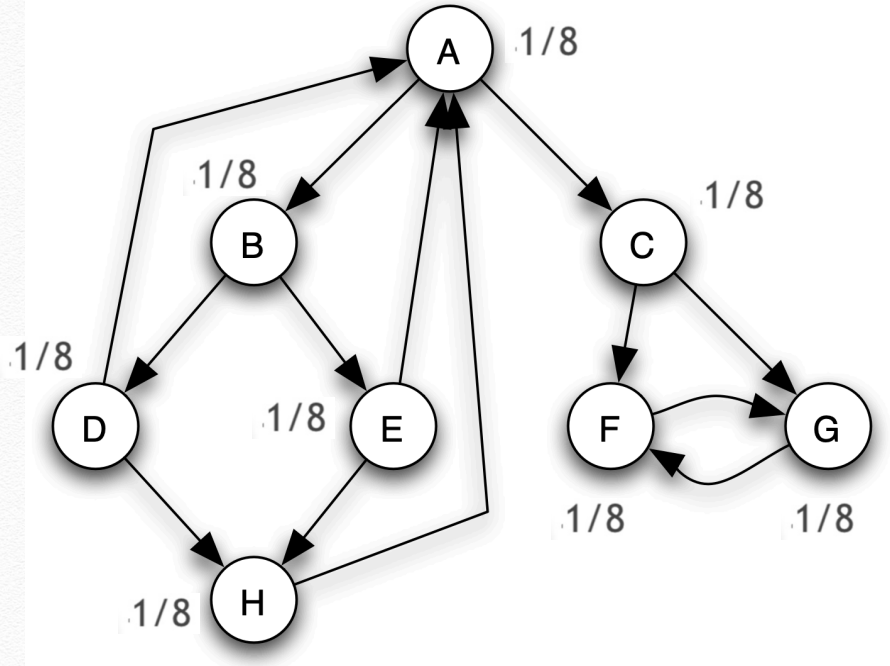
# Scaling PageRank

- We can choose a scaling factor $s$ strictly between 0 and 1.
- First we update the values. Then we scale the values down by $s$.
- The total pagerank in the network thus shrinks by $s$.
- The remaining $(1 - s)$ is divided equally among all the nodes, so each node gets $(1 - s)/n$.
- This approach also converges to a limiting value.
- In practise, $s$ is chosen to be 0.8 or 0.9.

# Scaling PageRank

- $s = 0.80 = 4/5$

- C = B = A/2
- E = D = B/2
- H = D/2 + E/2
- F(G) = C/2 + G(F)
- A = H + D/2 + E/2



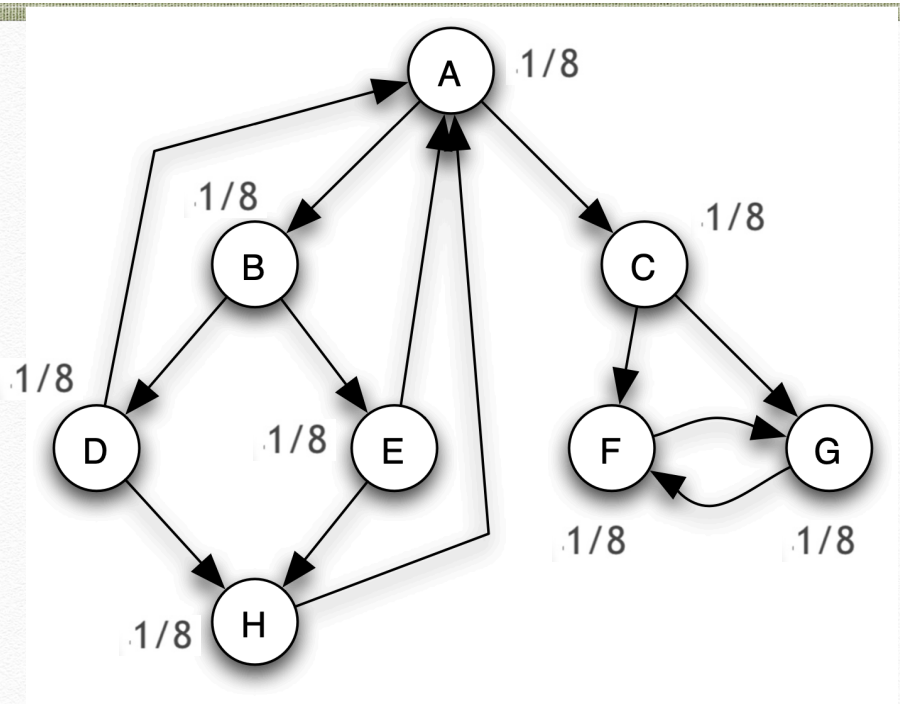

- Scaling down by $s$
- C = B = 1/8 x 1/2 = 1/16 x 4/5 = 4/80
- E = D = 1/8 x 1/2 = 1/16 x 4/5 = 4/80
- H = D/2 + E/2 = 1/16+1/16 = 1/8 x 4/5 = 4/40 = 8/80
- G (F) = C/2 + F(G) = 1/8 x 1/2 + 1/8 = 3/16 x 4/5 = 12/80
- A = 1/8 + 2 x 1/16 = 2/8 = 1/4 x 4/5 = 1/5 = 16/80

- Sum = 2 x 4/80 + 2 x 4/80 + 8/80 + 2 x 12/80 + 16/80 = 64/80 = 0.8

# Scaling PageRank

- $(1 - s) = 1/5$

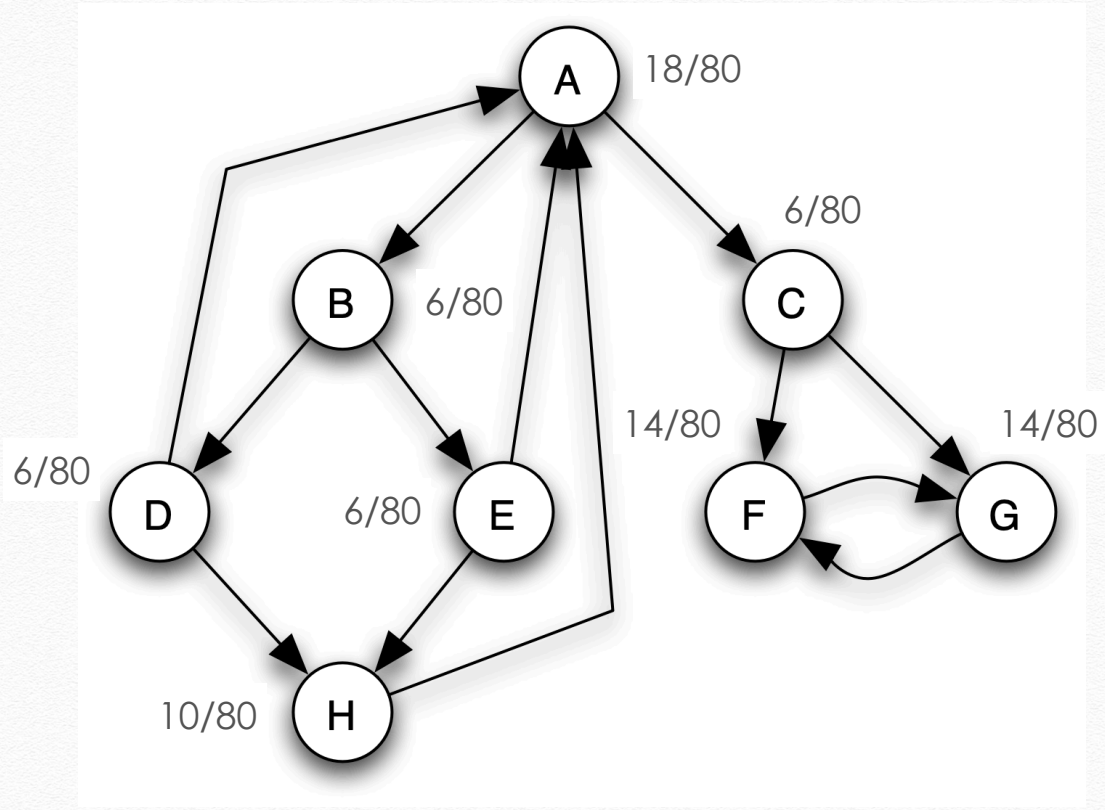
- $C = B = A/2$
- $E = D = B/2$
- $H = D/2 + E/2$
- $F(G) = C/2 + G(F)$
- $A = H + D/2 + E/2$




- Add $(1 - s)/n = 1/(5 \times 8)$ to every node.
- C = B = 1/8 x 1/2 = 1/16 x 4/5 = 4/80 + 1/40 = 6/80
- E = D = 1/8 x 1/2 = 1/16 x 4/5 = 4/80 + 1/40 = 6/80
- H = D/2 + E/2 = 1/16+1/16 = 1/8 x 4/5 = 4/40 = 8/80 + 1/40 = 10/80
- G (F) = C/2 + F(G) = 1/8 x 1/2 + 1/8 = 3/16 x 4/5 = 12/80 + 1/40 = 14/80
- A = 1/8 + 2 x 1/16 = 2/8 = 1/4 x 4/5 = 1/5 = 16/80 + 1/40 = 18/80

- Sum = 2 x 6/80 + 2 x 6/80 + 10/80 + 2 x 14/80 + 18/80 = 80/80 = 1.

# Scaling PageRank




- C = B = A/2 x $s$ + (1 — $s$)/8
- E = D = B/2 x $s$ + (1 — $s$)/8
- H = (D/2 + E/2) x $s$ + (1 — $s$)/8
- F = (C/2 + G) x $s$ + (1 — $s$)/8
- A = (H + D/2 + E/2) x $s$ + (1 — $s$)/8

Ahmedabad University

# Thank you!

Amit A. Nanavati
Ahmedabad University