MapReduce for Relational Structure

Mapreduce Examples

Example: Who Links to Me

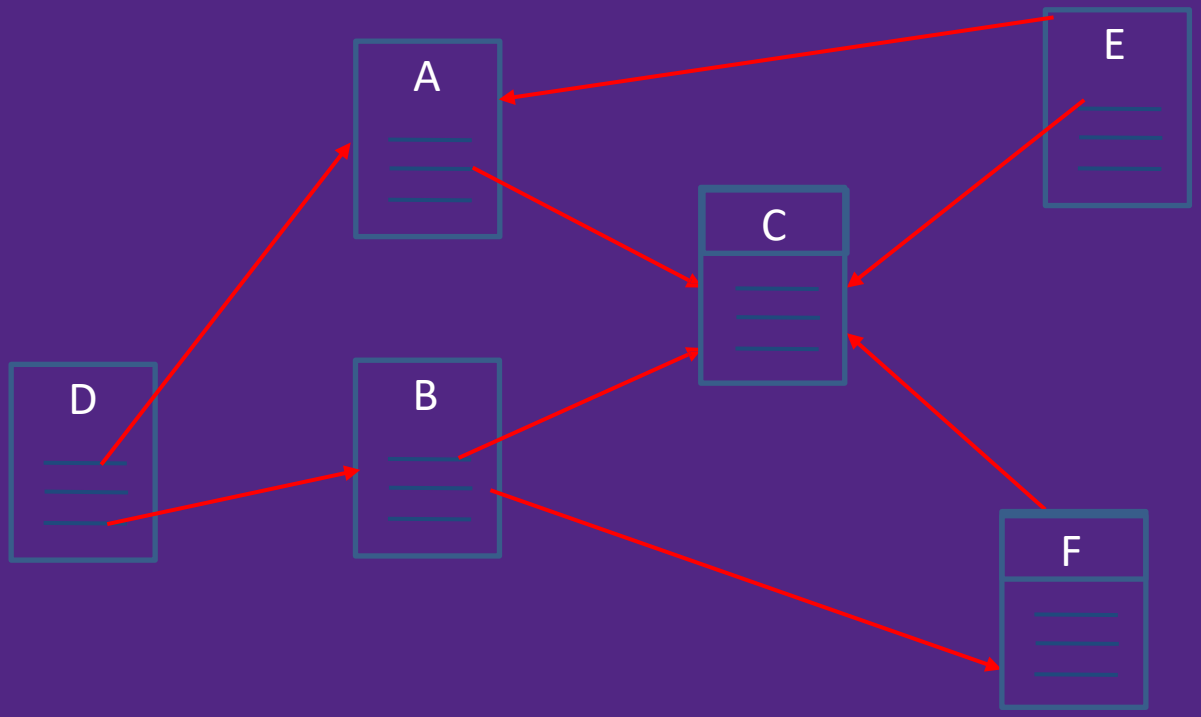
MapReduce reminder

- map function:
  - map(k1,v1) → list(k2,v2)

- reduce function
  - reduce(k2, list (v2)) → list((k3, v3))

Example: Who Links To Me

- Each source page has links to target pages
- Find (target, list (sources)) for all targets



Example: Who Maps to Me

- Example:
  - www.csd.uwo.ca
- We want URLs of websites that link to the above URL, e.g.,
  - https://www.uwo.ca/sci/departments/index.html

Example: Who Links to Me

- Inputs:
  - Web pages
- Example:
  - <!DOCTYPE html>
  - <html>
  - <body>

  - <h2>HTML Links</h2>
  - HTML links are defined with the a tag:

  - This is a link

  - </body>
  - </html>

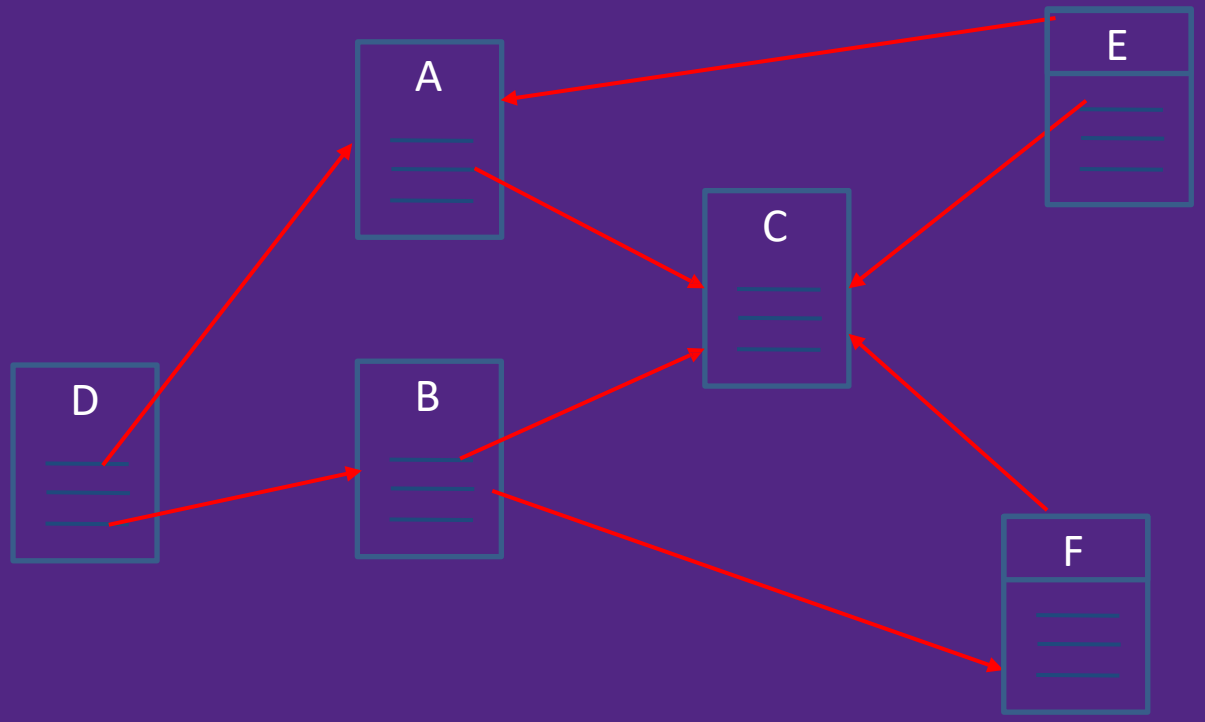
Example: Who Links to Me

- Input: The input are web pages represented by the URL of the web page and the content of the web page.
  - Represented as (URL, web page contents)
  - We are interested in the URLs found in the contents of the page

- Output: (URL, list-of-URLs)
  - list-of-URLs represents the URLs of web pages that that have a link to URL

Example: Who Links To Me

- Input: (A,C), (B,C), (B,F), (D,A), (D,B), (E,A), (E,C), (F,C)
- Output: (A,{D,E}), (B, {D}), (C, {A,B,E,F}), (D,{}), (E,{}), (F,{B})



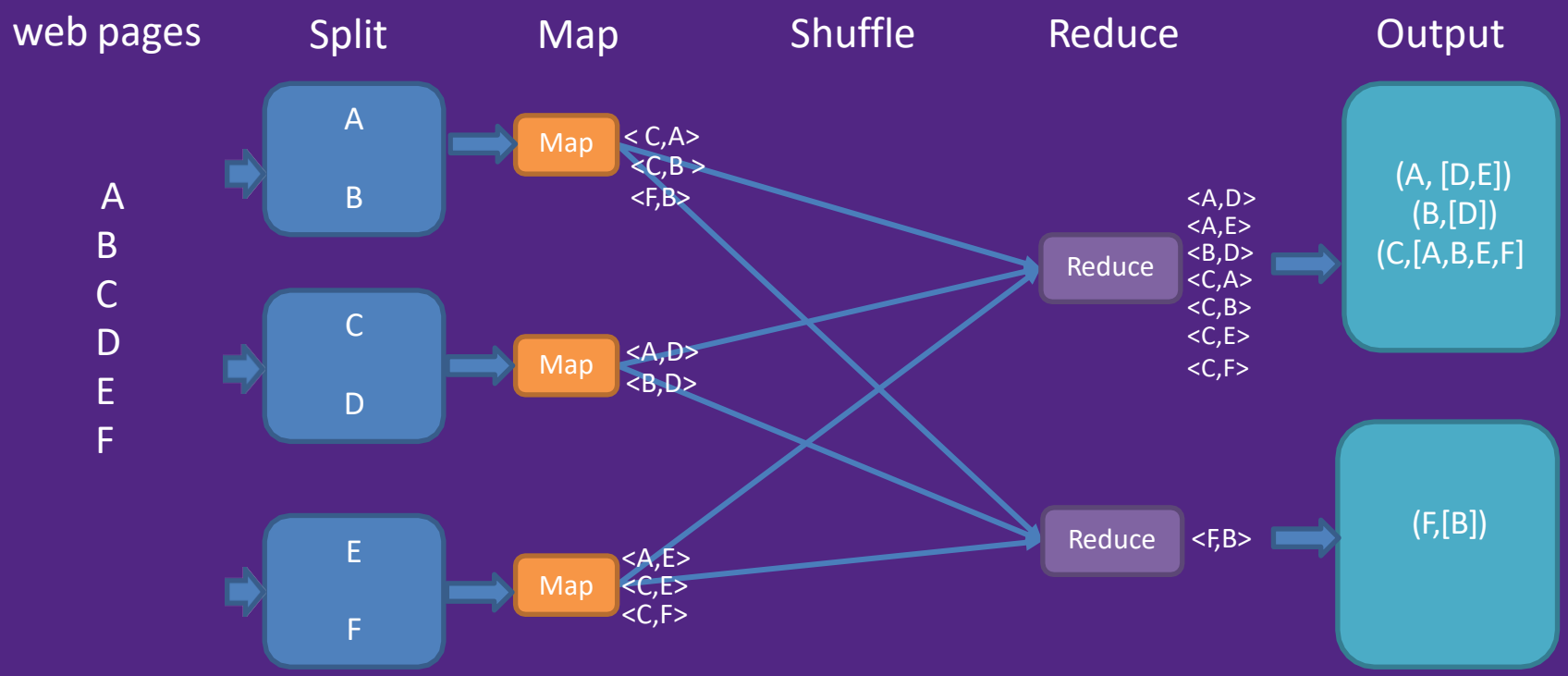
Example: Who Maps to Me

- Map
  - Input: a source page S
  - Output: Pairs (T,S) for every link T in S.
- Example:
  - Assume URL is D
  - D links to A and B
  - This means that for D we emit two pairs: (A,D) and (B,D)

Example: Who Maps to Me

- Reduce:
  - Input: Many pairs (T,S) from the mapper
  - Emits (T, list(S1, S2, …, Sk))
- Example:
  - Web page C is linked to from A,B,E,F. This means that in the map phase the output includes
    - <C,A>, <C,B> <C,E>,<C,F>
  - In the reduce phase this becomes <C, [A,B,E,F]>

Example: Who Maps To Me



Keys with A,B,C are sent to the first reducer
Keys with D,E,F are sent to the second reducer

PageRank

Introduction

- PageRank is what Google used (maybe still?) to determine a page's importance
  - Named after one of the founders of Google: Larry Page
- It's one of many factors used to determine which pages appear in search results.
- PageRank was developed as part of a research project about a new kind of search engine
  - Project was started in 1995
- In 1998 Google was formed

Random Surfer Model

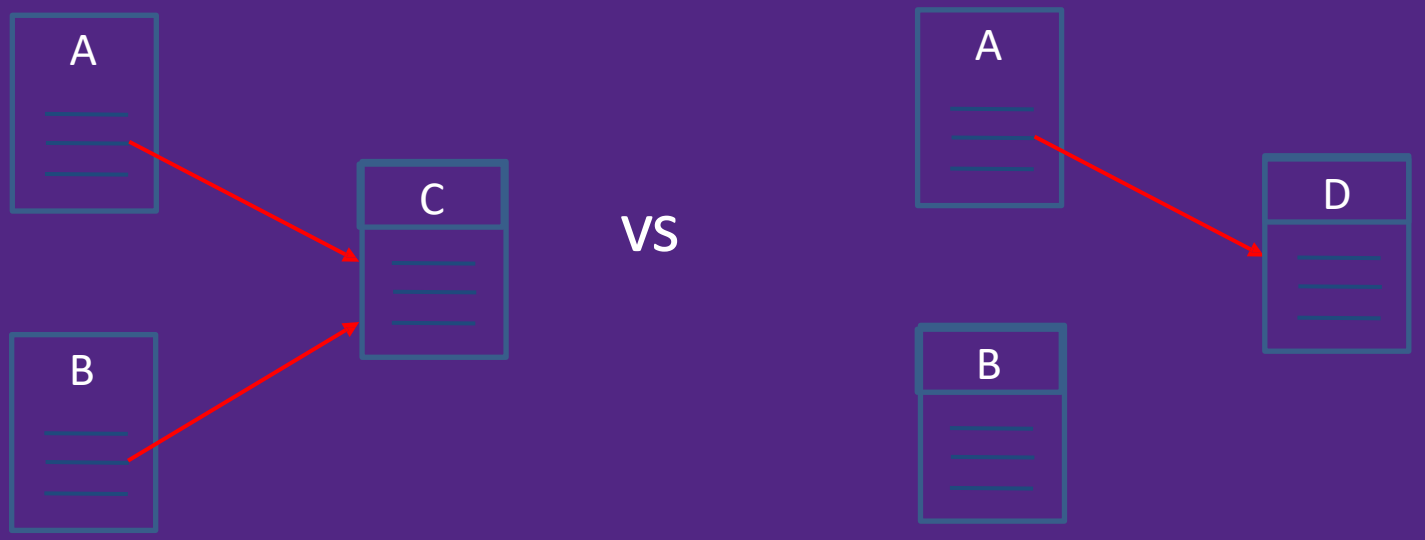
- A random surfer who is given a web page at random and keeps clicking on links, never hitting "back",
  - If it returns to a given web page the surfer may start on another random page.
- The probability that the random surfer clicks on a link is based on the number of links on that page
- The probability that the random surfer visits a page is its PageRank.

Ranking

- The goal of ranking web pages is to get a measure of how popular a web page is
  - Based on the number of pages that are linked to it.
  - Or if there are some pages that point to it, and have a high PageRank.

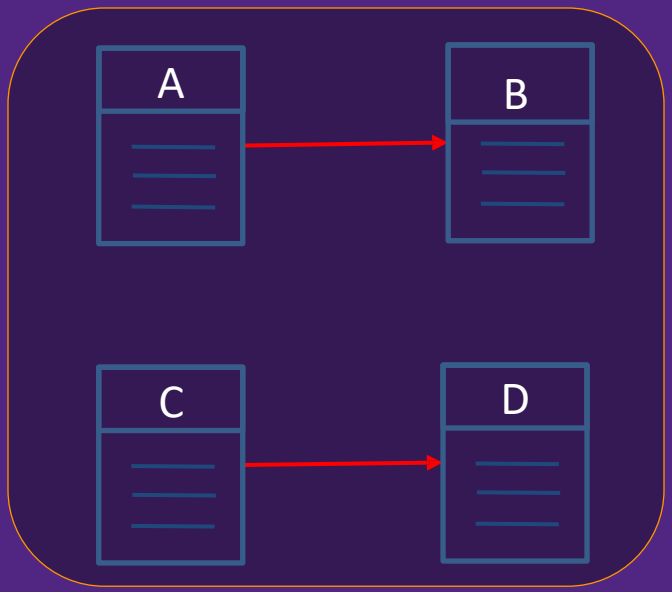
- A page can have a high PageRank
  - If there are many pages that point to it
  - Or if there are some pages that point to it, and have a high PageRank.

View a Link as a Recommendation



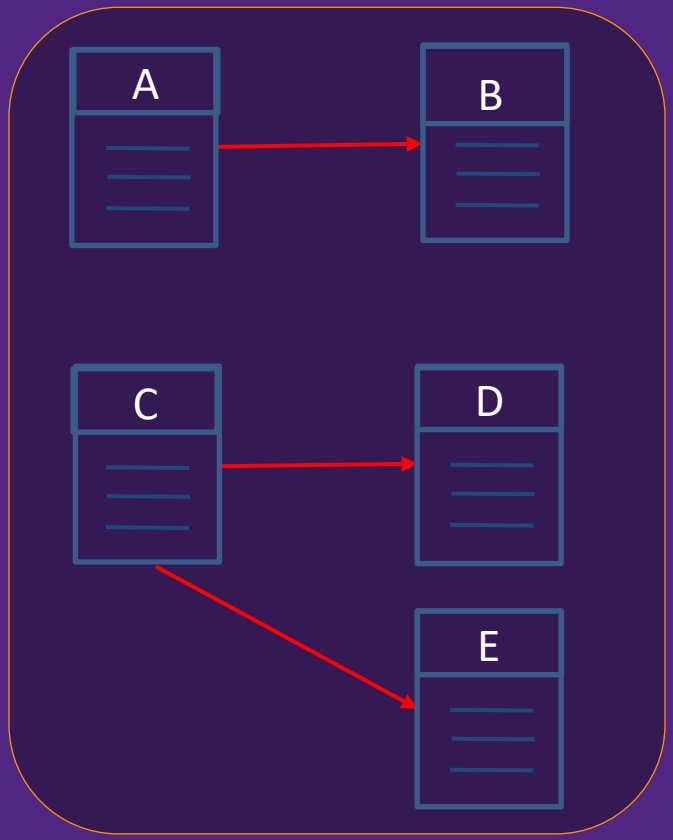
- View a link as a recommendation or a vote
  - Interpret a link from page A to page C as a "vote"
- The more in-links that a web page has the better
  - The page is considered more important
- However, it is relatively easy for someone to inflate their web page's links by creating web pages that point to it.

View a Link as a Recommendation



- Not just about incoming links but also the importance of the webpages that the incoming links are from
- What if page A has 10,000 inlinks and C has no inlinks
- Which is more importantt, B or D?

View a Link as a Recommendation



- What if B and D have the same number of inlinks but C has more outlinks?
- Which is more important B or D? (iClicker)
  - Perhaps B since C is making more recommendations

PageRank: Defined

- Pagerank of P_i is denoted by $PR(P_i)$,
- The value of $PR(P_i)$ is the sum of the normalized page ranks of all webpages pointing to P_i

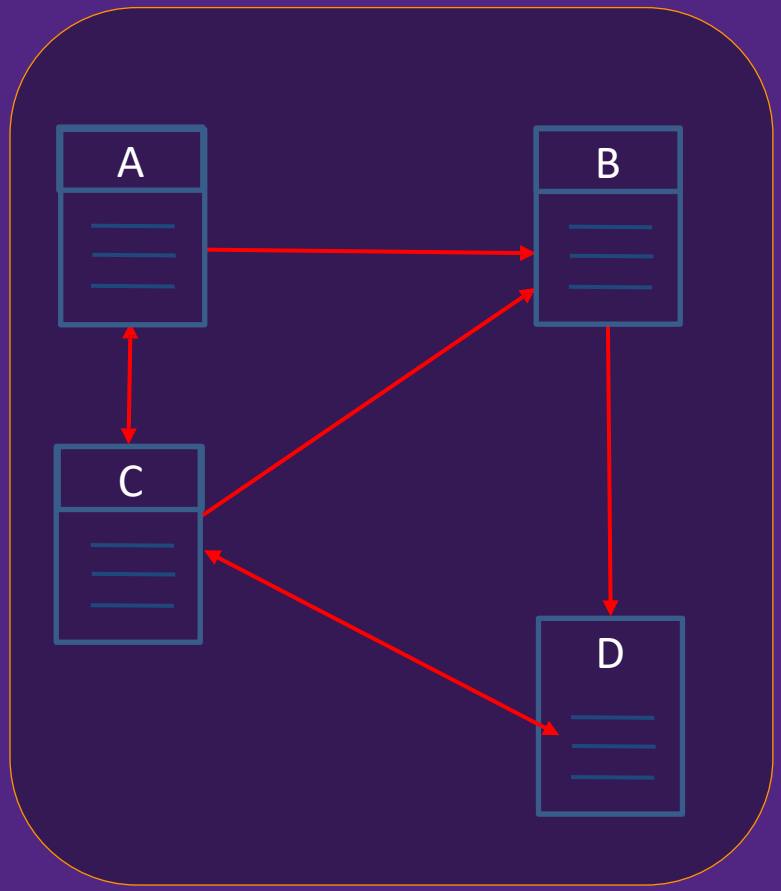
$$PR(P_i) = \sum_{P_i \epsilon inlinks[P_i]} \frac{PR(P_j)}{|Pj|}$$

  - $inlinks[P_i]$: Set of webpages pointing to P_i
  - $|P_j|$: Number of outlinks from page P_j
  - $PR(P_j)/|P_i|$ is the normalized pagerank
    - The pagerank of webpage, P_j, is shared by all webpages P_j points to

Computation of PageRank

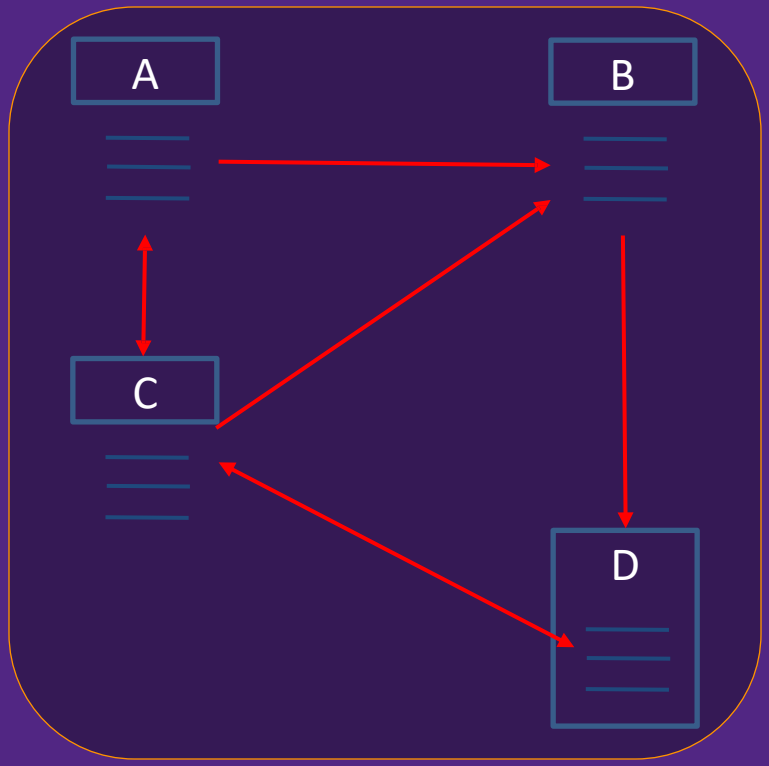
- Problem
  - In the beginning, all pageranks are unknown.
  - How do we determine the first pagerank value?
- Solution
  - Give an initial pagerank toevery webpage
    - Example: 1/N where N is the total number of webpages
  - Perform the calculation of pagerank iteratively
    - Use the pagerank formula to update the pagerank over every webpage
    - Repeat the above step a number of times until the pagerank values are stable (converge)

Example Used to Illustrate PageRank



- A:Inlinks:1 Outlinks:2
- B: Inlinks: 2 Outlinks:1
- C: Inlinks: 2 Outlinks:3
- D: Inlinks: 2 Outlinks:1

Example Used to Illustrate PageRank



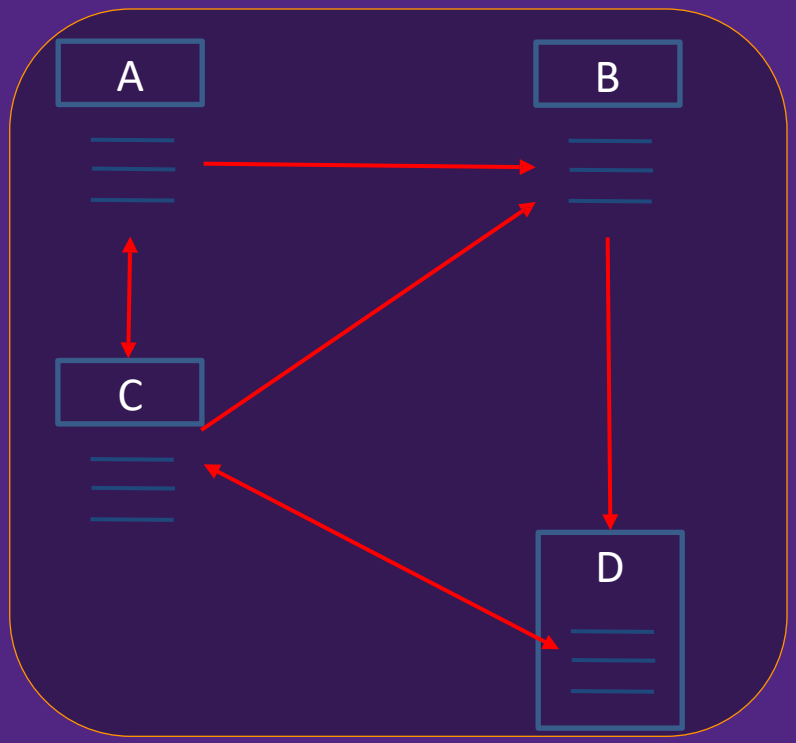
- Assume that initially PR(A), PR(B), PR(C), PR(D)are 0.25
- PR(D) = PR(B)/1 + PR(C)/3
- C has three outgoing links which means that it contributes less to D's PageRank

Computation of PageRank

- Let $r_k(P_i)$ be the PageRank of page P_i at iteration k
  - Starting with $r_0(P_i) = 1/n$ for all pages P_i
- At iteration k+1, the pagerank of every page P_i is updated using the pageranks at iteration k

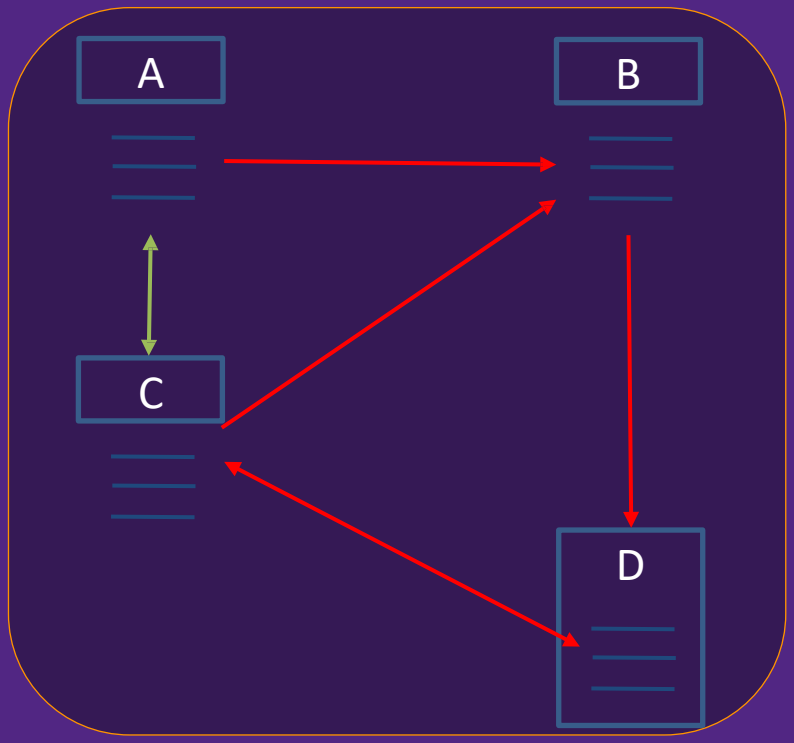
$$r_{k+1}(P_i) = \sum_{P_i \epsilon inlinks[P_i]} \frac{r_k(P_j)}{|Pj|}$$

Example – Iteration 0



- Need an initial pagerank
- Let's assume 1/N where N is the number of pages
- In our example, N is 4

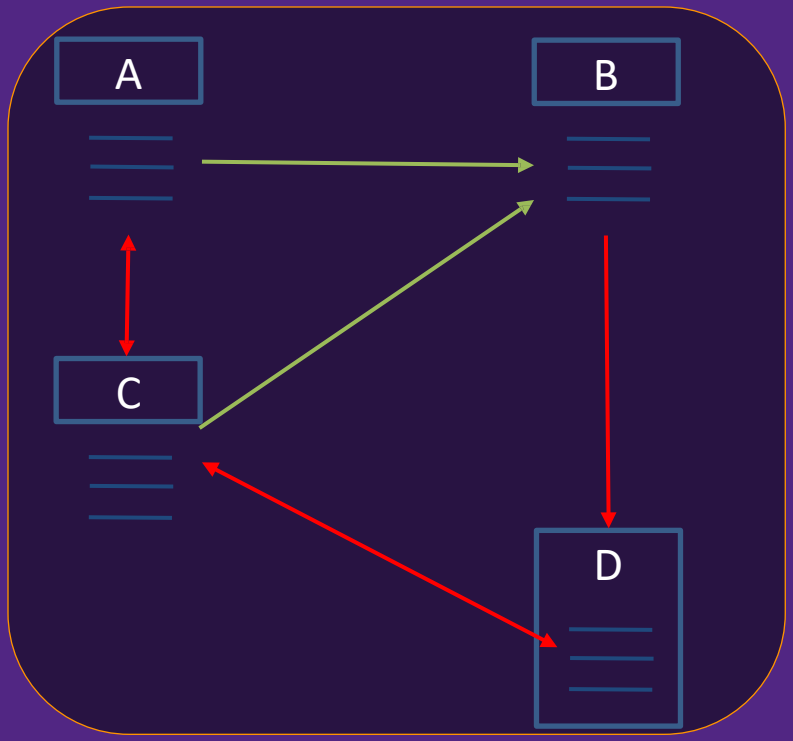
	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/4			
B	1/4			
C	1/4			
D	1/4			



Example – Iteration 1

- Iteration 1 uses pagerank values of the other web pages calculated in iteration 0
- Let's look at page A
- Only C points to A
- C has 3 outlinks
- Thus $PR_1(A)$ is (1/4)/3 =1/12

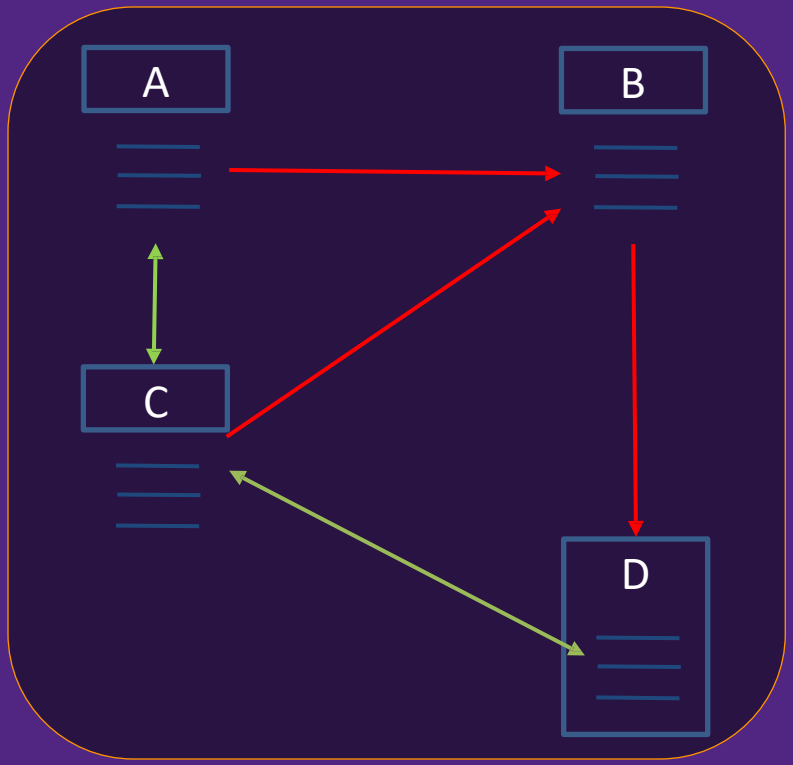
	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/4	1/12		
B	1/4			
C	1/4			
D	1/4			



Example – Iteration 1

- Let's look at page B
- A and C point to B
- A has 2 outlinks
- C has 3 outlinks
- Thus $PR_1(B)$ is (1/4)/2 + (1/4)/3

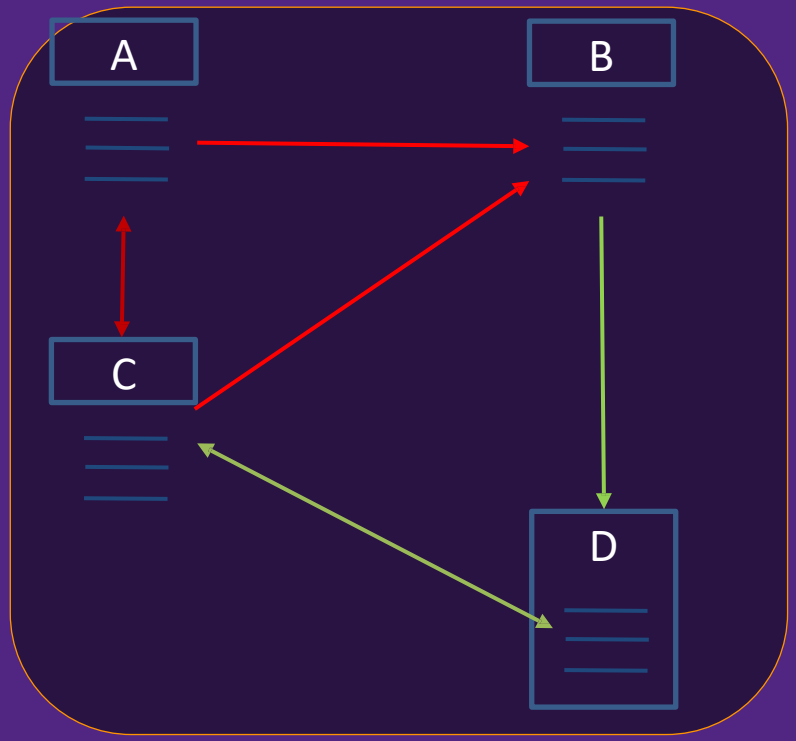
	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/4	1/12		
B	1/4	2.5/12		
C	1/4			
D	1/4			



Example – Iteration 1

- Let's look at page C
- Pages A and D link to it
- Page A has 2 outlinks
- Page D has 1 outlink
- $PR_1(C)$ is (1/4)/2 + (1/4)/1

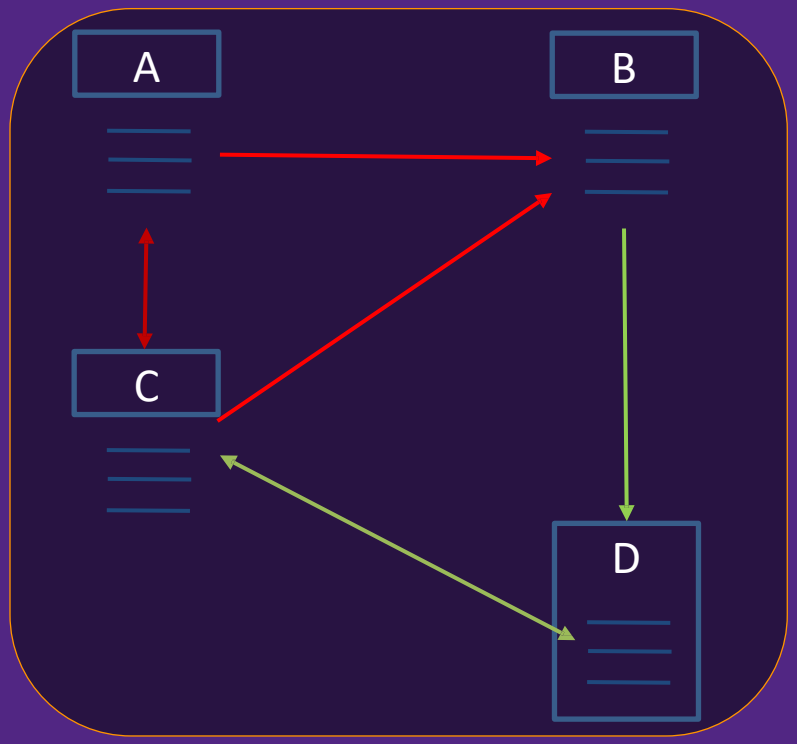
	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/4	1/12		
B	1/4	2.5/12		
C	1/4	4.5/12		
D	1/4			



Example – Iteration 1

- Let's look at page D
- Pages B and C link to it
- Page B has 1 outlink
- Page C has 3 outlinks
- $PR_1(D)$ is (1/4)/1 + (1/4)/3

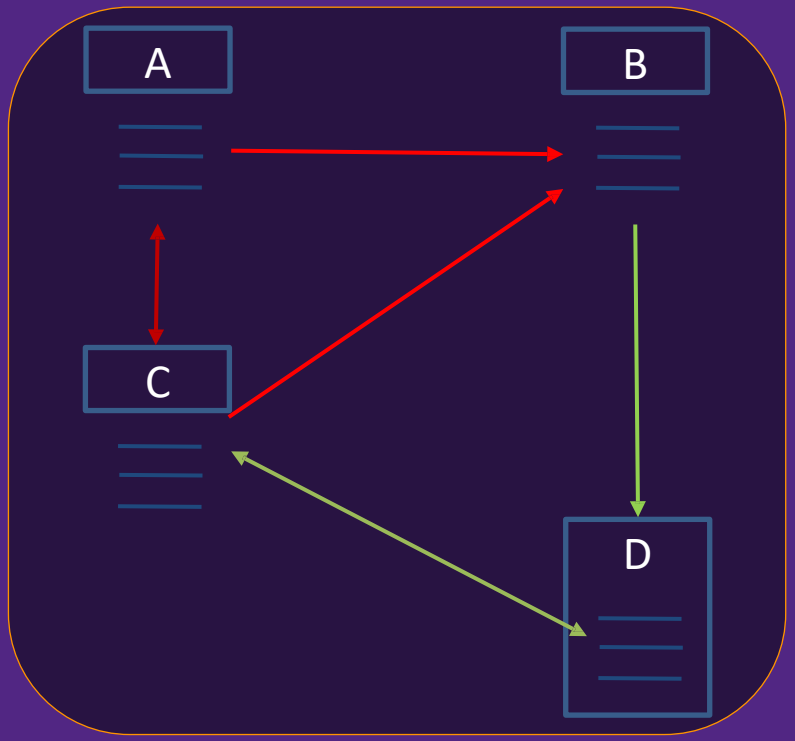
	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/4	1/12		
B	1/4	2.5/12		
C	1/4	4.5/12		
D	1/4	4/12		



End of Iteration 1

- Let's look at page A for iteration 2
- C links to A
- C has 3 outlinks
- $PR_2(A)$ is (4.5/12)/3

	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/4	1/12	1.5/12	
B	1/4	2.5/12		
C	1/4	4.5/12		
D	1/4	4/12		



After Iteration 2

- Each column sums to 1
- Higher values in Iteration 2 result in a higher pagerank
- C is important due to the number of inlinks/one recommendation

	Iteration 0	Iteration 1	Iteration 2	PageRank
A	1/4	1/12	1.5/12	?
B	1/4	2.5/12	2/12	?
C	1/4	4.5/12	4.5/12	?
D	1/4	4/12	4/12	?

How Many Iterations?

- When the differences between values across different iterations are below a given threshold.

Implementing Page Rank Using Map Reduce

- Multiple mapreduce stages are needed
- Output of reducers are fed into the mappers of the next stage

PageRank Using MapReduce

- Mapper:
  - Input:<y, (PR(y), {x_1….x_n})>
    - y is a webpage
    - {x_1…x_n} are the outlinks
  - for j= 1…n: emit $\left(x_j, \frac{PR(y)}{out(y)}\right)$
- Reducer:
  - The reducer receives values from mappers
  - Use the PageRank formula to aggregate values and calculate new PageRank values
- Repeat

End of MapReduce.