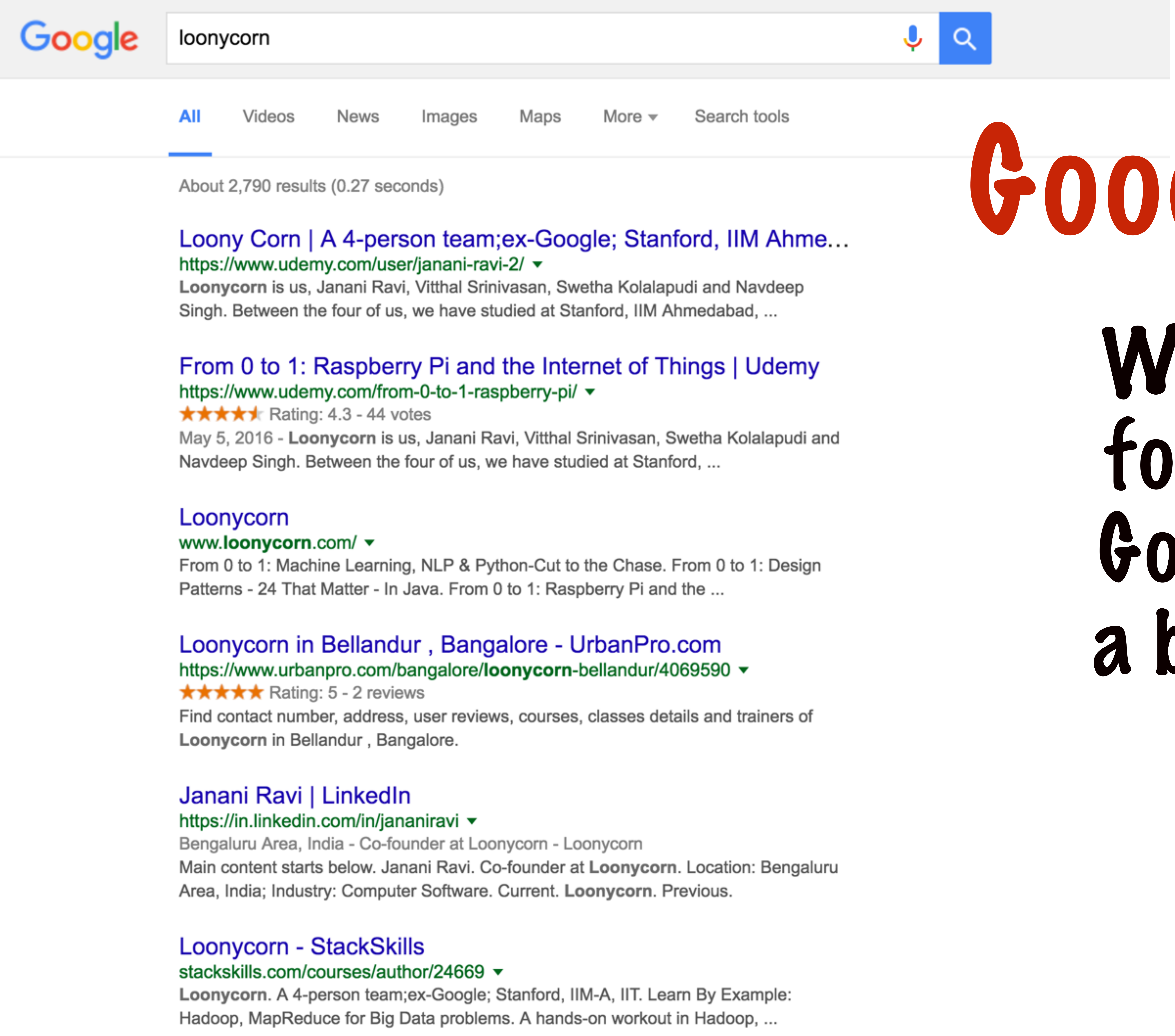
PAGERANK

Have you ever wondered how Google Search works?



Google Search

When you search for something on Google, it returns a bunch of results

Google Search

loonycorn

All Videos News Images Maps More ▾ Search to...

About 2,790 results (0.27 seconds)

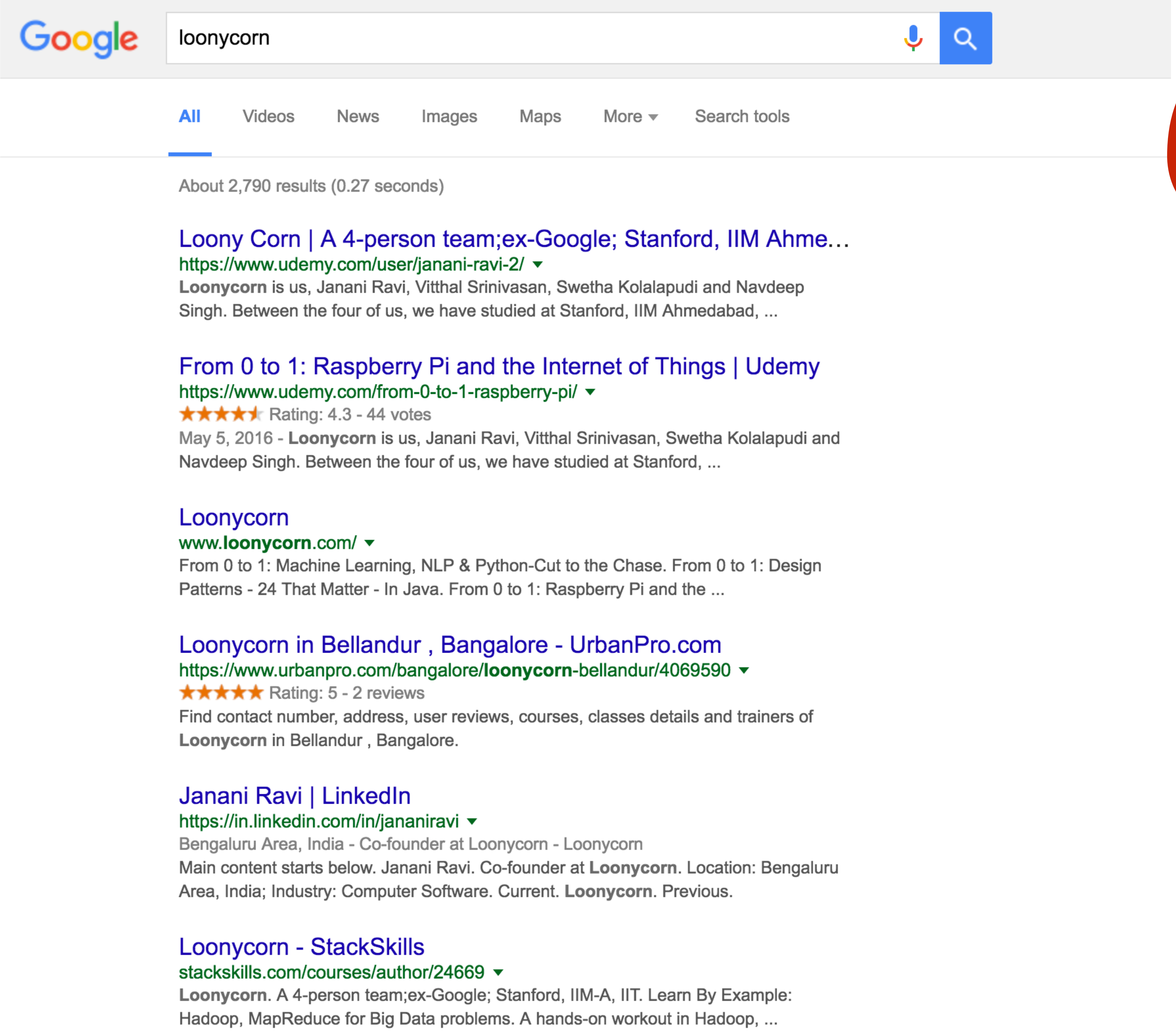
Loony Corn | A 4-person team;ex-Google; Stanford, IIM Ahme...
https://www.udemy.com/user/janani-ravi-2/ ▾
Loonycorn is us, Janani Ravi, Vitthal Srinivasan, Swetha Kolalapudi and Navdeep Singh. Between the four of us, we have studied at Stanford, IIM Ahmedabad, ...

From 0 to 1: Raspberry Pi and the Internet of Things | Udemy
https://www.udemy.com/from-0-to-1-raspberry-pi/ ▾
★★★★★ Rating: 4.3 - 44 votes
May 5, 2016 - **Loonycorn** is us, Janani Ravi, Vitthal Srinivasan, Swetha Kolalapudi and Navdeep Singh. Between the four of us, we have studied at Stanford, ...

Loonycorn
www.**loonycorn**.com/ ▾
From 0 to 1: Machine Learning, NLP & Python-Cut to the Chase. From 0 to 1: Design Patterns - 24 That Matter - In Java. From 0 to 1: Raspberry Pi and the ...

Loonycorn in Bellandur - Bangalore - UrbanPro.com

Often, there are 1000s or even millions of pages which are related to your search



Google

loonycorn

All Videos News Images Maps More Search tools

About 2,790 results (0.27 seconds)

Loony Corn | A 4-person team;ex-Google; Stanford, IIM Ahme...
https://www.udemy.com/user/janani-ravi-2/
Loonycorn is us, Janani Ravi, Vitthal Srinivasan, Swetha Kolalapudi and Navdeep Singh. Between the four of us, we have studied at Stanford, IIM Ahmedabad, ...

From 0 to 1: Raspberry Pi and the Internet of Things | Udemy
https://www.udemy.com/from-0-to-1-raspberry-pi/
★★★★★ Rating: 4.3 - 44 votes
May 5, 2016 - **Loonycorn** is us, Janani Ravi, Vitthal Srinivasan, Swetha Kolalapudi and Navdeep Singh. Between the four of us, we have studied at Stanford, ...

Loonycorn
www.**loonycorn**.com/
From 0 to 1: Machine Learning, NLP & Python-Cut to the Chase. From 0 to 1: Design Patterns - 24 That Matter - In Java. From 0 to 1: Raspberry Pi and the ...

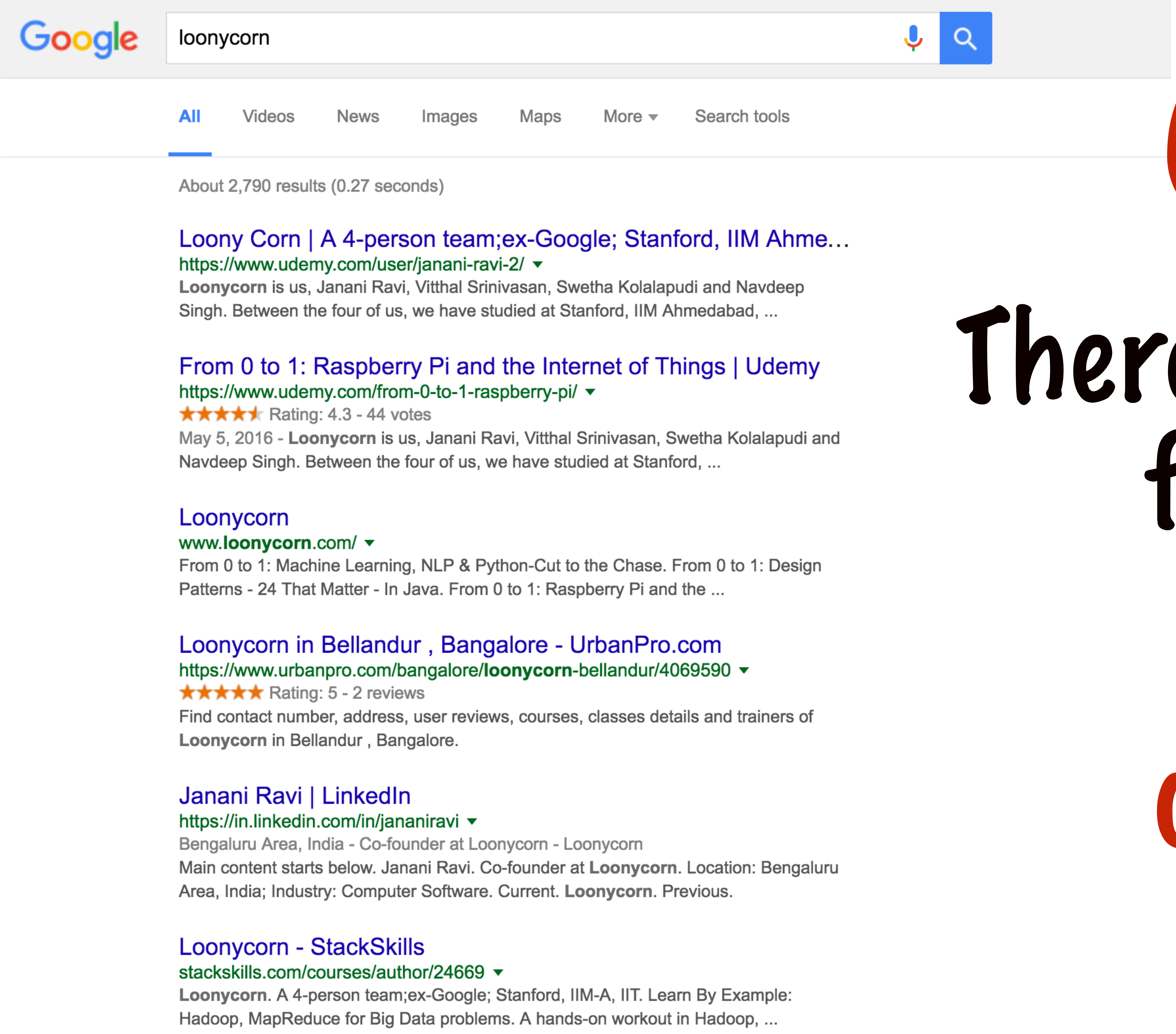
Loonycorn in Bellandur , Bangalore - UrbanPro.com
https://www.urbanpro.com/bangalore/**loonycorn**-bellandur/4069590
★★★★★ Rating: 5 - 2 reviews
Find contact number, address, user reviews, courses, classes details and trainers of **Loonycorn** in Bellandur , Bangalore.

Janani Ravi | LinkedIn
https://in.linkedin.com/in/jananiravi
Bengaluru Area, India - Co-founder at Loonycorn - Loonycorn
Main content starts below. Janani Ravi. Co-founder at **Loonycorn**. Location: Bengaluru Area, India; Industry: Computer Software. Current. **Loonycorn**. Previous.

Loonycorn - StackSkills
stackskills.com/courses/author/24669
Loonycorn. A 4-person team;ex-Google; Stanford, IIM-A, IIT. Learn By Example: Hadoop, MapReduce for Big Data problems. A hands-on workout in Hadoop, ...

Google Search

How does Google decide which results should be shown first?



Google Search

There are many factors

One of them is the overall importance of a webpage

Google Search

How do we decide what the overall importance of a webpage is?

We let other webpages tell us!

Google Search

We let other webpages tell us!

The more links there are to a page

.. the more important it is

Google Search

We let other webpages tell us!

If the pages that link to it are themselves important

The page's importance increases further

Google Search

Google has an algorithm to capture exactly this idea

PageRank

PageRank

(Named after Larry Page)

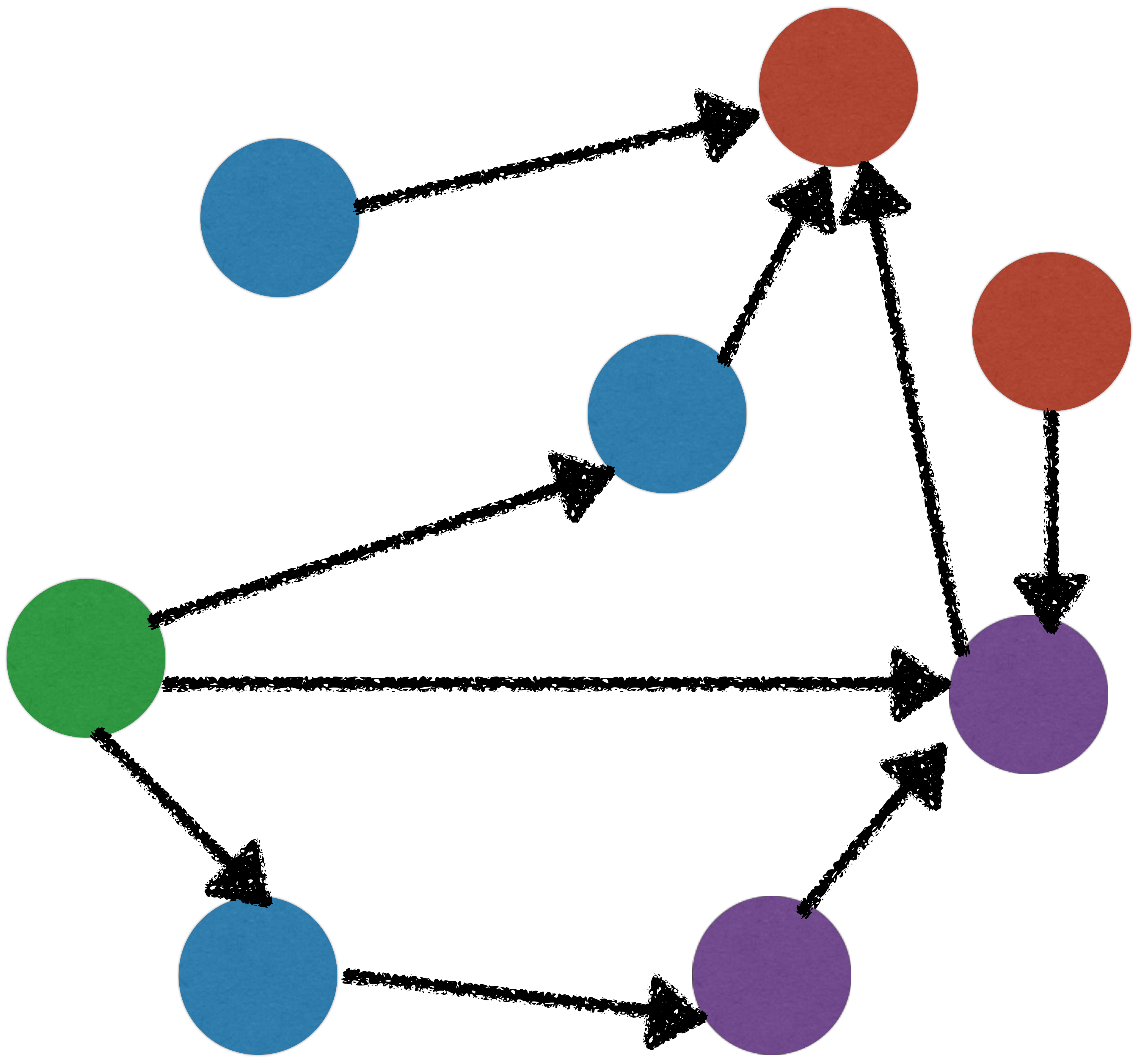
PageRank plays a significant role in deciding the ranking of search results

PageRank

PageRank is best understood using graphs

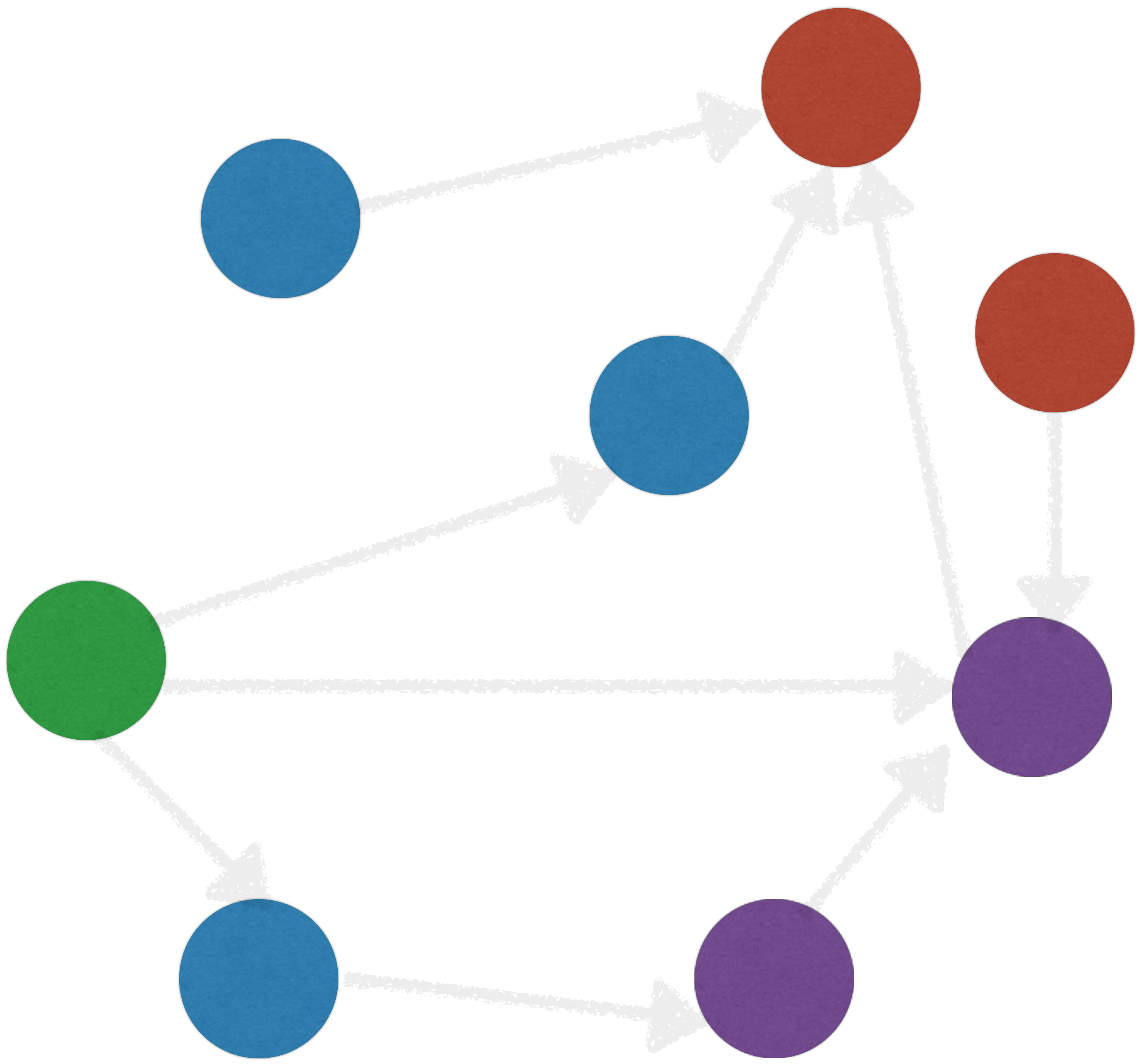
PageRank

Here is a graph representing **links between webpages**



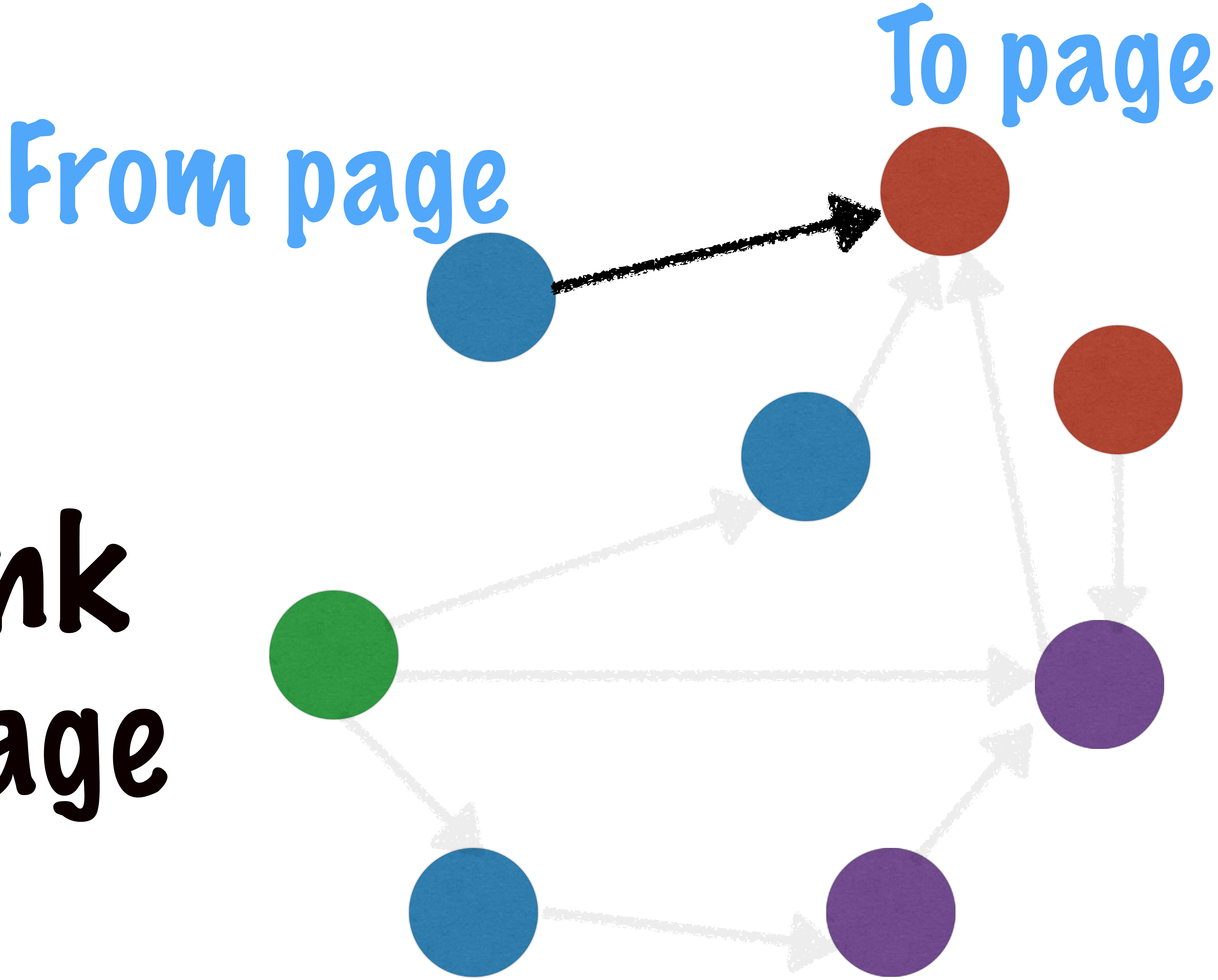
PageRank

Nodes represent **webpages**



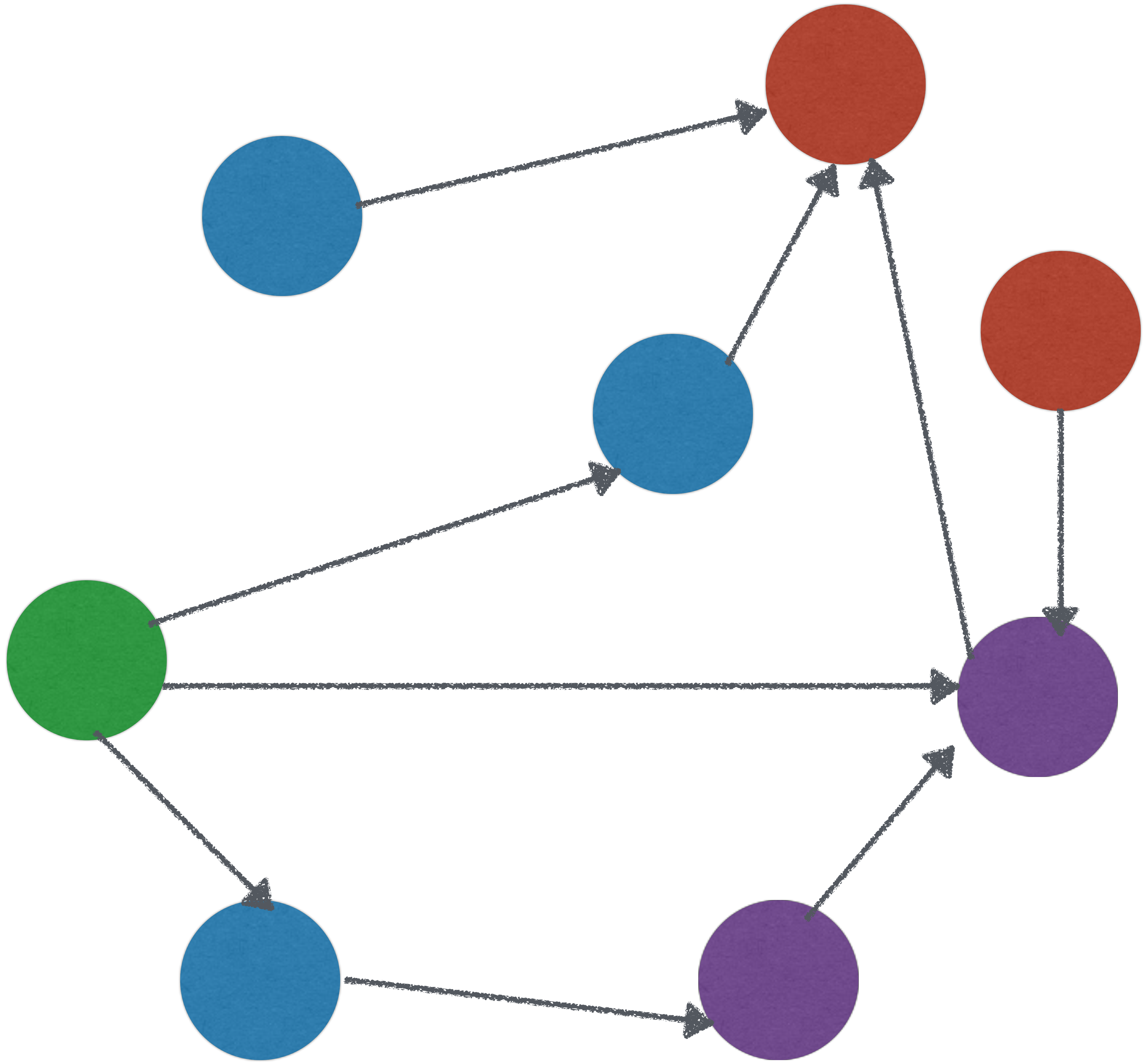
PageRank

An edge represents a link from one webpage to another



PageRank

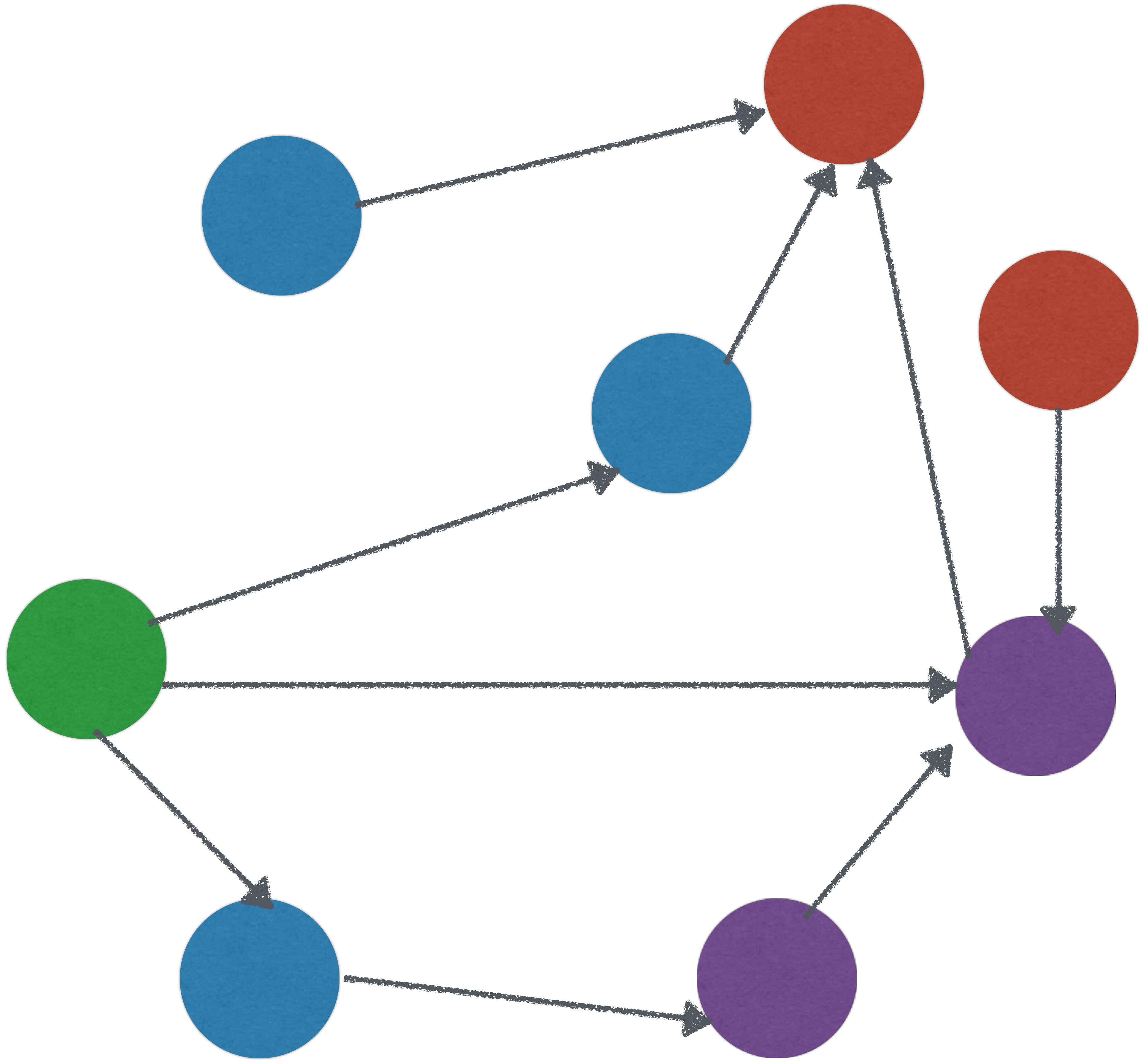
Using PageRank we can **assign a rank** to each of these webpages



PageRank

The rank increases with the number of links to the page

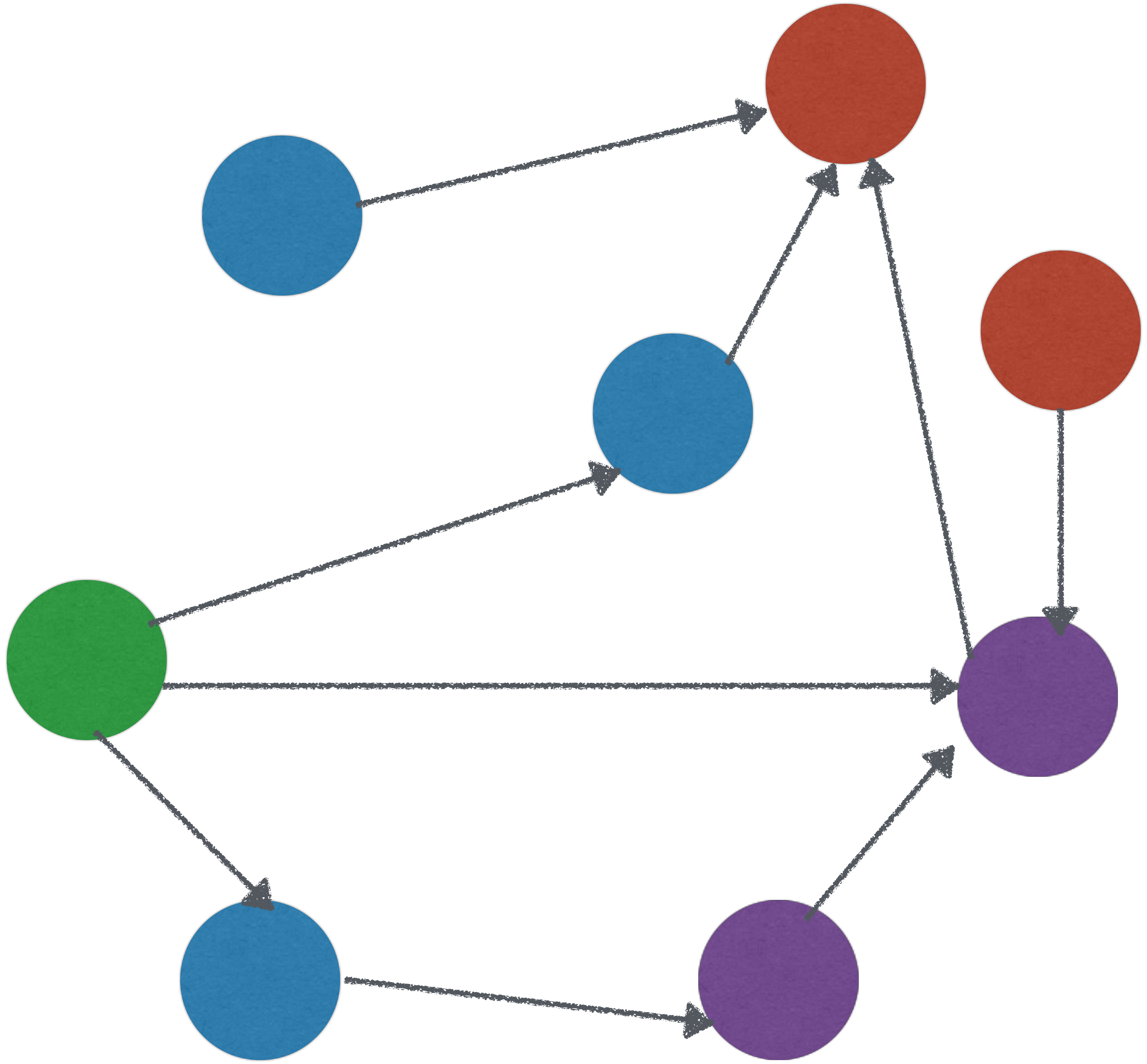
The higher the value of rank, the more important the webpage



PageRank

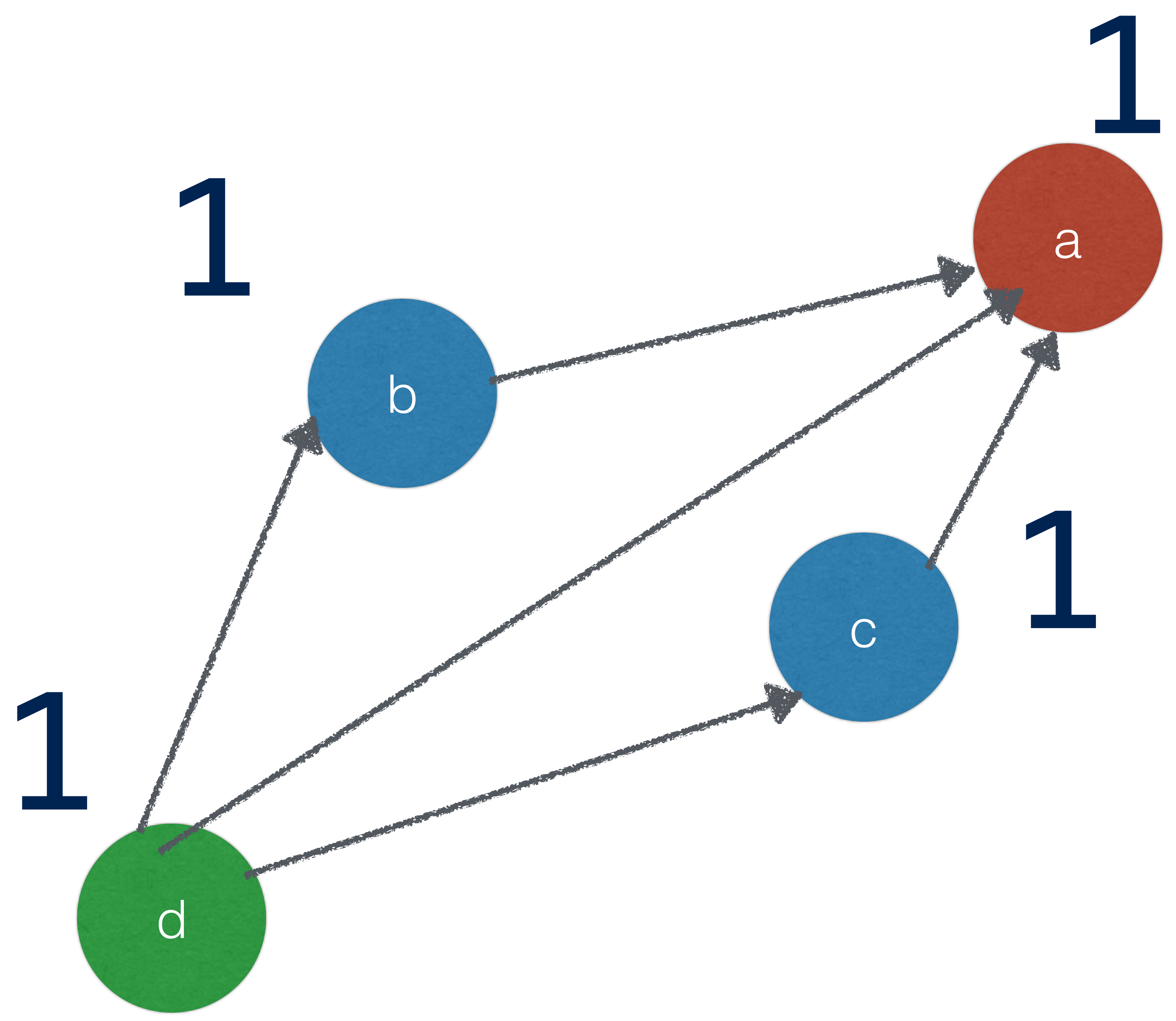
Let's take a small graph
with just 4 webpages

And go through the
PageRank algorithm



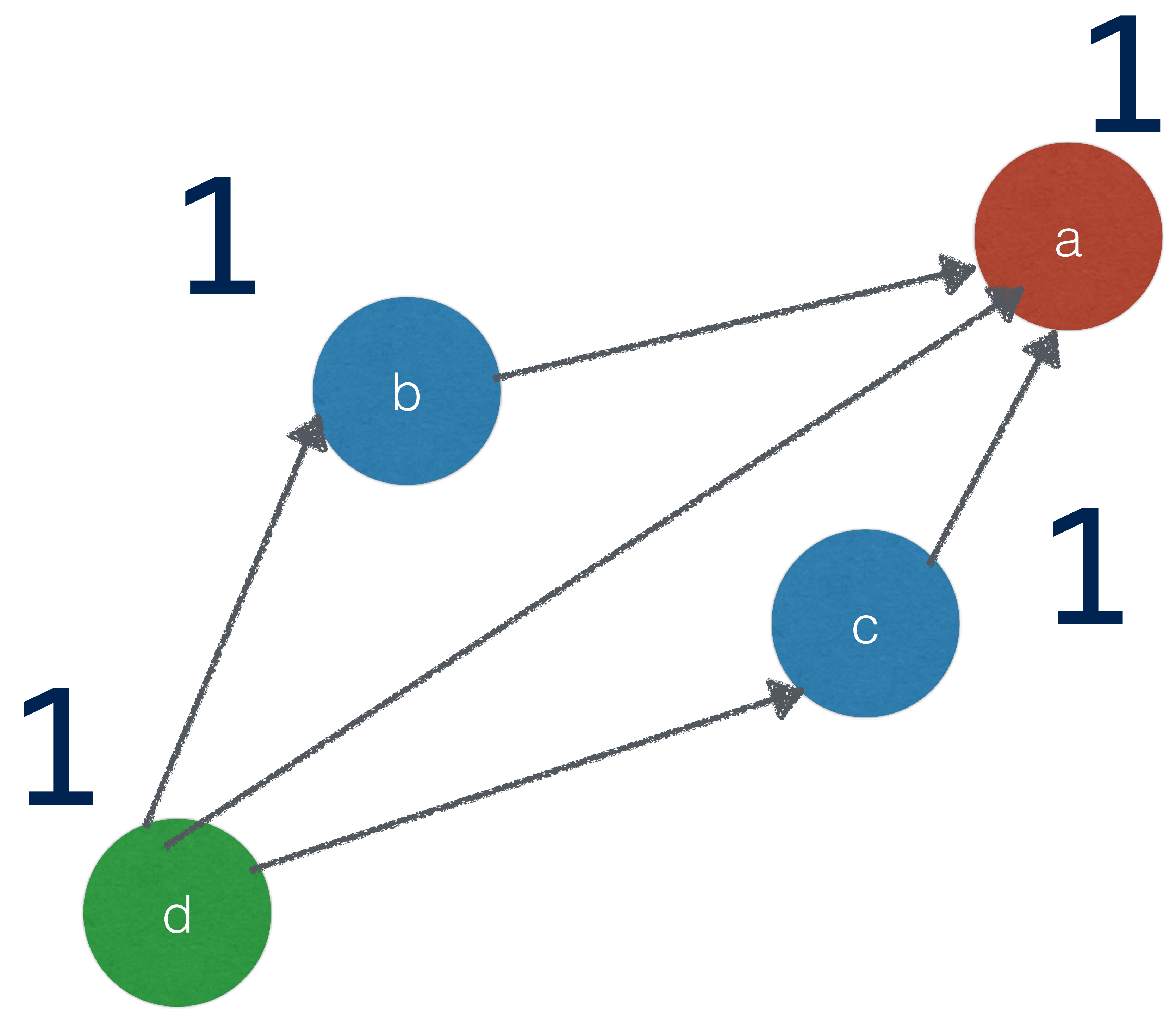
PageRank

Initially, all the webpages are given a rank of 1



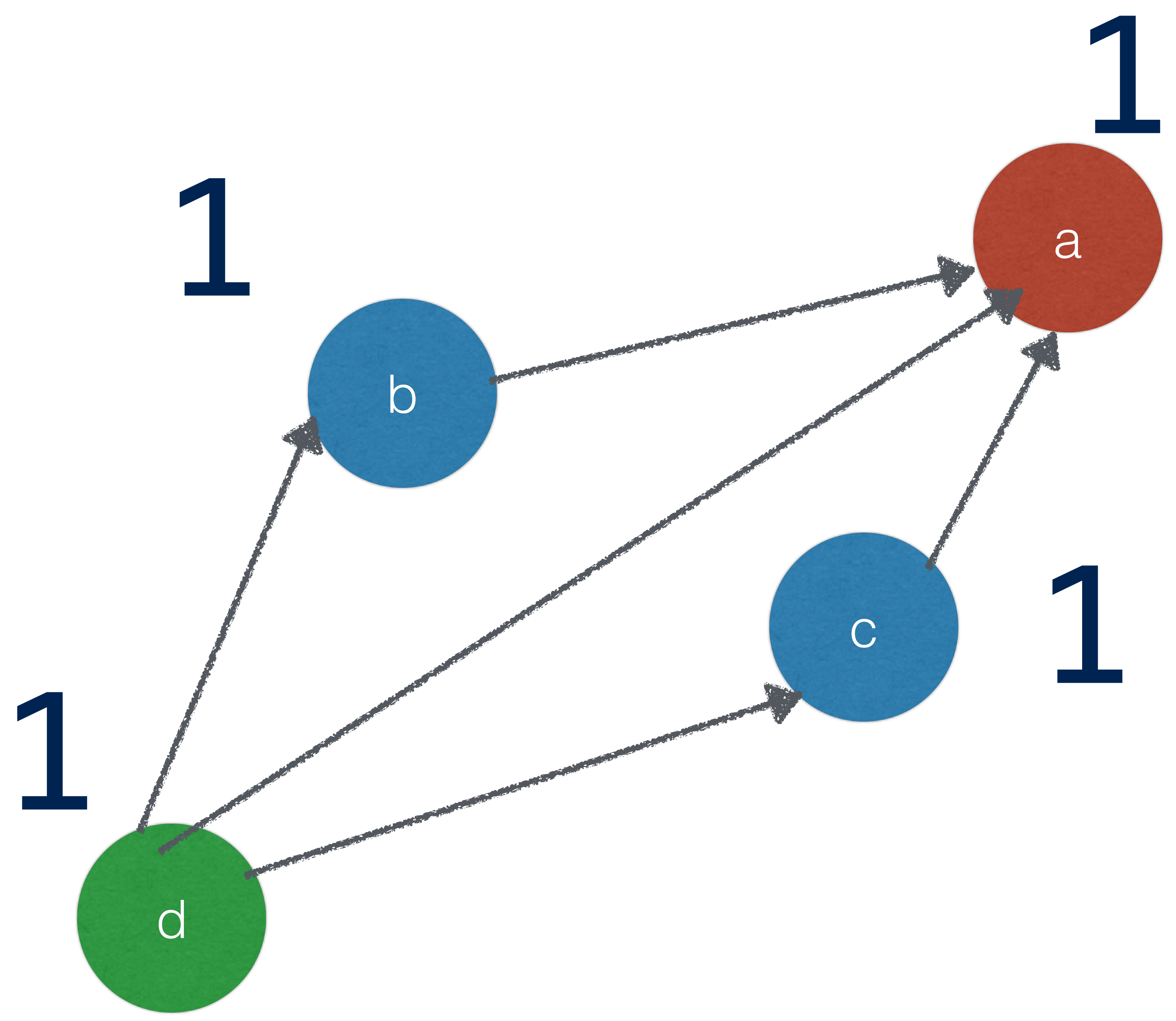
PageRank

The ranks are updated iteratively



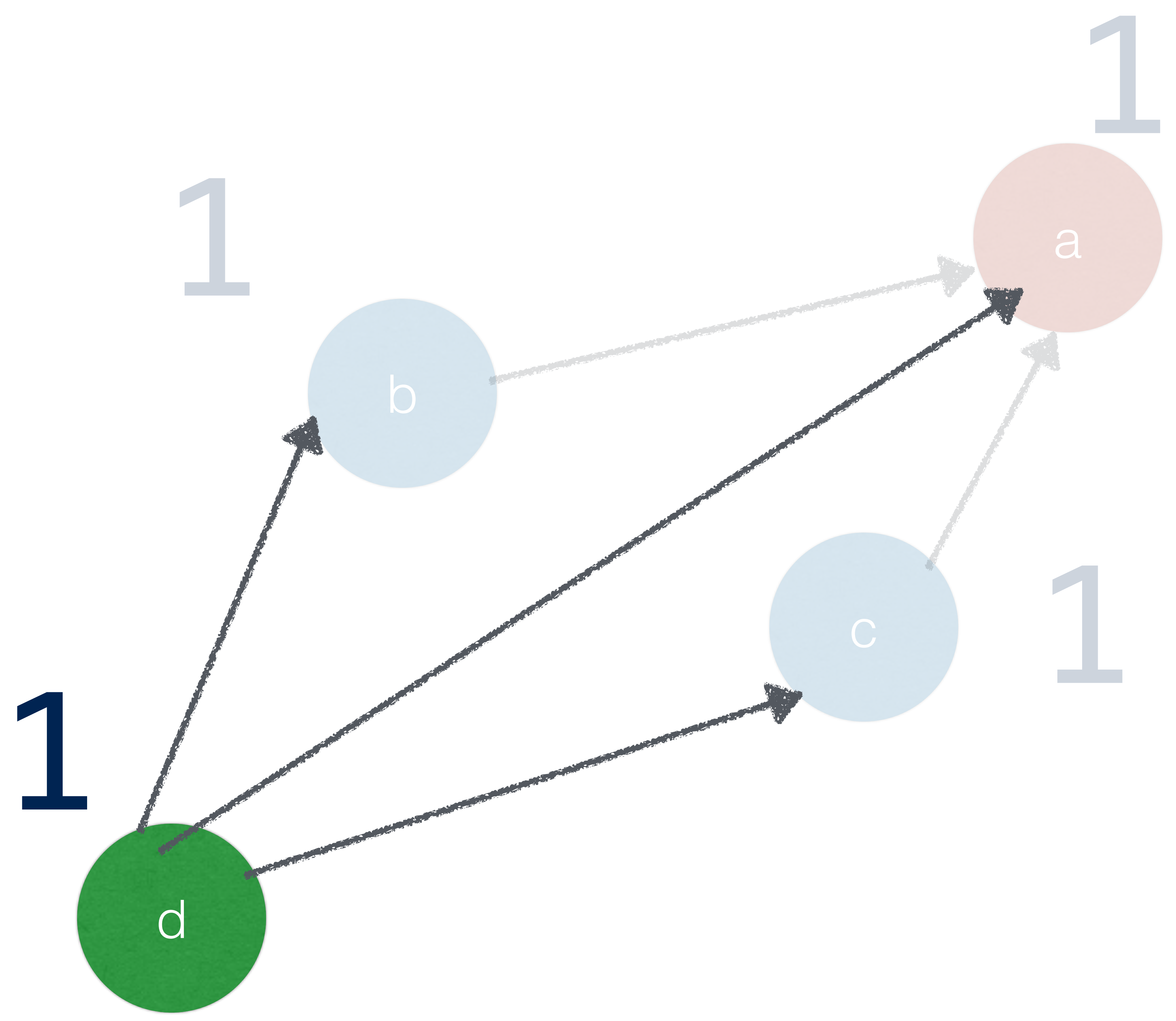
PageRank

In each iteration, pages transfer their rank equally to each of their neighbors



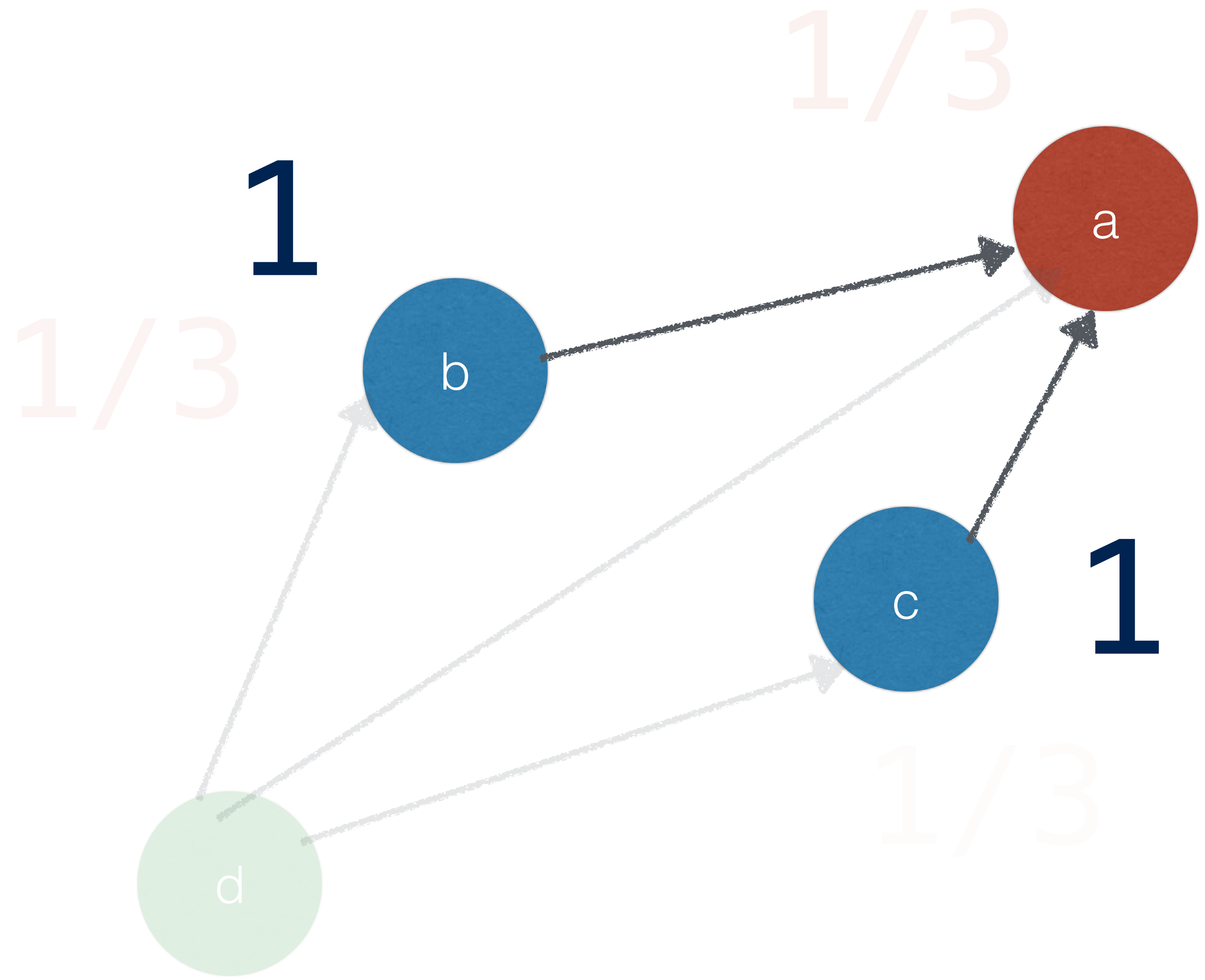
PageRank

d will transfer 1/3 to each of a,b,c



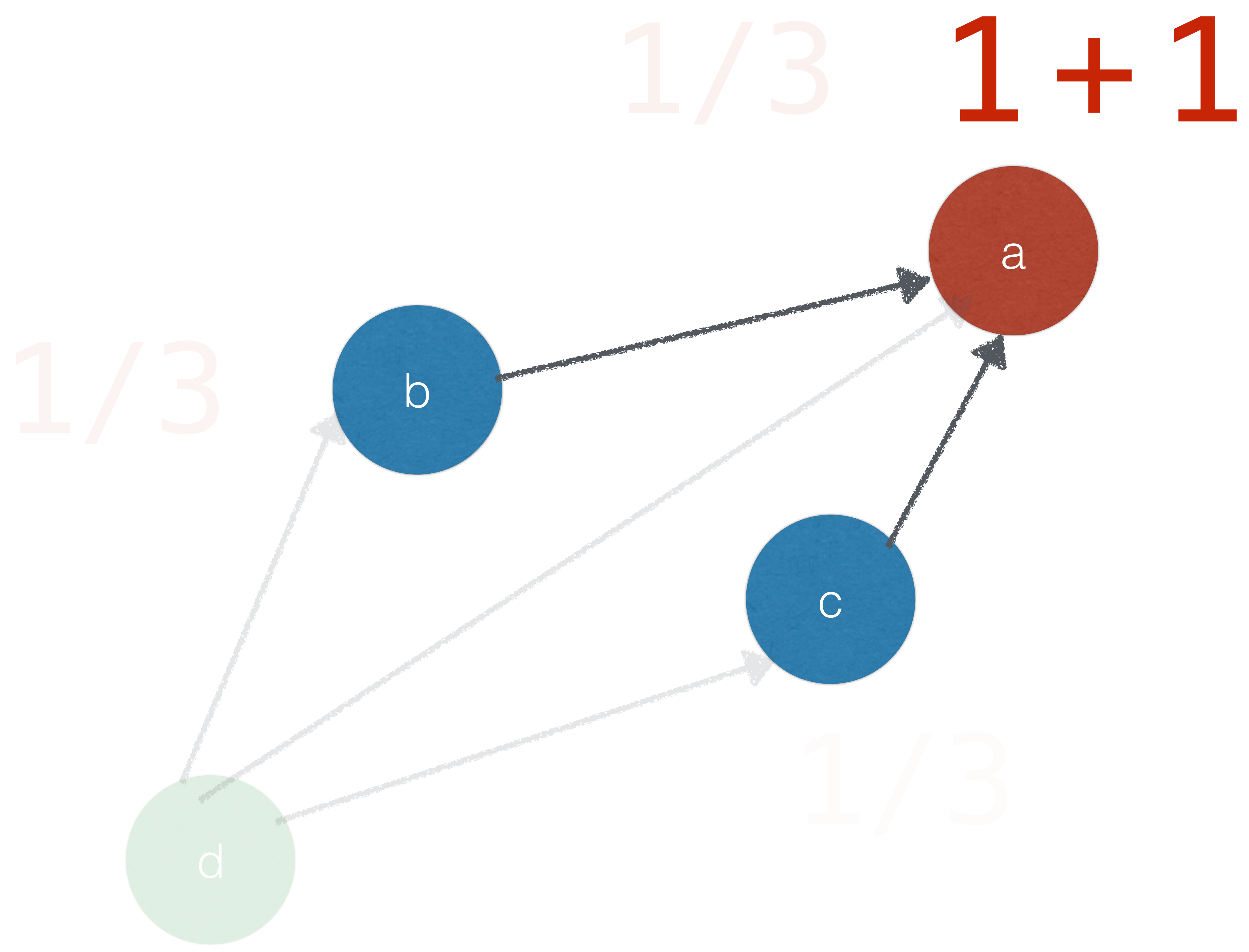
PageRank

b and c will each transfer the value 1 to a



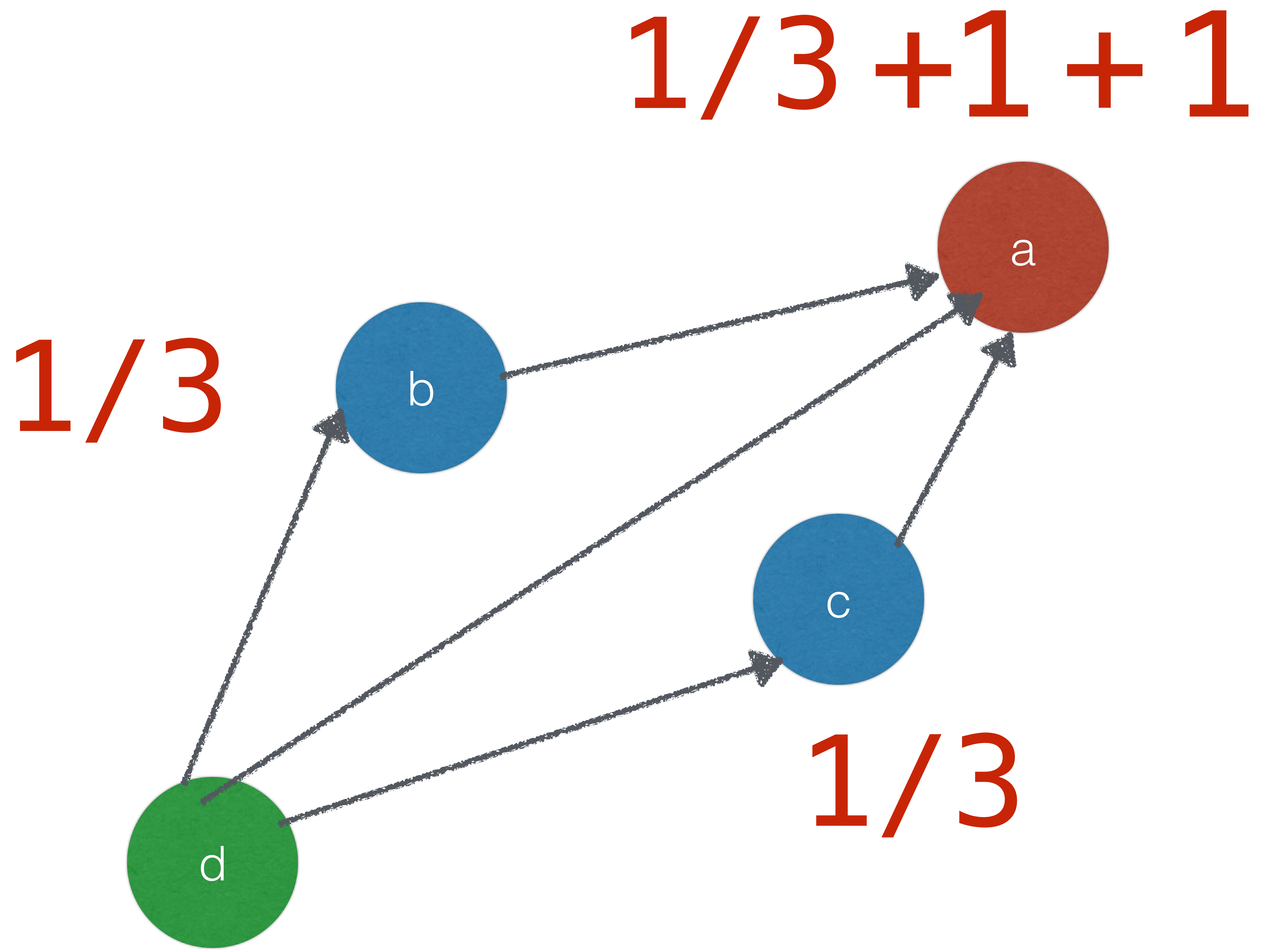
PageRank

b and c will each transfer the value 1 to a



PageRank

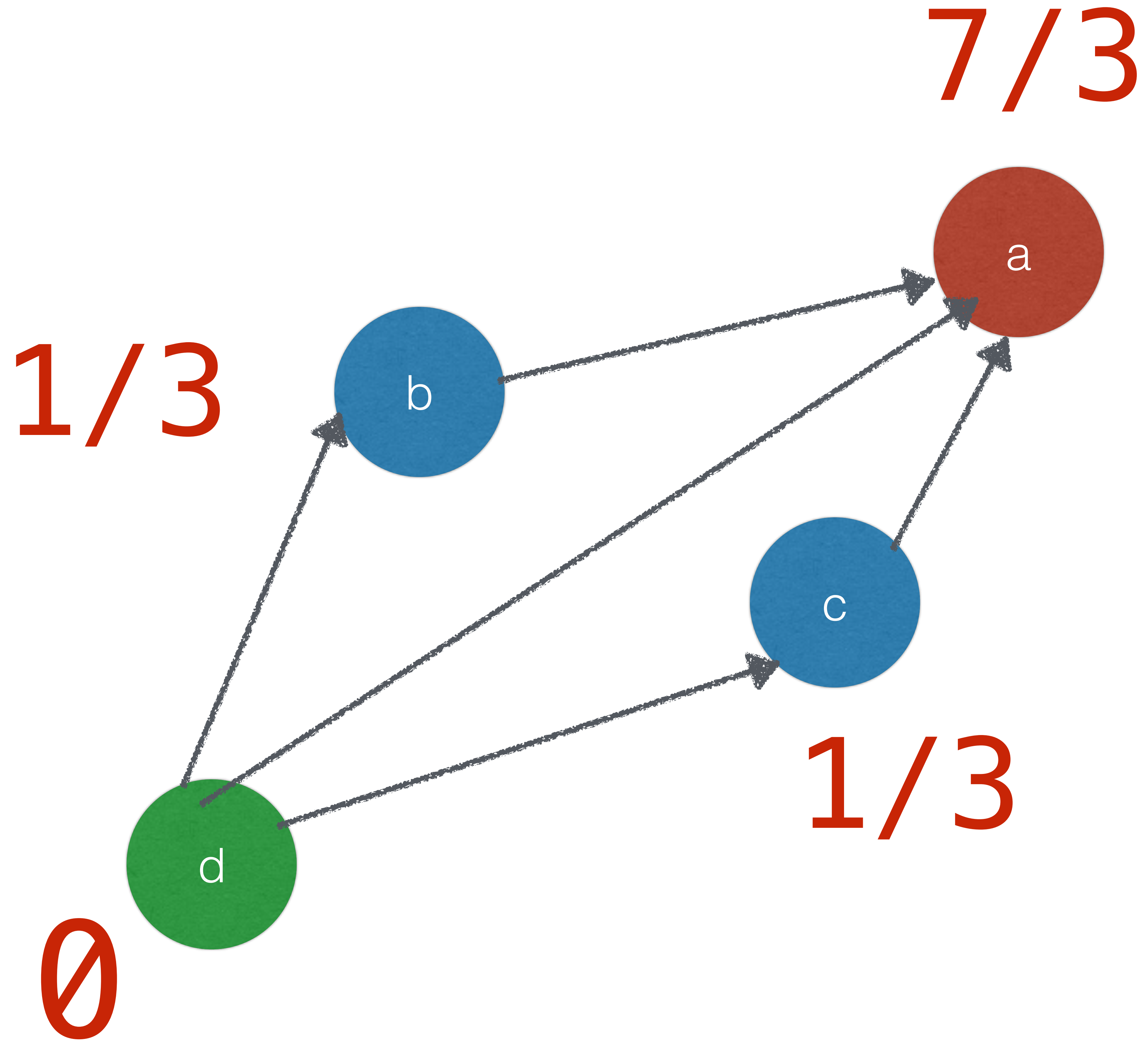
The new ranks are calculated using the sum of transferred values



PageRank

This process is repeated until the ranks converge

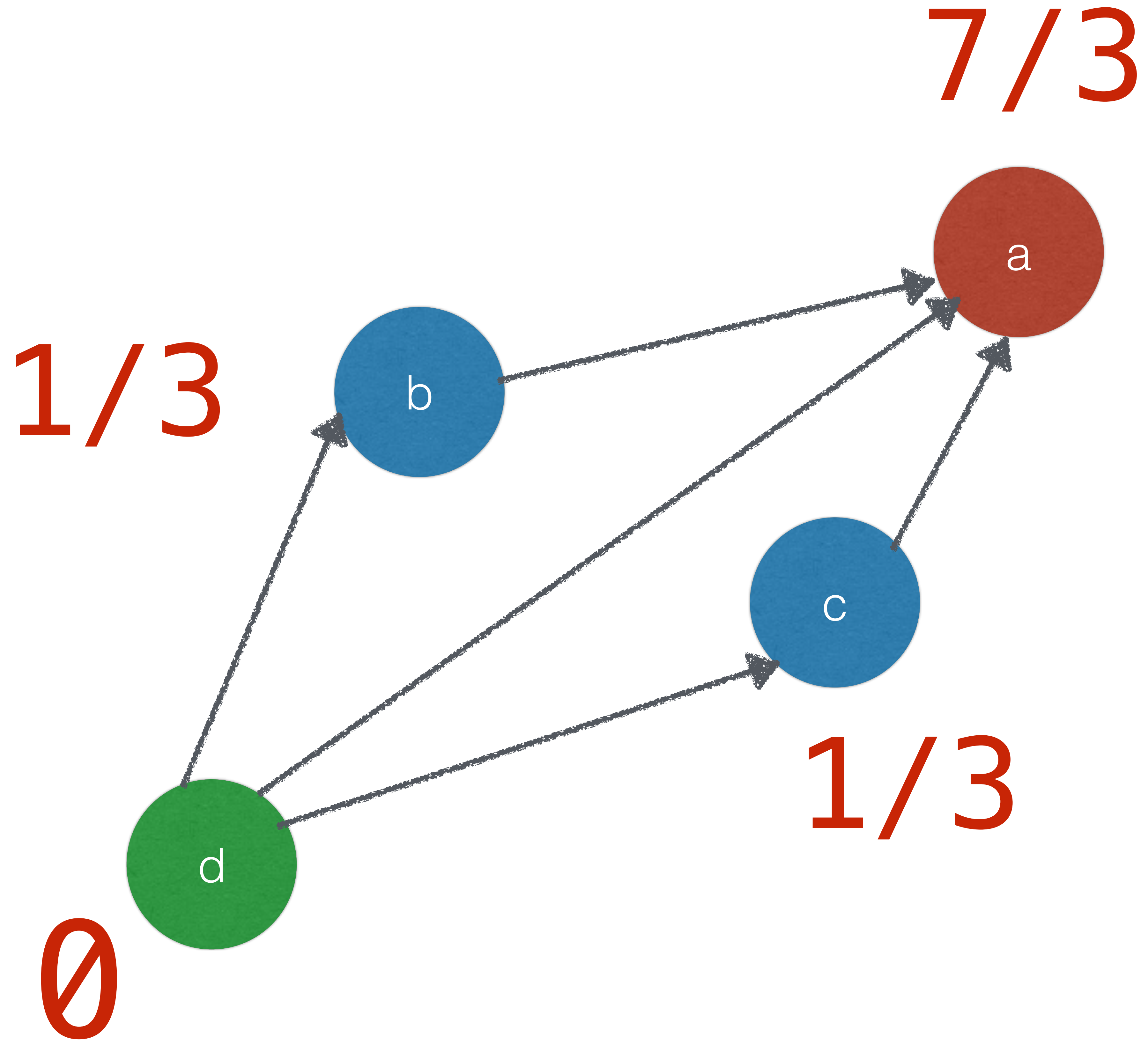
ie. until the values don't change any further



PageRank

Damping factor

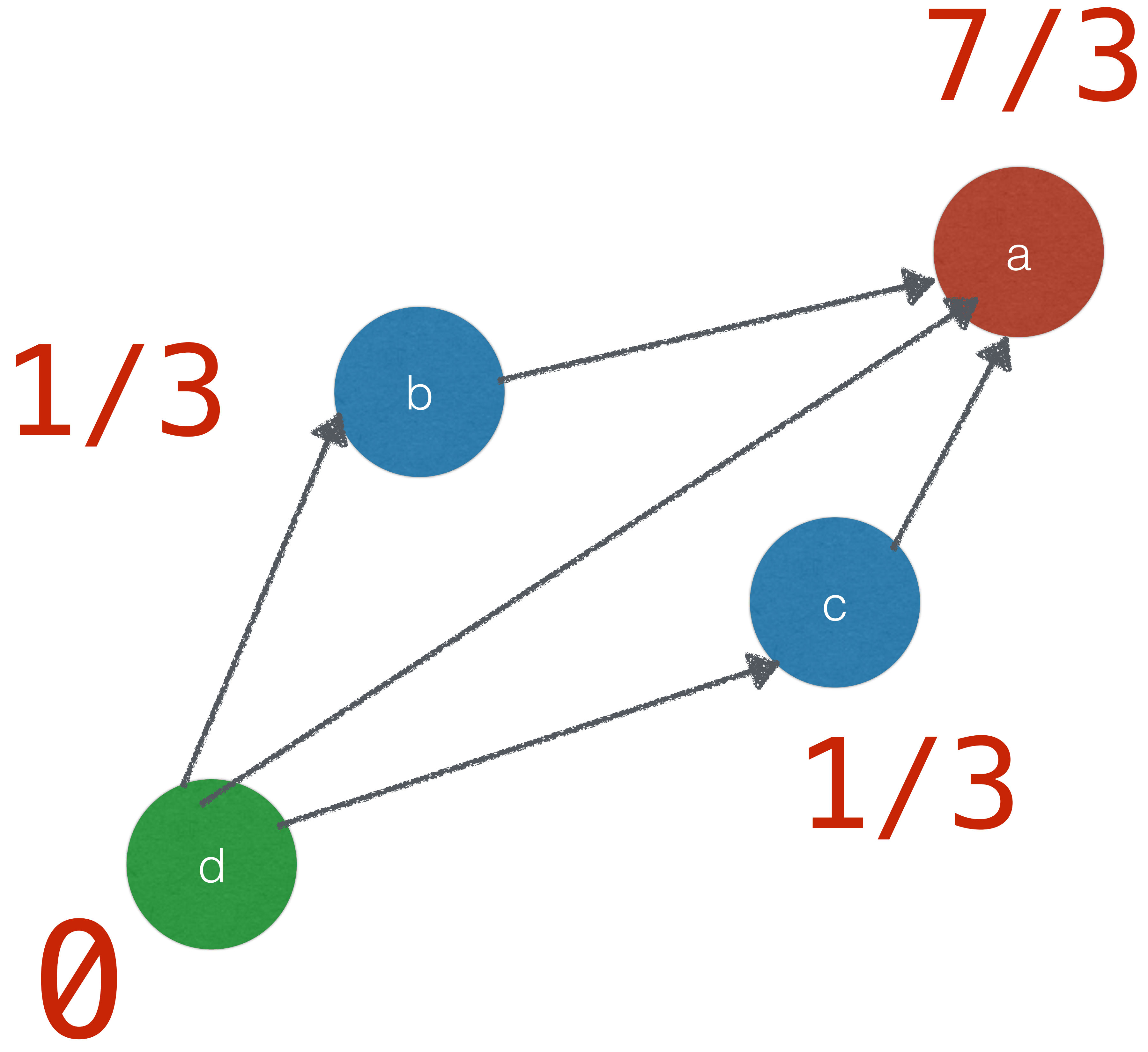
A damping factor is applied to the updated ranks after each iteration



PageRank

Damping factor

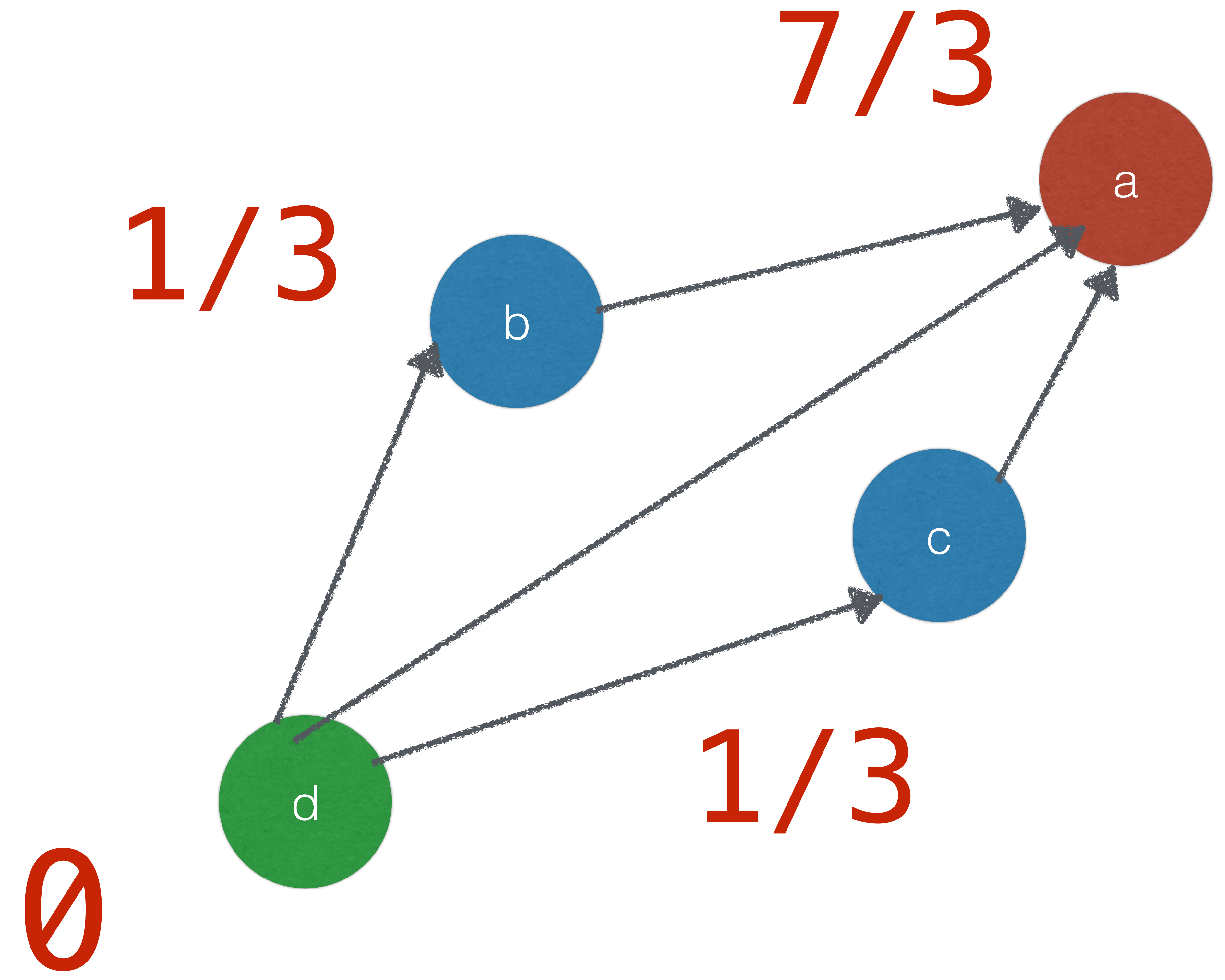
This is because in each iteration, the transferred value **has travelled farther** across the graph



PageRank

Damping factor

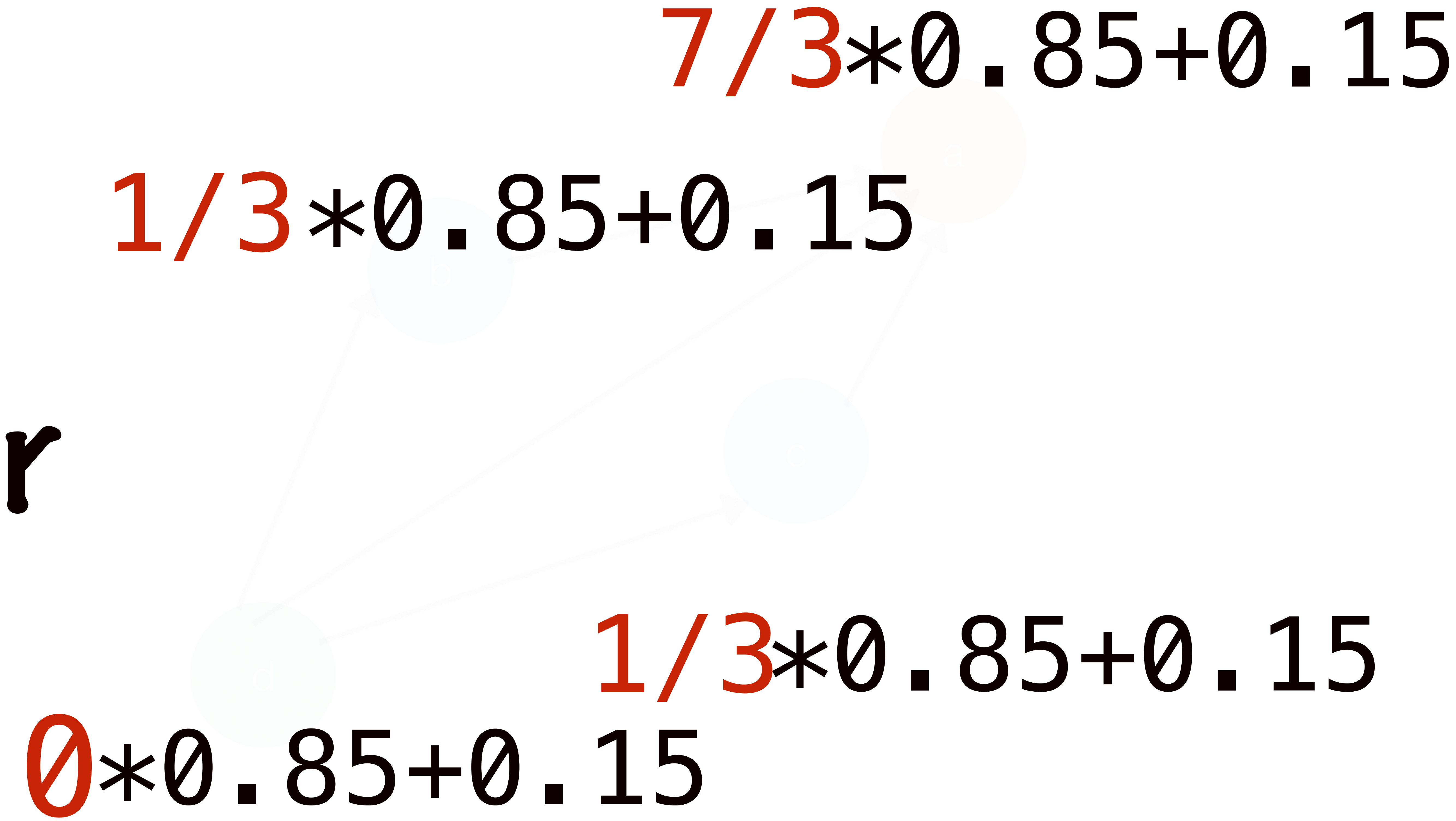
The damping factor accounts for this



PageRank

Damping factor

The damping factor accounts for this



PageRank

Let's see how to implement this in Spark

PageRank

We'll use a dataset released by Google in 2002 for a programming contest

PageRank

SNAP

Google web graph

Dataset information

Nodes represent web pages and directed edges represent hyperlinks between them. The data was released in 2002 by Google as a part of Google Programming Contest.

Dataset statistics	
Nodes	875713
Edges	5105039
Nodes in largest WCC	855802 (0.977)
Edges in largest WCC	5066842 (0.993)
Nodes in largest SCC	434818 (0.497)
Edges in largest SCC	3419124 (0.670)
Average clustering coefficient	0.5143
Number of triangles	13391903
Fraction of closed triangles	0.01911
Diameter (longest shortest path)	21
90-percentile effective diameter	8.1

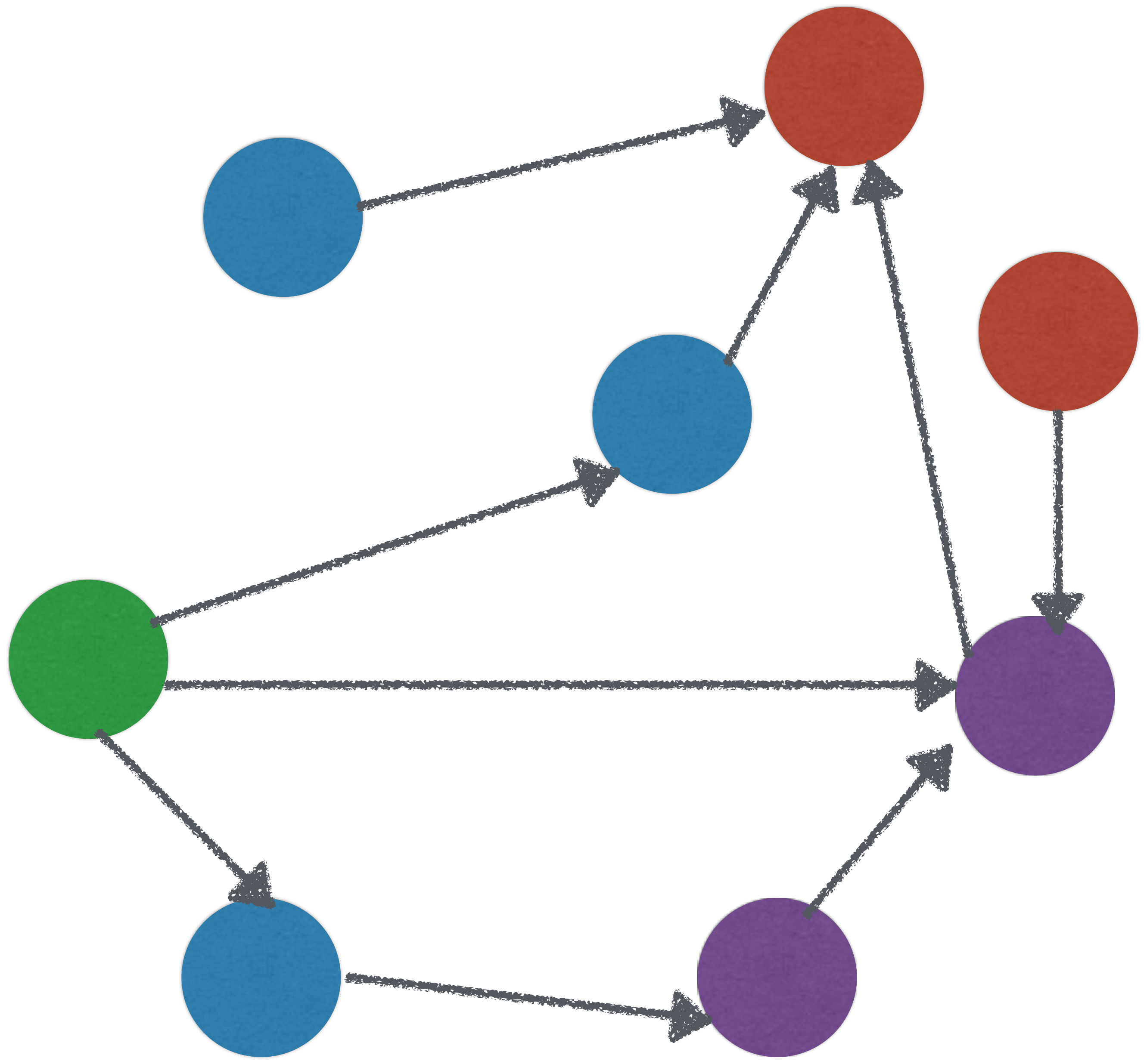
This dataset is publicly available as part of the Stanford Network Analysis Project

PageRank

FromNodeId	ToNodeId
0	11342
0	824020
0	867923
0	891835
11342	0
11342	27469
11342	38716
11342	309564
11342	322178
11342	387543
11342	427436
11342	538214
11342	638706
11342	645018
11342	835220
11342	856657
11342	867923
11342	891835

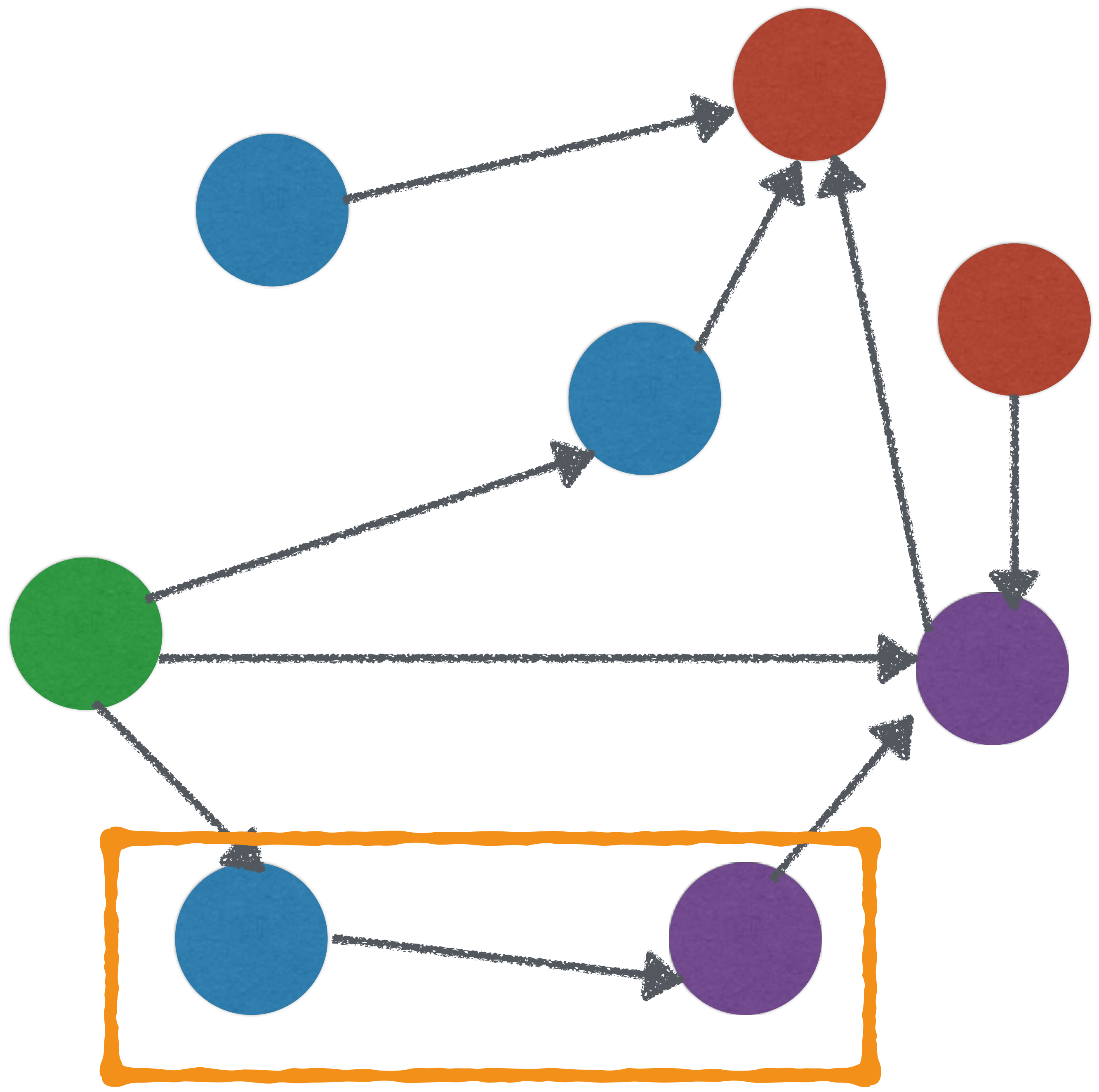
=

The dataset consists of a set of edges that form a web graph



PageRank

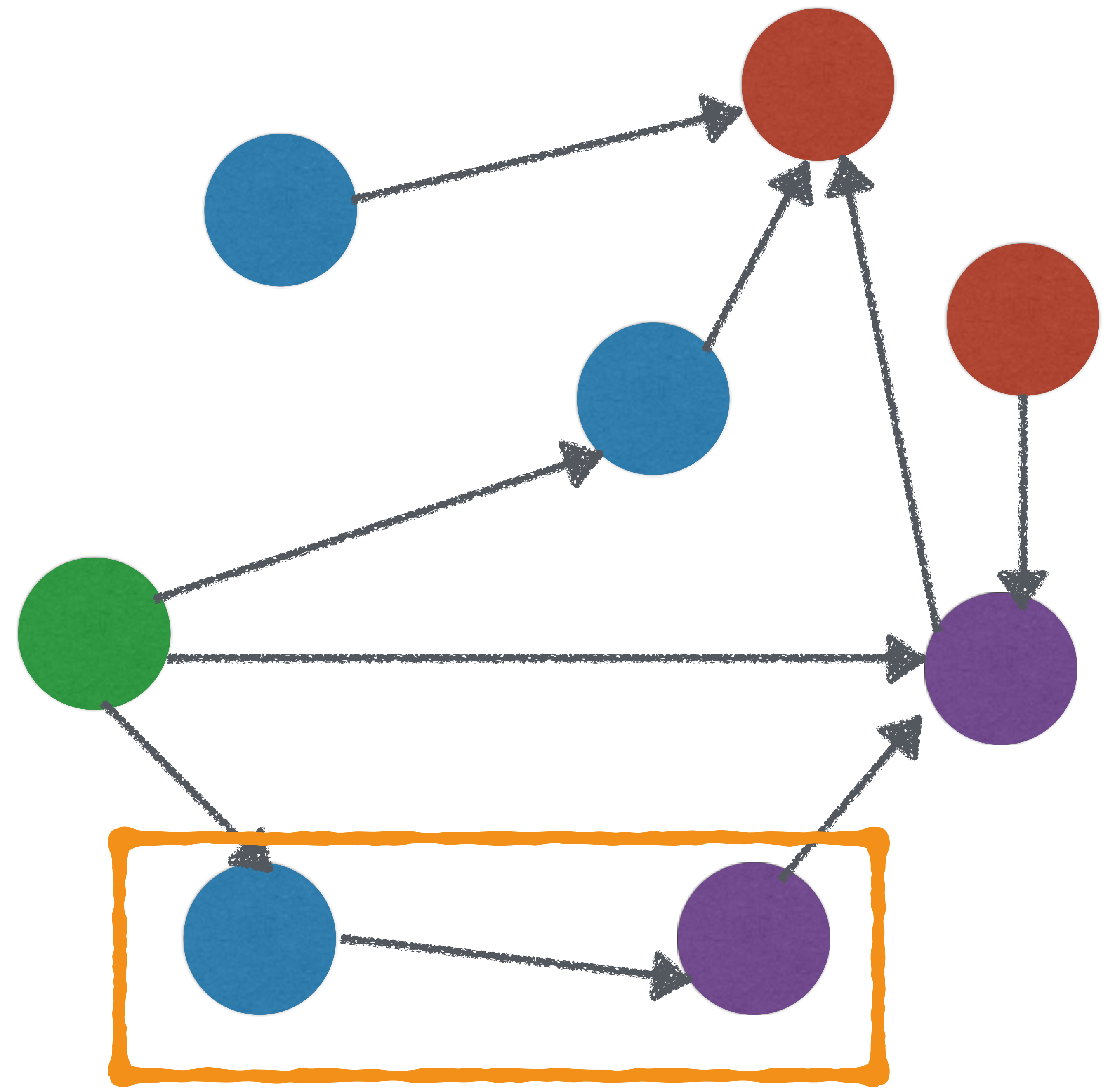
FromNodeId	ToNodeId
0	11342
0	824020
0	867923
0	891835
11342	0
11342	27469
11342	38716
11342	309564
11342	322178
11342	387543
11342	427436
11342	538214
11342	638706
11342	645018
11342	835220
11342	856657
11342	867923
11342	891835



Each row represents an edge in the graph

PageRank

FromNodeId	ToNodeId
0	11342
0	824020
0	867923
0	891835
11342	0
11342	27469
11342	38716
11342	309564
11342	322178
11342	387543
11342	427436
11342	538214
11342	638706
11342	645018
11342	835220
11342	856657
11342	867923
11342	891835



Each row represents an edge in the graph

weblinks

FromNodeId	ToNodeId
0	11342
0	824020
0	867923
0	891835
11342	0
11342	27469
11342	38716
11342	309564
11342	322178
11342	387543
11342	427436
11342	538214
11342	638706
11342	645018
11342	835220
11342	856657
11342	867923
11342	891835

Step 1: We'll load this dataset into an RDD

weblinks

FromNodeId	ToNodeId
0	11342
0	824020
0	867923
0	891835
11342	0
11342	27469
11342	38716
11342	309564
11342	322178
11342	387543
11342	427436
11342	538214
11342	638706
11342	645018
11342	835220
11342	856657
11342	867923
11342	891835

Step 1: We'll load this dataset into an RDD

Step 2: Create a links RDD with all outgoing links from a page

PageRank

Step 1: We'll load this dataset into an RDD

Links	
FromNodeId	List of ToNodeIds
0	11342, 824020,867923,891835
11342	0,27469,38716,309564,322178....
..	..

Step 2: Create a links RDD with all outgoing links from a page

PageRank

Step 1: We'll load this dataset into an RDD
Step 2: Create a links RDD with all outgoing

Links	
FromNodeId	List of ToNodeIds
0	11342, 824020,867923,891835
11342	0,27469,38716,309564,322178....
..	..

Ranks	
NodeI	Rank
0	1
11342	1
..	..

Step 3: Initialize a ranks RDD with all ranks=1

PageRank

Step 1: We'll load this dataset into an RDD

Step 2: Create a links RDD with all outgoing

Step 3: Initialize a ranks RDD with all ranks=1

Links	
FromNodeId	List of ToNodeIds
0	11342, 824020,867923,891835
11342	0,27469,38716,309564,322178….
..	..

Ranks	
NodeI	Rank
0	1
11342	1
..	..

Step 4: Join the links and ranks RDDS

PageRank

Step 1: We'll load this dataset into an RDD

eate a links RDD with all outgoing links from a pag

ep 3: Initialize a ranks RDD with all ranks=1

Step 4: Join the links and ranks RDDS

Links		Ranks
FromNodeId	List of ToNodeIds	Rank
0	11342, 824020,867923,891835	1
11342	0,27469,38716,309564,322178....	1
..	..	..

Step 5: Each node transfers its rank equally to its neighbors

NodeId	TransferredRank
11342	0.25
824020	0.25
867923	0.25
891835	0.25

NodeId	TransferredRank
11342	0.25
824020	0.25
867923	0.25
891835	0.25
11342	0.1
27469	0.2
638706	0.25
891835	0.15

PageRank

Step 1: We'll load this dataset into an RDD

Step 2: Create a links RDD with all outgoing links from a pa

Step 3: Initialize a ranks RDD with all ranks=1

Step 4: Join the links and ranks RDDS

Step 5: Each node transfers its rank equally to its neighbors

Step 6: Apply a reduce operation on this RDD, to sum up values for the same node

Links	
FromNodeId	List of ToNodeIds
0	11342, 824020,867923,891835
11342	0,27469,38716,309564,322178....
..	..

PageRank

Step 1: We'll load this dataset into an RDD

Step 2: Create a links RDD with all outgoing links from a pa

Step 3: Initialize a ranks RDD with all ranks=1

Step 4: Join the links and ranks RDDS

Step 5: Each node transfers its rank equally to its neighbors

Step 6: Apply a reduce operation on this RDD to sum up values for the same node

Ranks	
NodeId	NewRank
11342	0.44
824020	0.36
867923	0.36
891835	0.36

Step 7: Apply the damping factor and use these as the updated ranks RDD

Links	
FromNodeId	List of ToNodeIds
0	11342, 824020,867923,891835
11342	0,27469,38716,309564,322178....
..	..

PageRank

Step 1: We'll load this dataset into an RDD

Step 2: Create a links RDD with all outgoing links from a pa

Step 3: Initialize a ranks RDD with all ranks=1

Step 4: Join the links and ranks RDDS

Step 5: Each node transfers its rank equally to its neighbors

Step 6: Apply a reduce operation on this RDD to sum up values for the same node

Step 7: Apply the damping factor and use these as the updated ranks RDD

Ranks	
NodeId	NewRank
11342	0.44
824020	0.36
867923	0.36
891835	0.36

Step 8: Repeat Steps 4-7 for a number of iterations

PageRank

Step 1: We'll load this dataset into an RDD

Step 2: Create a links RDD with all outgoing links from a page

Step 3: Initialize a ranks RDD with all ranks=1

Step 4: Join the links and ranks RDDS

Step 5: Each node transfers its rank equally to its neighbors

Step 6: Apply a reduce operation on this RDD, to sum up values for the same node

Step 7: Apply the damping factor and use these as the updated ranks RDD

Step 8: Repeat Steps 4-7 for a number of iterations