Pioneer academics

Ranking

Tim Chartier

Department of Math & CS

DAVIDSON

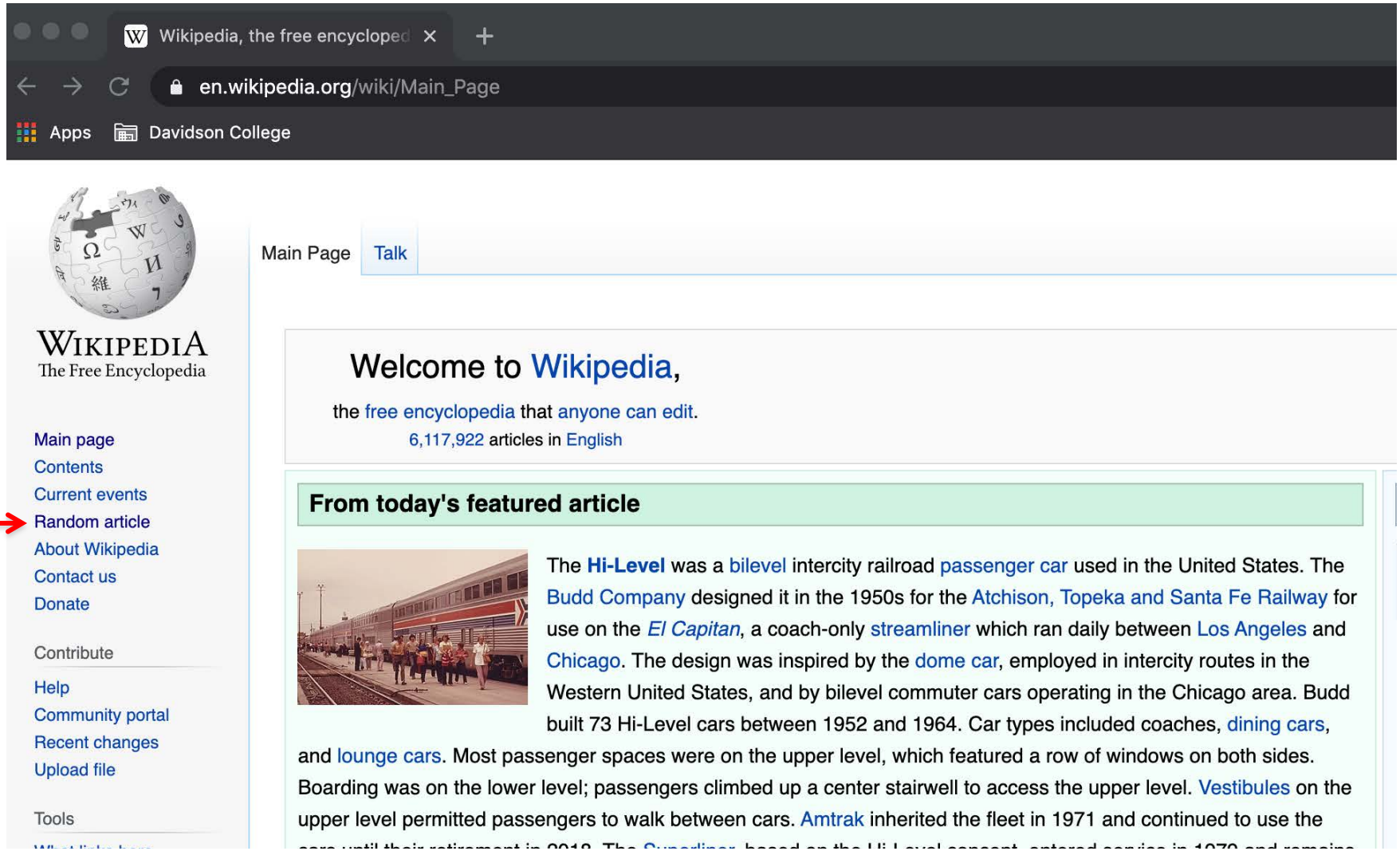
5 clicks to Google

A form of Wikiracing that mimics golf

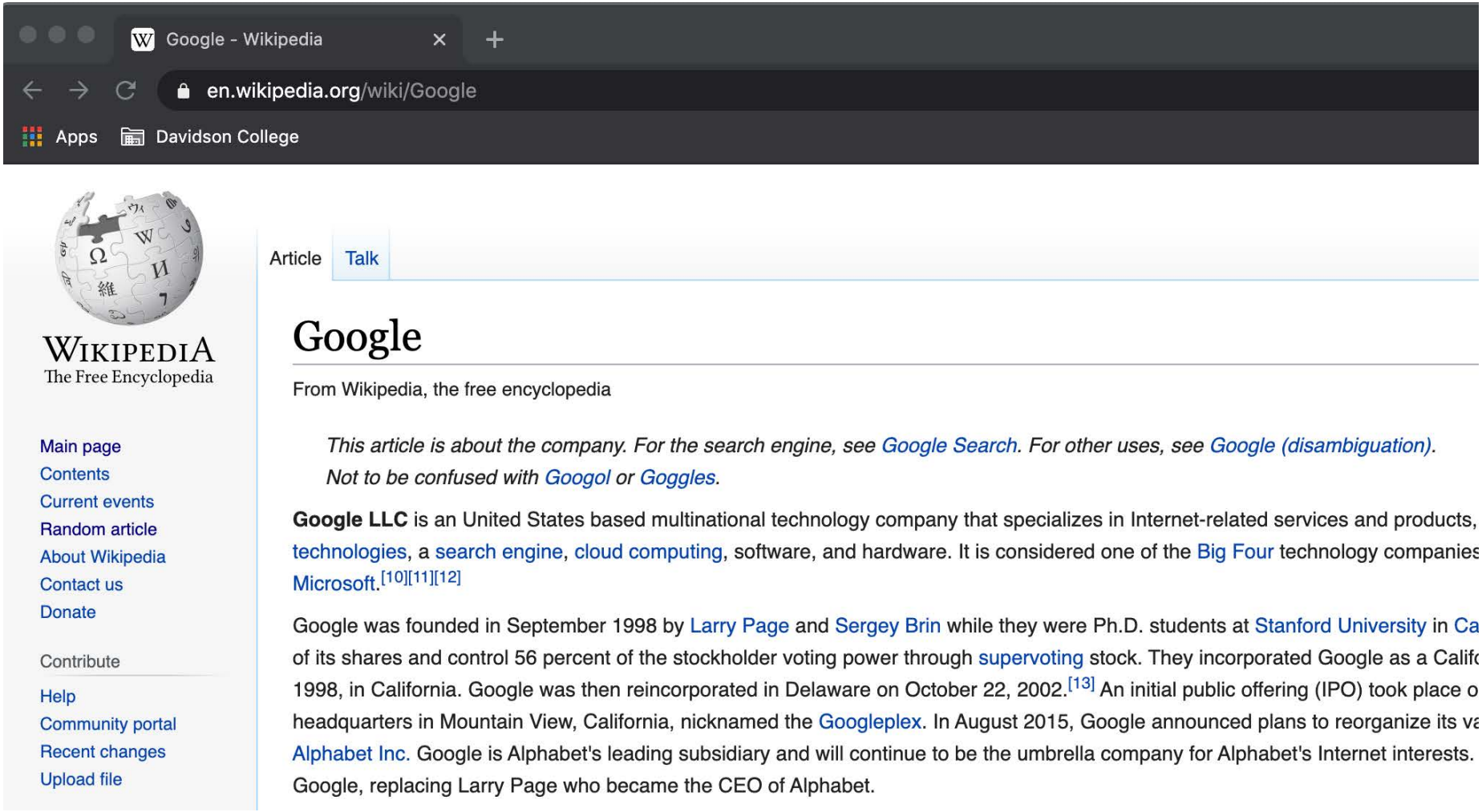
Challenge:

- Surf from a Random Article to the Google entry of Wikipedia in as few clicks as possible.
- Reaching the article in 5 clicks is considered 'par', with clicks over or under five being referred to as 'bogeys' and 'birdies' respectively.
- Lowest score wins!

Random start



Defined end


Google - Wikipedia
en.wikipedia.org/wiki/Google
Apps
Davidson College
WIKIPEDIA
The Free Encyclopedia
Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate
Contribute
Help
Community portal
Recent changes
Upload file
Article
Talk
Google
From Wikipedia, the free encyclopedia
This article is about the company. For the search engine, see Google Search. For other uses, see Google (disambiguation).
Not to be confused with Googol or Goggles.
Google LLC is an United States based multinational technology company that specializes in Internet-related services and products,
technologies, a search engine, cloud computing, software, and hardware. It is considered one of the Big Four technology companies
Microsoft.[10][11][12]
Google was founded in September 1998 by Larry Page and Sergey Brin while they were Ph.D. students at Stanford University in Ca
of its shares and control 56 percent of the stockholder voting power through supervoting stock. They incorporated Google as a Califo
1998, in California. Google was then reincorporated in Delaware on October 22, 2002.[13] An initial public offering (IPO) took place o
headquarters in Mountain View, California, nicknamed the Googleplex. In August 2015, Google announced plans to reorganize its va
Alphabet Inc. Google is Alphabet's leading subsidiary and will continue to be the umbrella company for Alphabet's Internet interests.
Google, replacing Larry Page who became the CEO of Alphabet.

Terminology

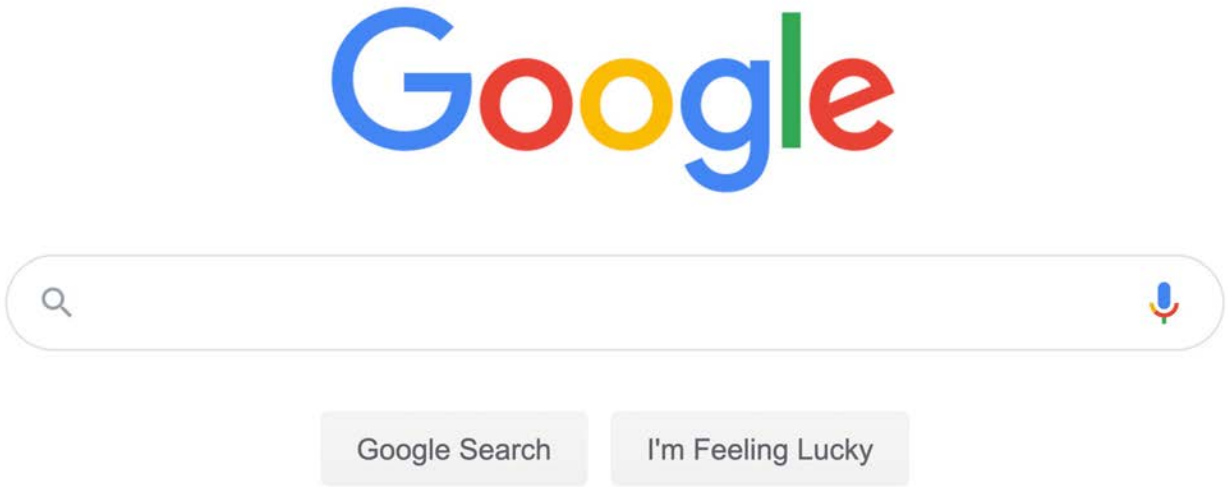
As you surfed through Wikipedia, you:

- clicked a link (*outlink* or *hyperlink*) on a web page to go to another page.
- used the hyperlink structure of Wikipedia to surf. That is, you got from one place to another only by clicking links.

A web address is also called a URL.

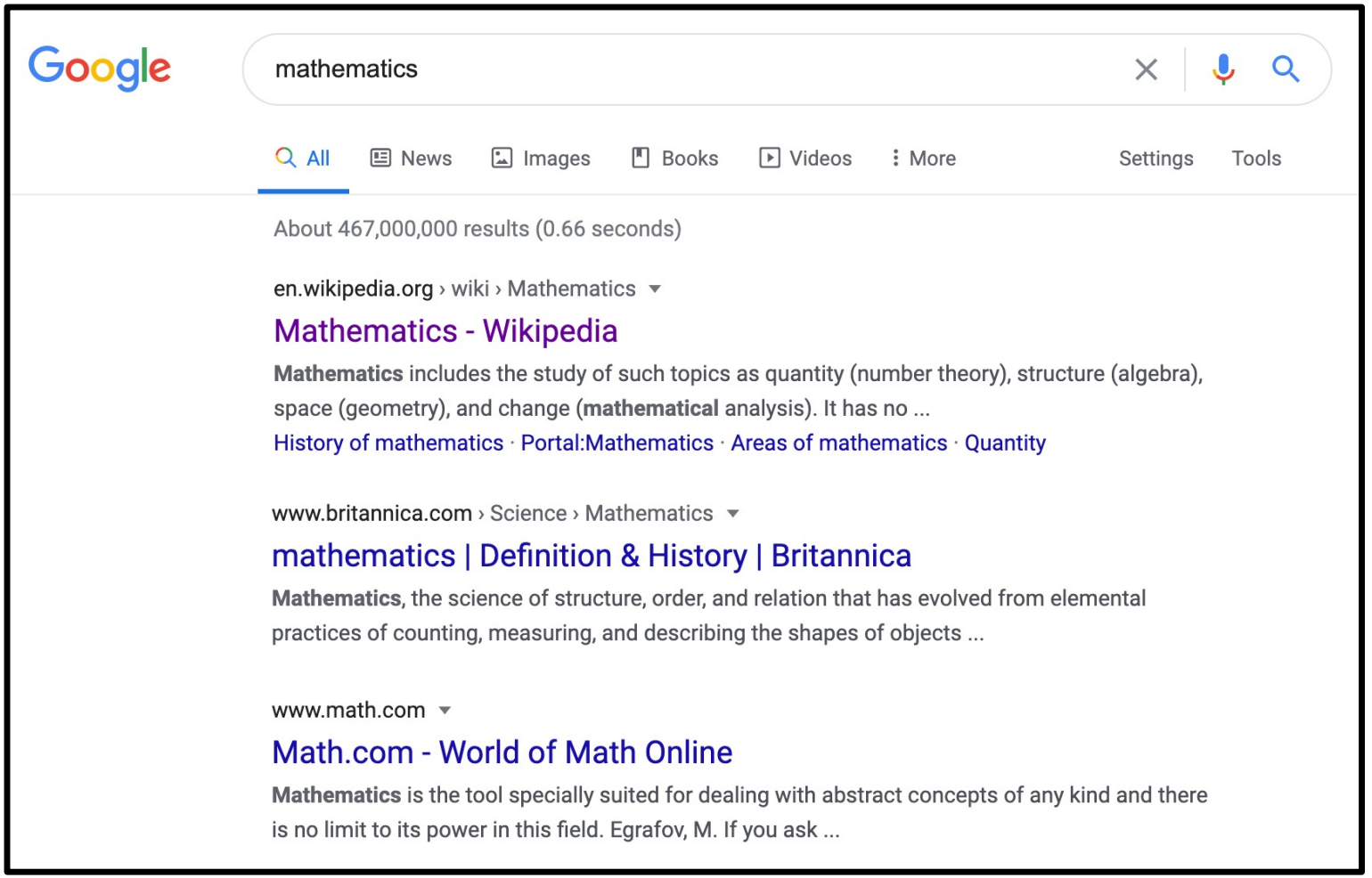
Query

- Your Wikipedia surfing will help us understand the linear algebra used by Google.
- Suppose you submit the word “mathematics” to Google.



Ranked results

A ranked list of web pages is returned.

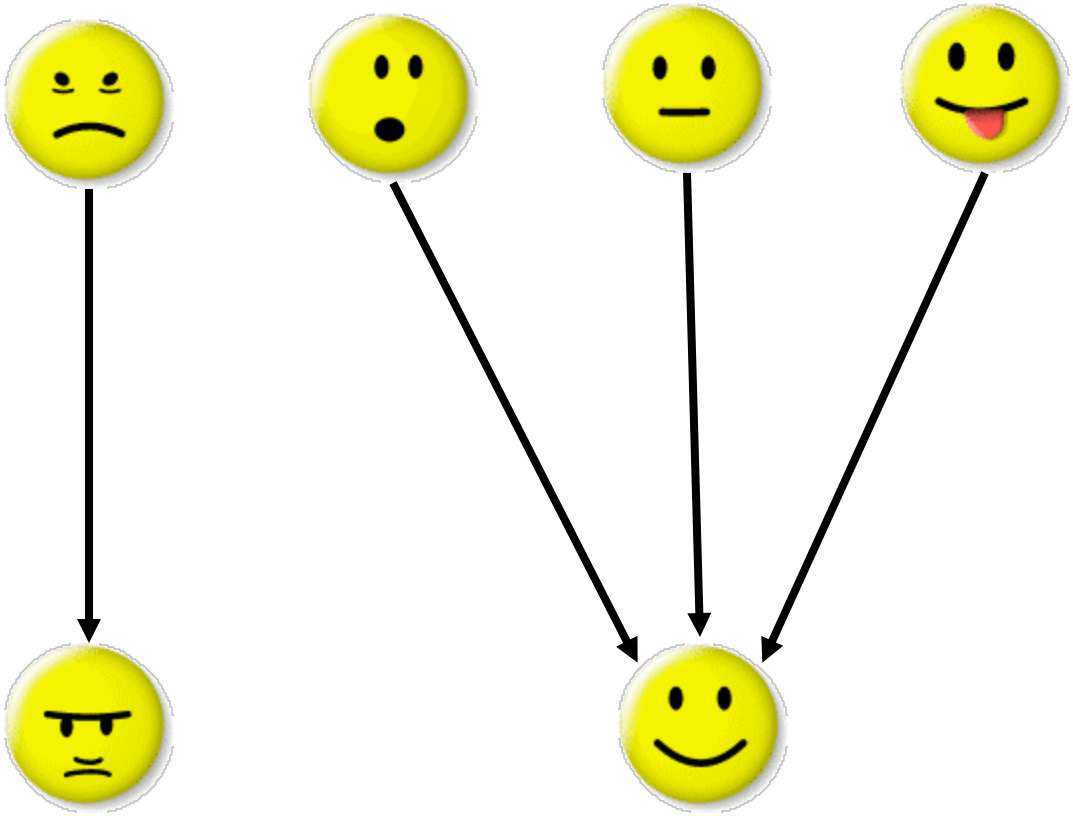


PageRank

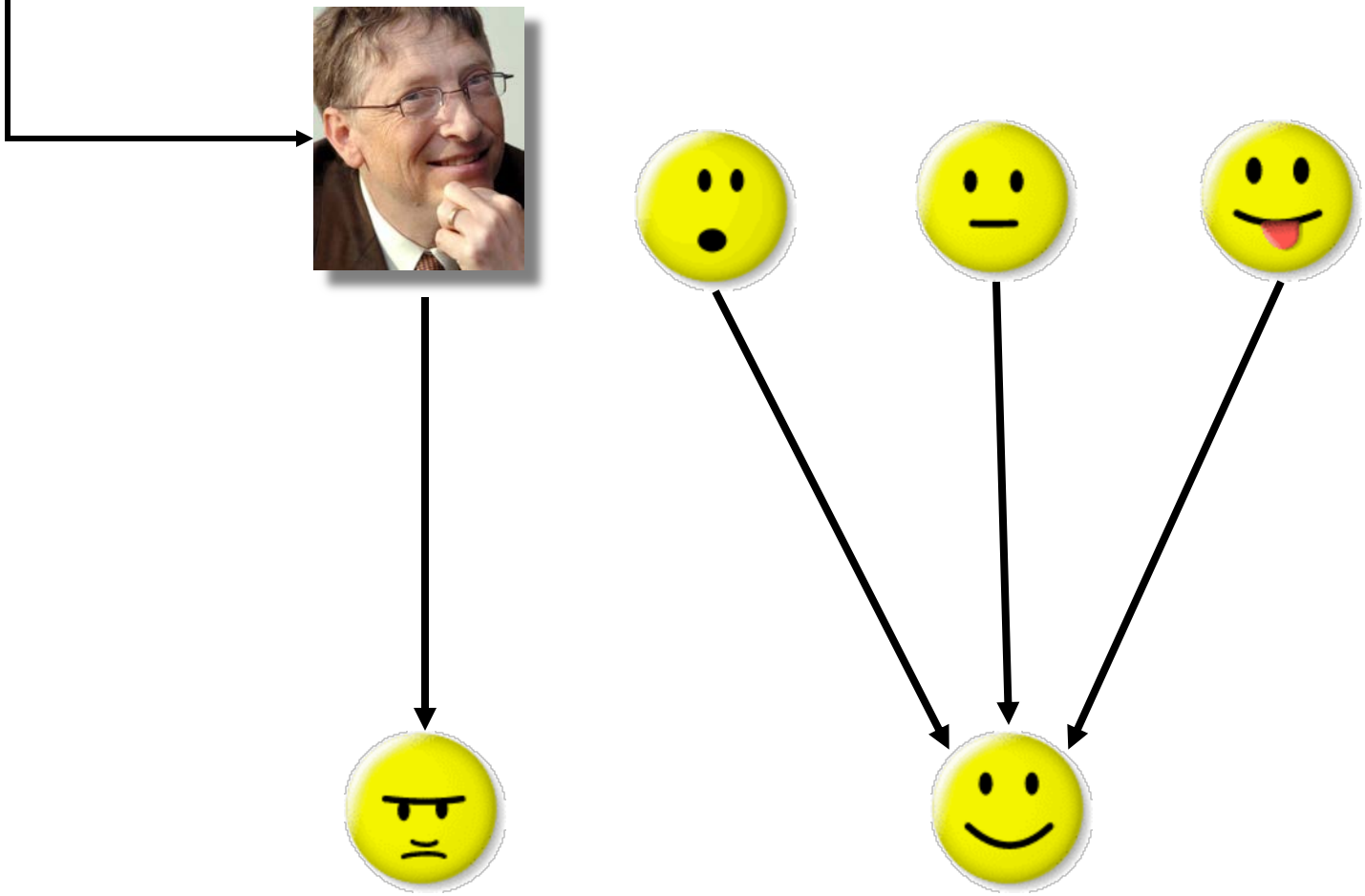
- Assuming 2 web pages are deemed equally relevant to a query, why is one page ranked over the other?
- Google measures the quality of pages.
- Quality pages are linked by quality pages!

Google

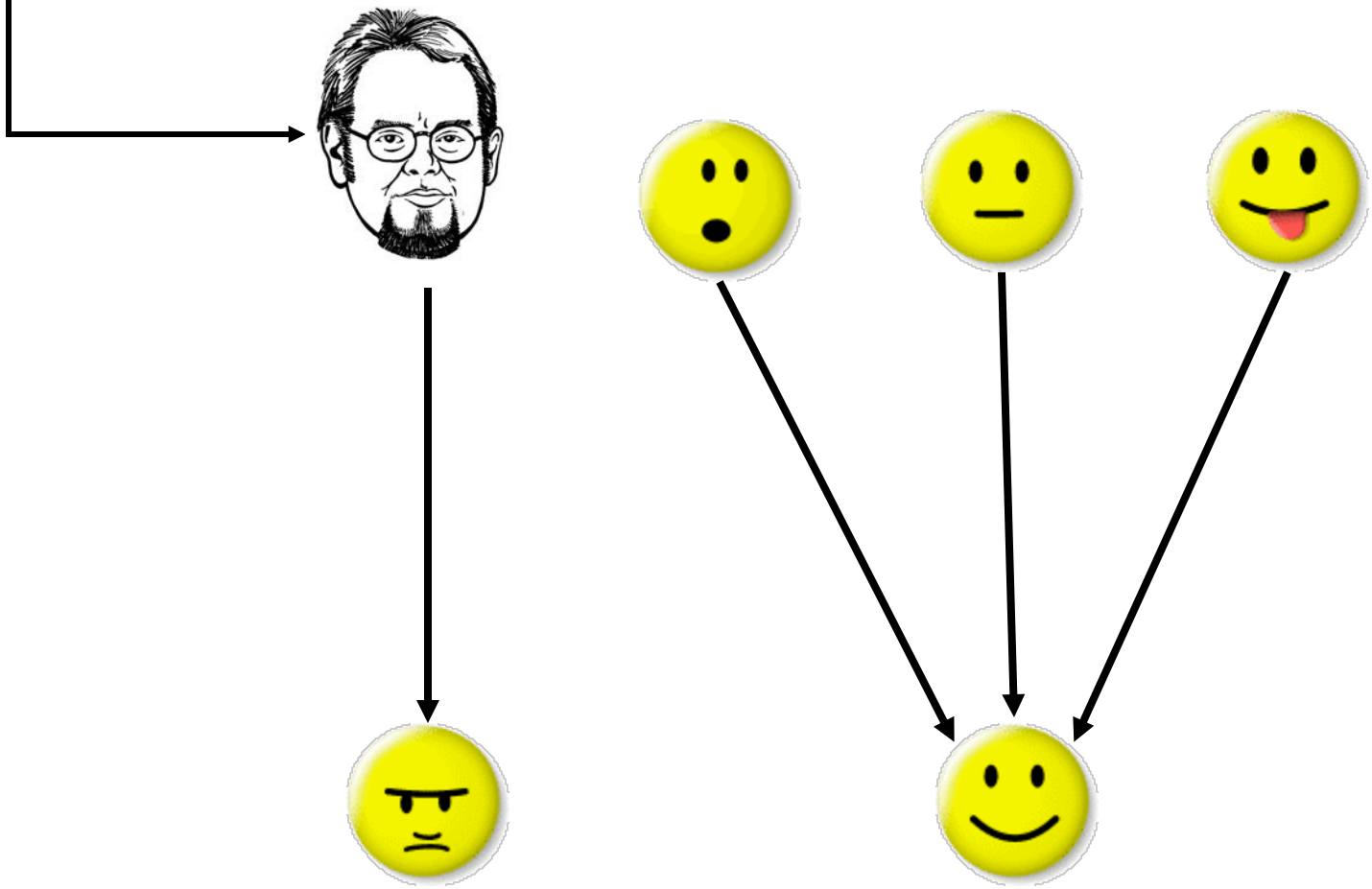
Interviewing



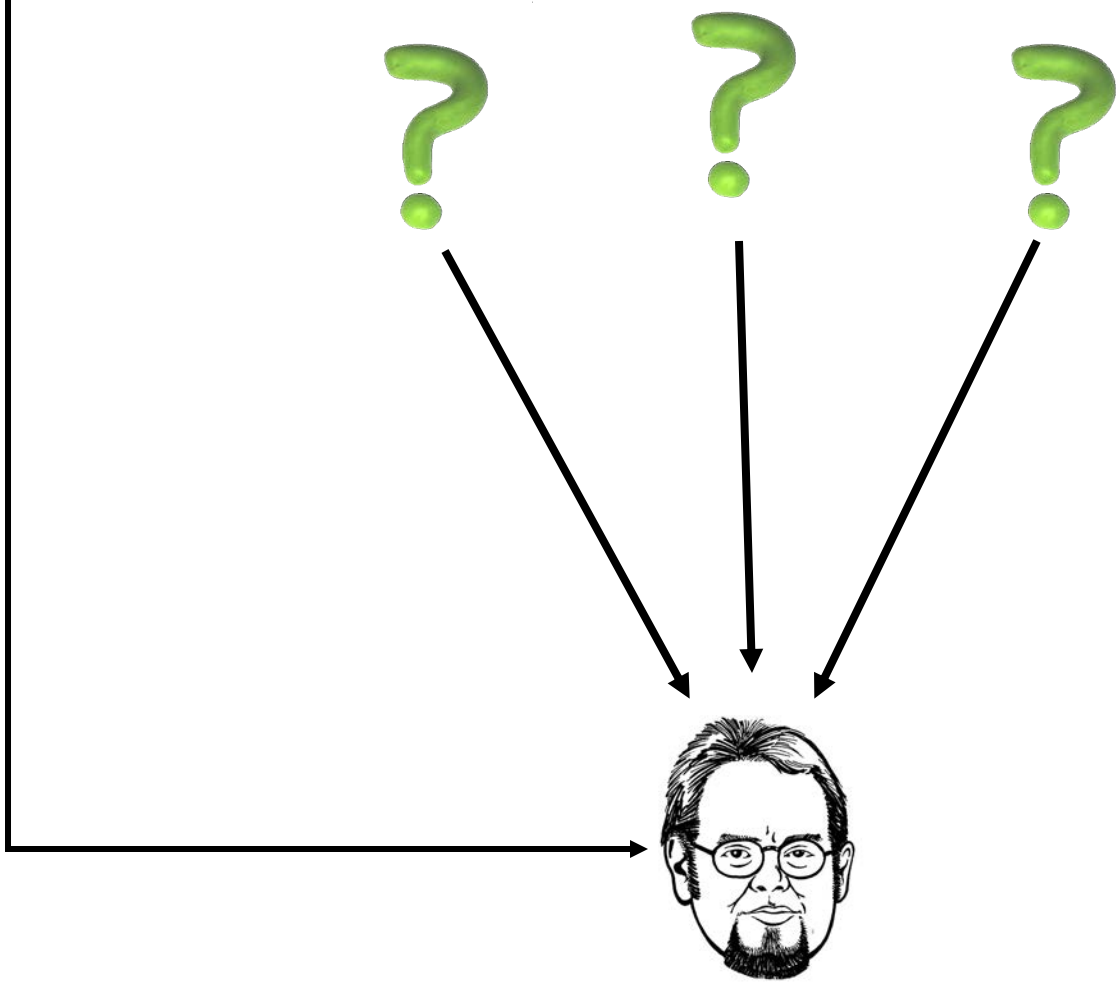
Wait – How do we know that this endorser is important?

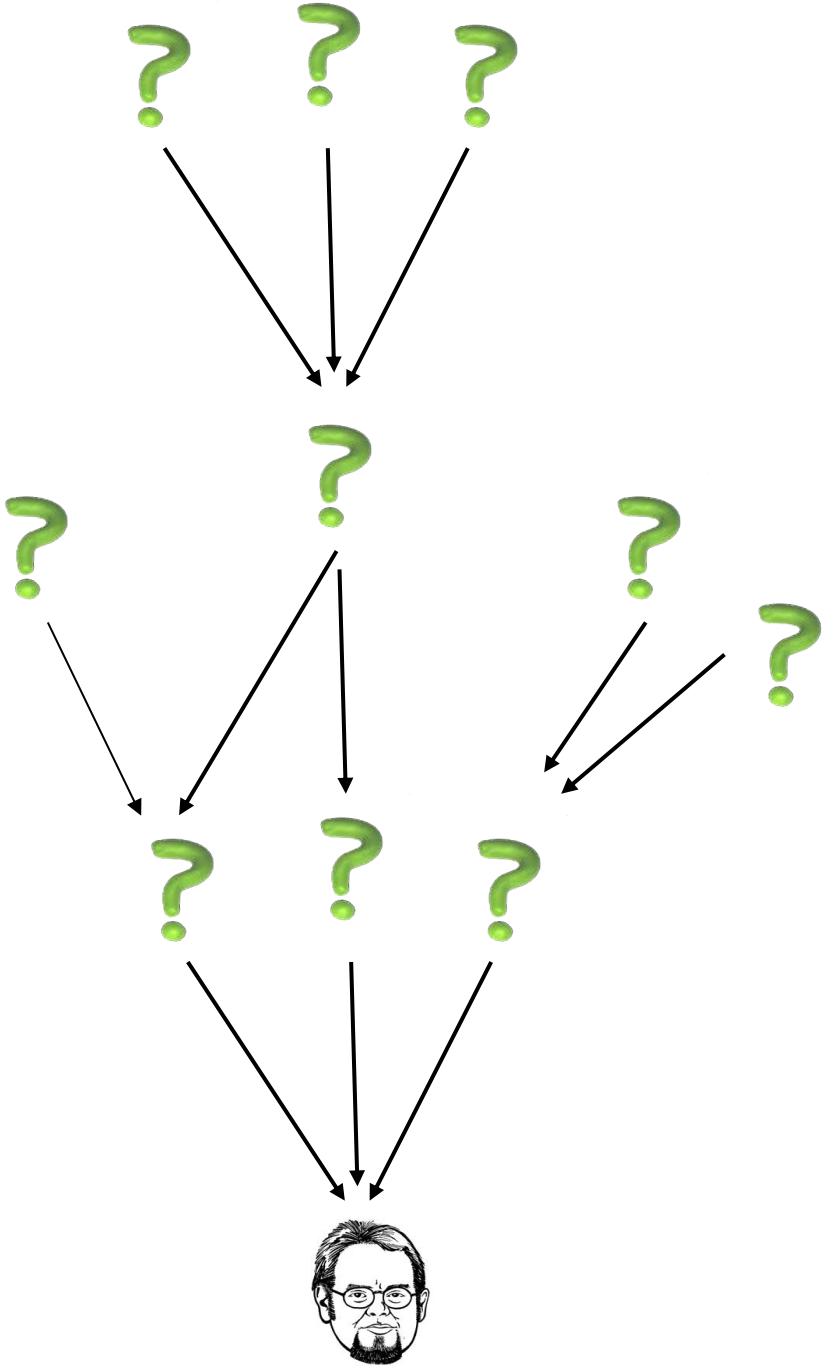


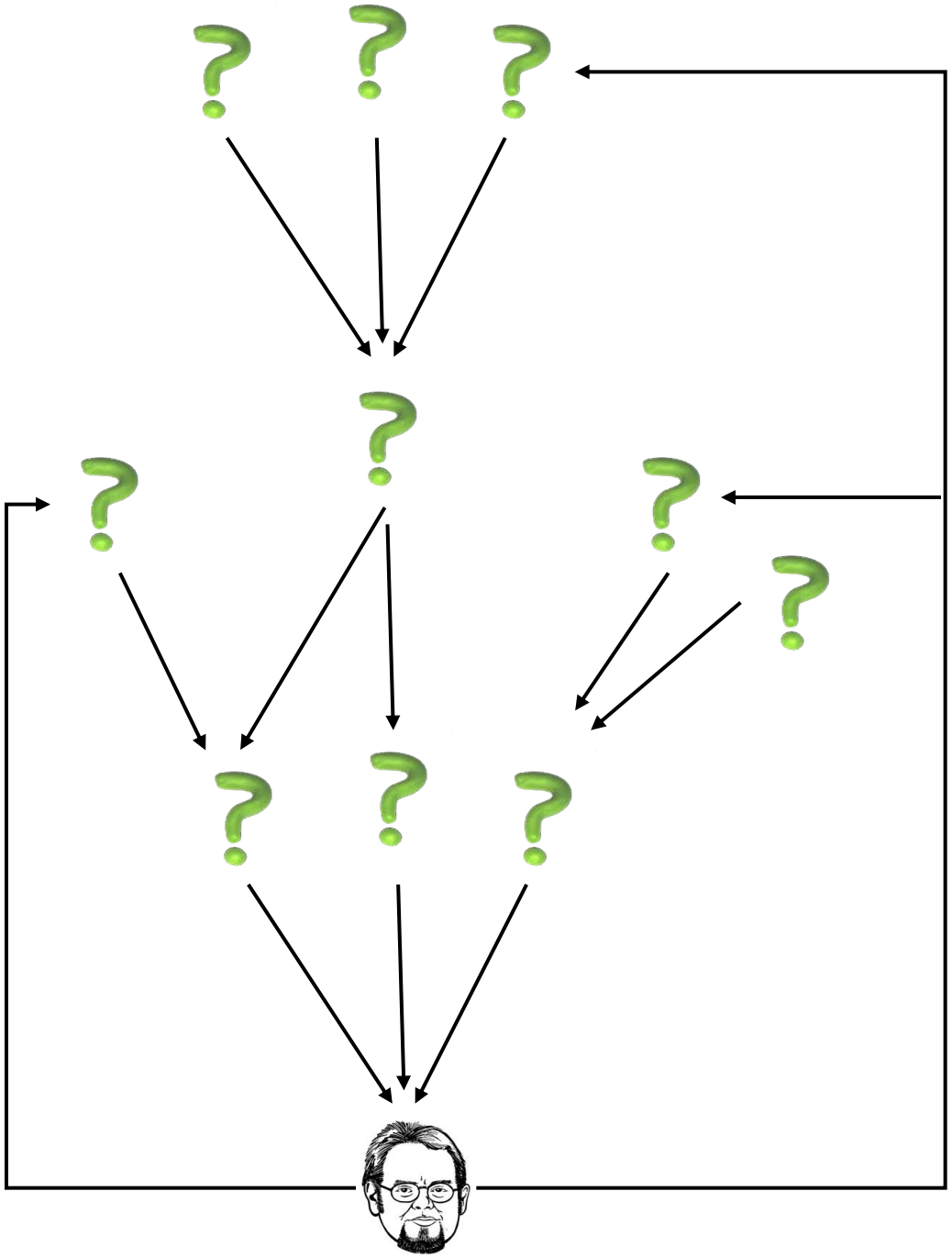
Wait – How do we know that this endorser is important?



Remember – It would depend on his endorsements.

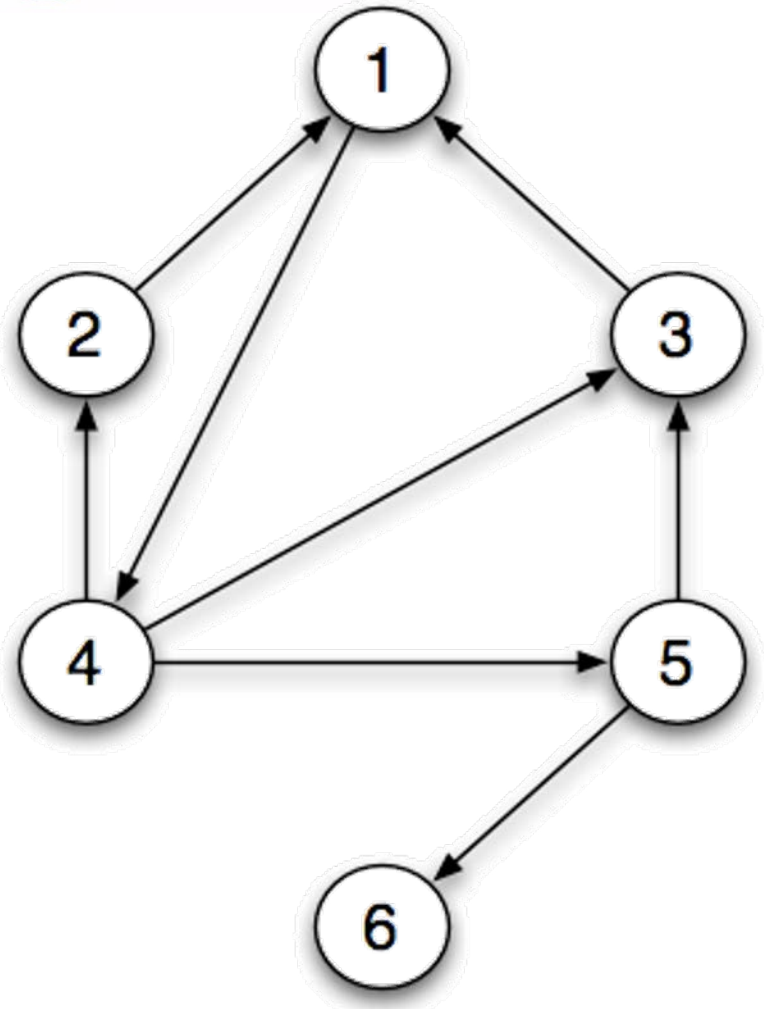






Google

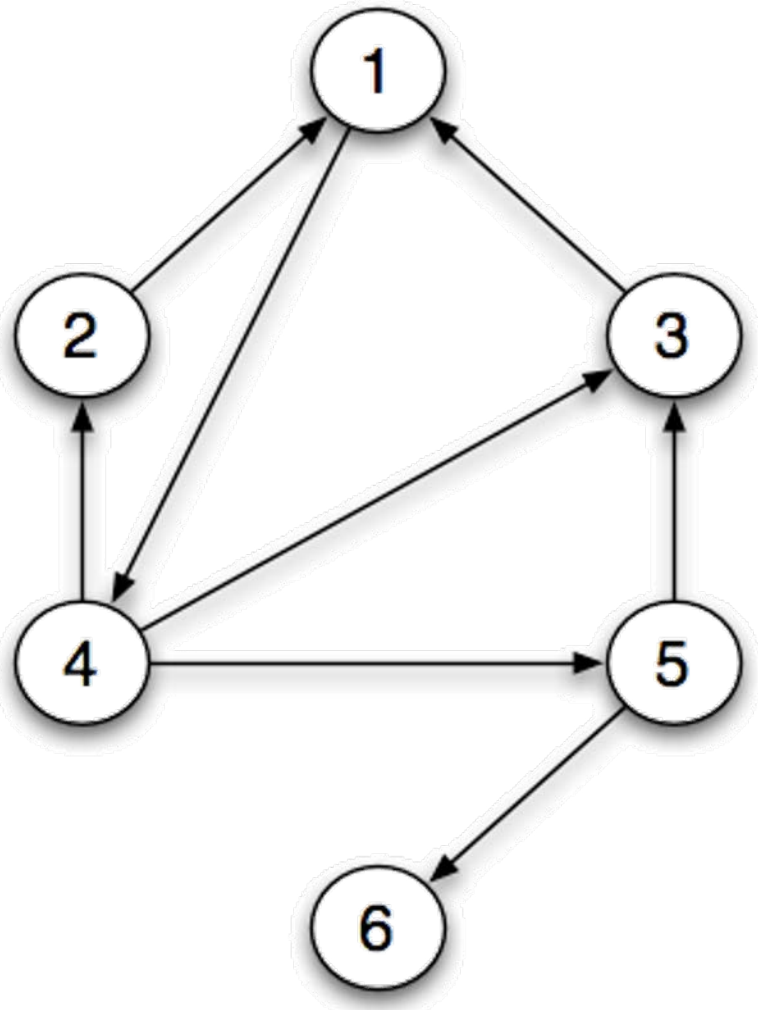
- This represents a small network of 6 web pages as a directed graph.
- Web page 1 links to web page 4.
- Web page 4 links to pages 2, 3, and 5.



Adjacency

- We can store the graph in an adjacency matrix.
- This network would be stored as:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Keep in mind that Google indexes billions of web pages!

Image thanks to David Gleich

Random Surfer

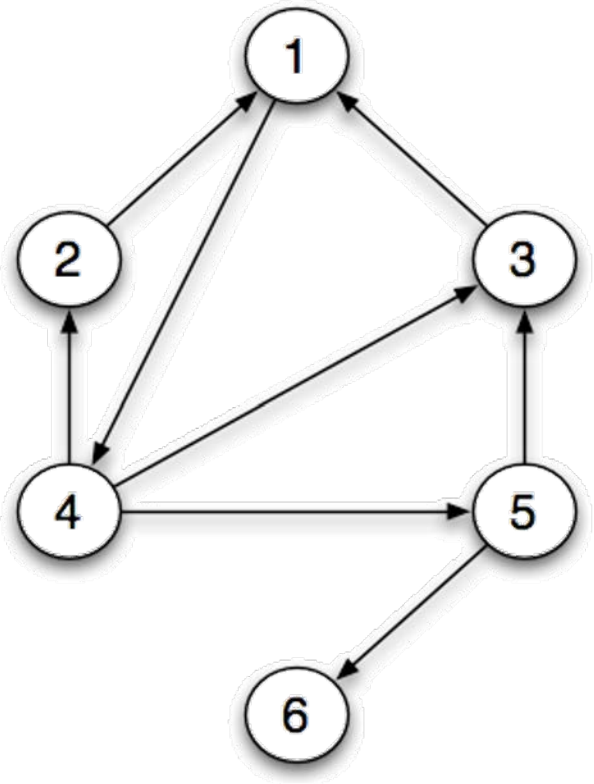
- PageRank measures quality by the hyperlink structure of the web.
- It models internet activity as as the actions of a random surfer who randomly follows links on a web page.

Chance visit

- Suppose a random surfer surfed the web indefinitely.
- The probability the surfer visits a web page is that pages PageRank.
- Higher PageRank correlates to higher quality.

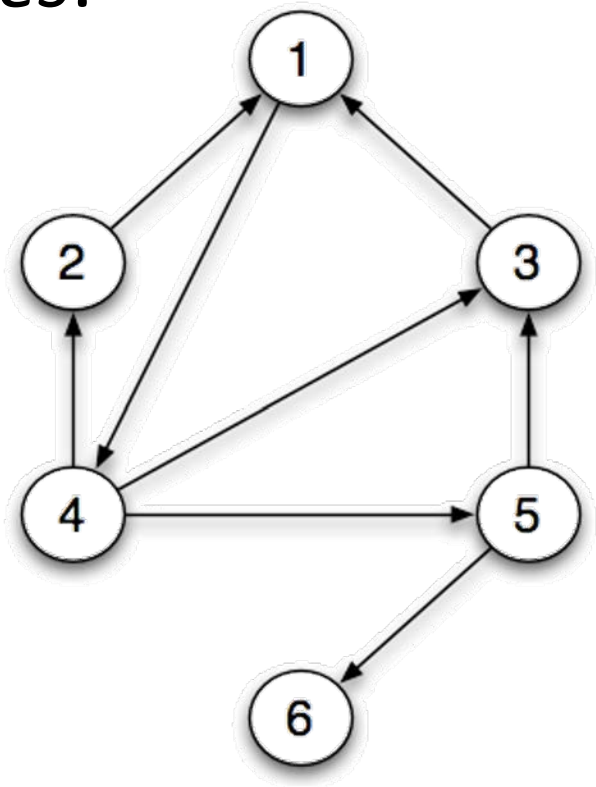
Linked in

- Earlier, we surfed Wikipedia by following links.
- This isn't a realistic model of surfing.
- Why?



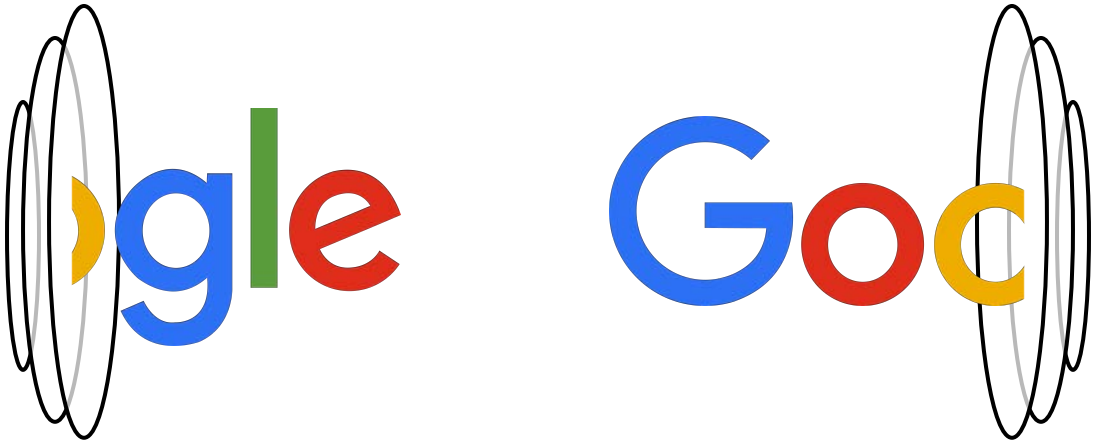
Caught dangling

- If you can't jump, you can get stuck!
- Web pages with no outlinks are called *dangling nodes*.



PageRank model

- If the web page has outlinks:
  - 85% chance of following a hyperlink on a page
  - 15% chance of teleporting (jumping to) any web page in the network (with uniform probability)
- If the web page is a dangling node:
  - uniform probability of teleporting to any web page in the network

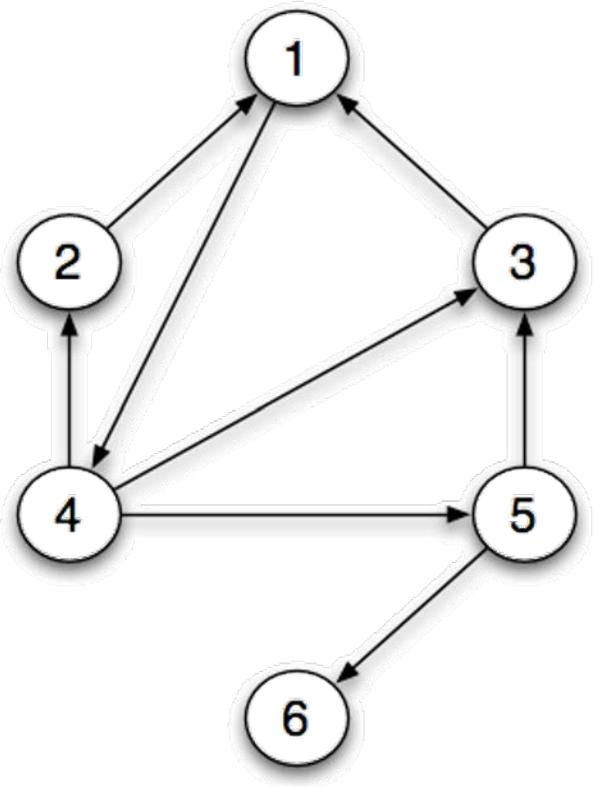


Helpful?

- Do you surf like this? I don't!
- Could this actually work and give useful results?
- It's a valid question asked of Google's founders.
- Note, simple models can give powerful insight.

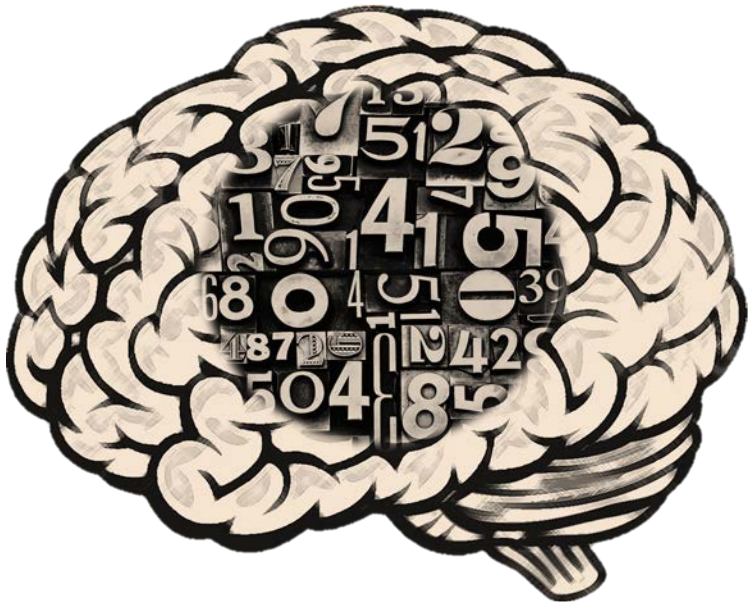
Google in Monte Carlo

We can use Monte Carlo simulation to determine the quality of pages.



Keep in mind

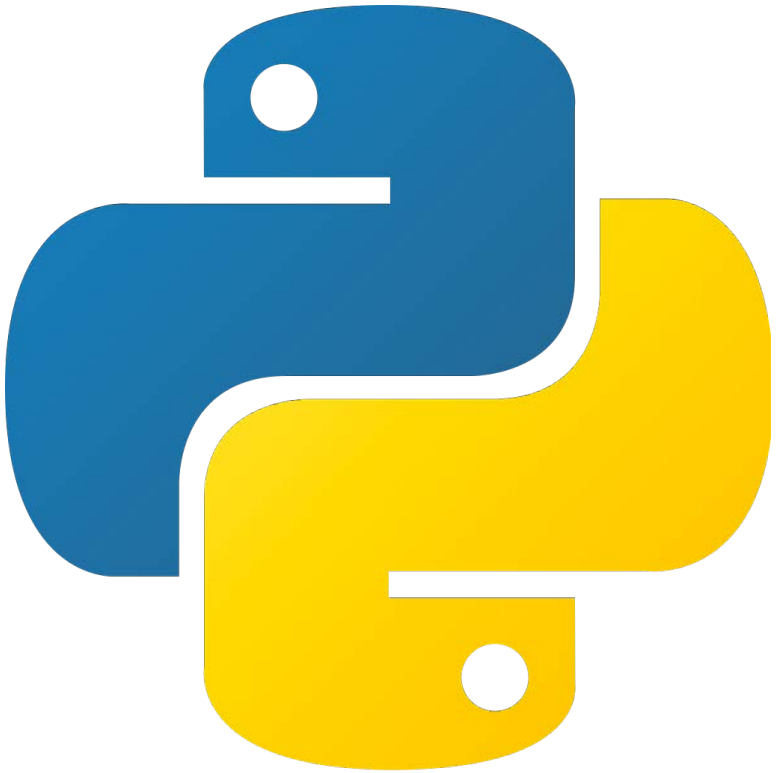
- Monte Carlo methods depend on randomness.
- Things are slightly different between runs.
- Repeat experiment or double run size to see if values have converged.



Code it!

Let’s see this in python code.

simulateGooglePython.ipynb



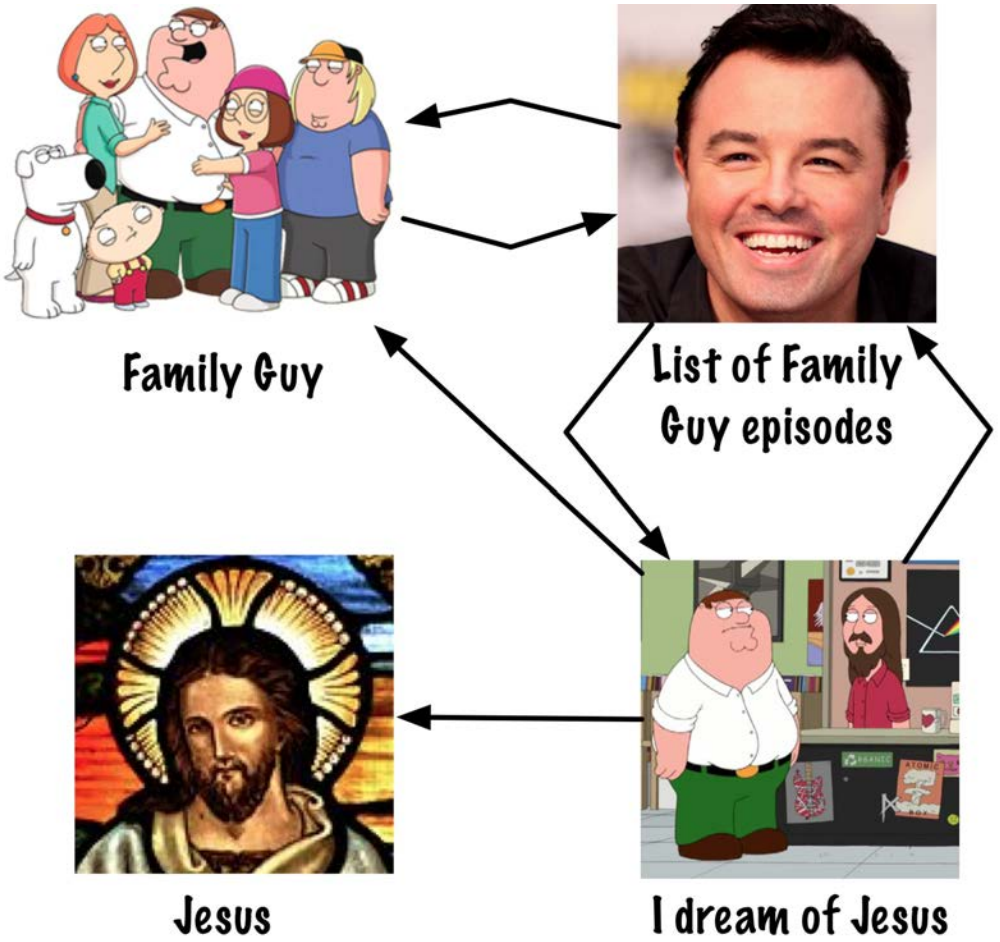

WAIT
is this really
how it's done?

Thinking linear

- The PageRank algorithm uses linear algebra.
- In fact, it uses math ideas developed 100 years ago.

Wiki-Jesus

- Let's return to the 5-clicks to Jesus exercise.
- Here is a path from the Family Guy to Jesus pages on Wikipedia.
- Let's consider Google's model constrained only to this network.

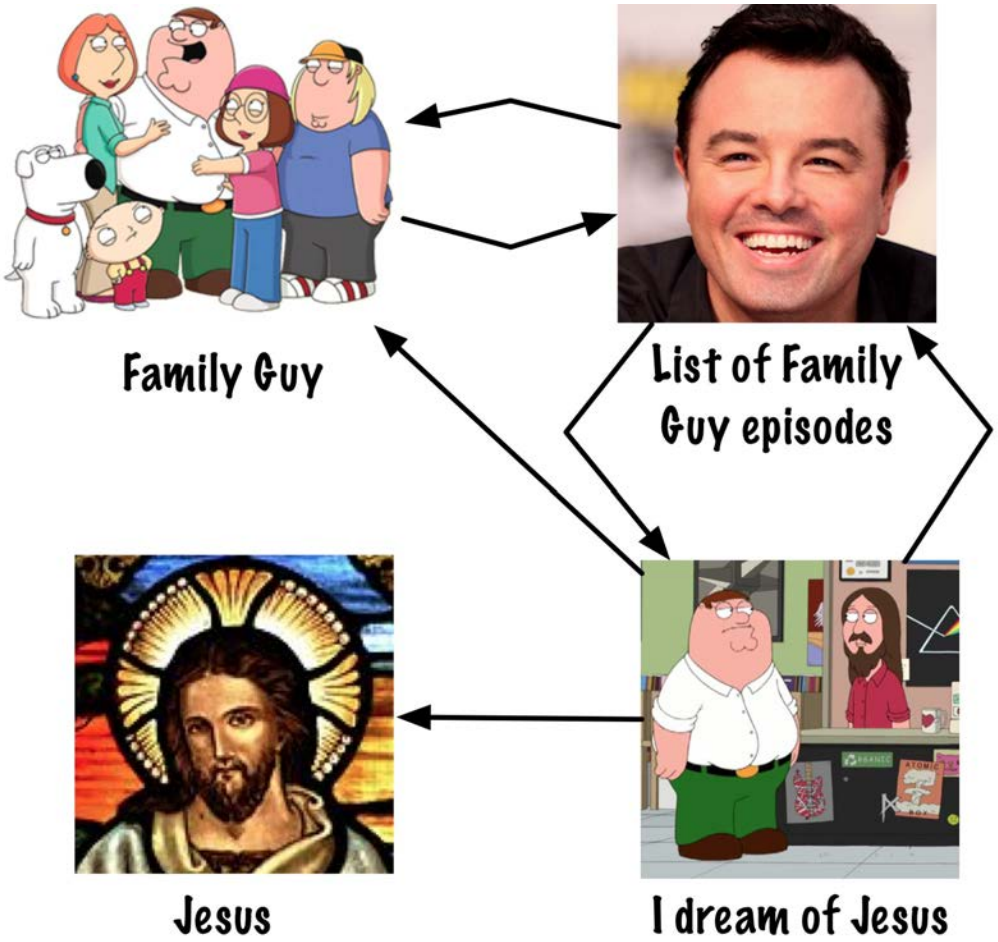


Probable surfing

Under Google's model, if you are at the *Family Guy* web page, what is the probability of:

- visiting the page listing episodes?

- visiting Jesus?



Probable surfing

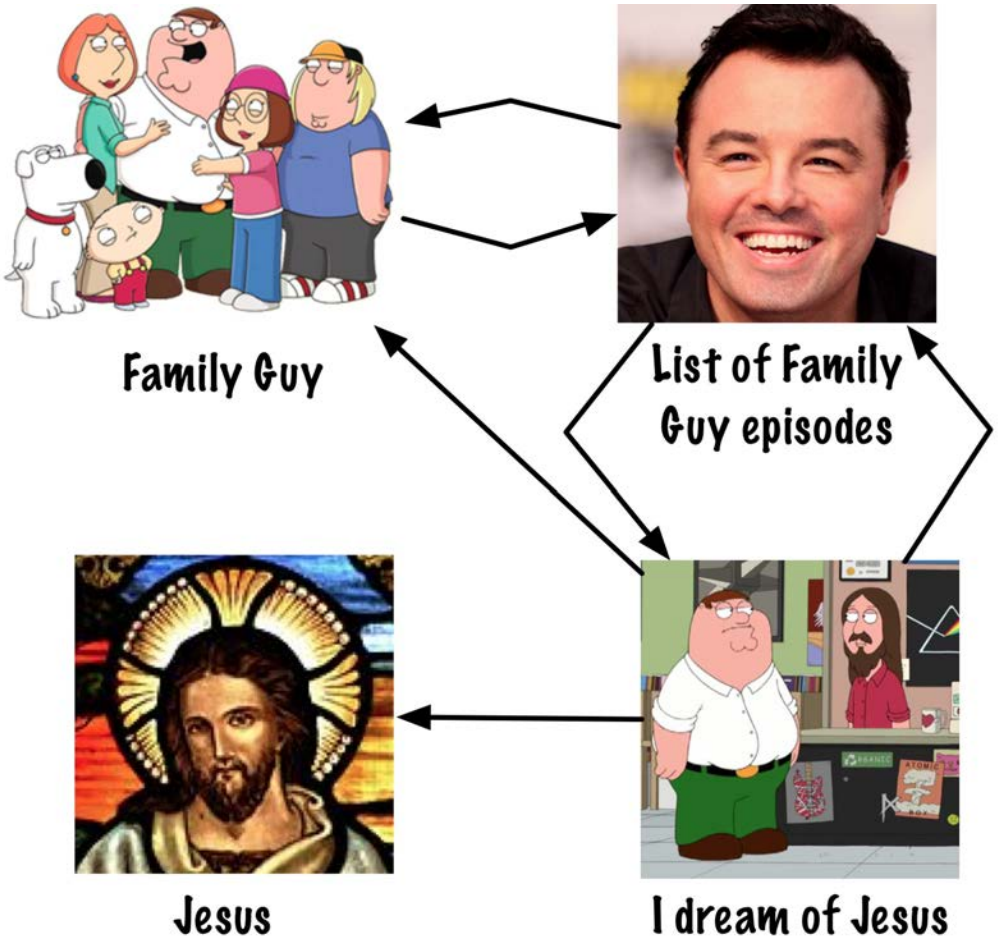
Under Google's model, if you are at the *Family Guy* web page, what is the probability of:

- visiting the page listing episodes?

  .85 + .15/4 = .8875

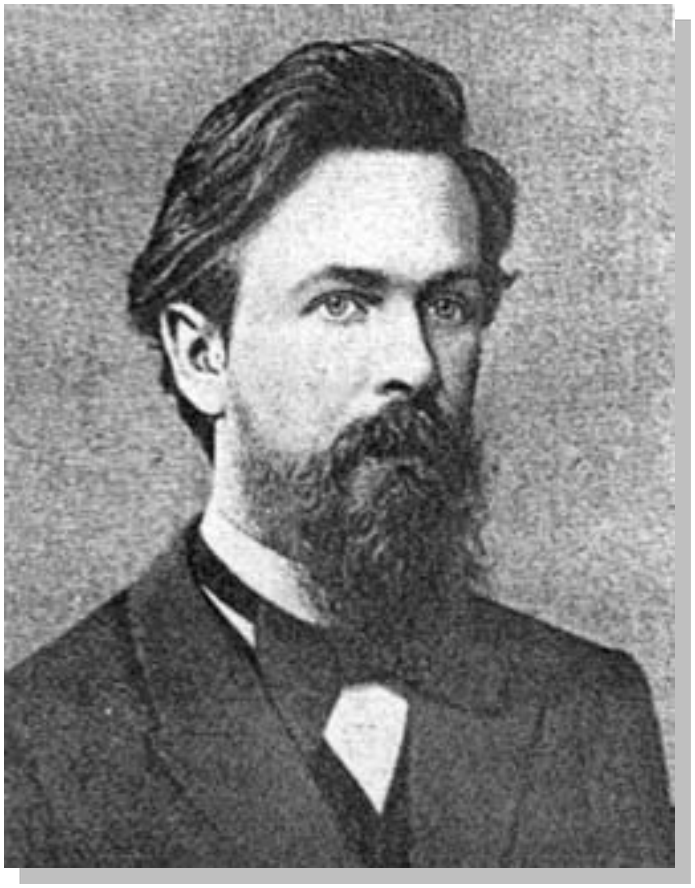
- visiting Jesus?

  .0375



Leaning on Markov

- Finding the probability of visiting web page *j* from web page *i* allows us to use Markov Chains (processes).
- First used for linguistic purposes to model the letter sequences in works of Russian literature.



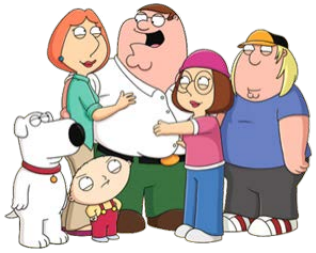
Andrei Andreevich Markov
(1856 - 1922)

Enter the matrix

We create a transition matrix G where g_{ij} equals the probability of moving from web page i to web page j.

Time to Order

First, order the columns (and rows)

(   )

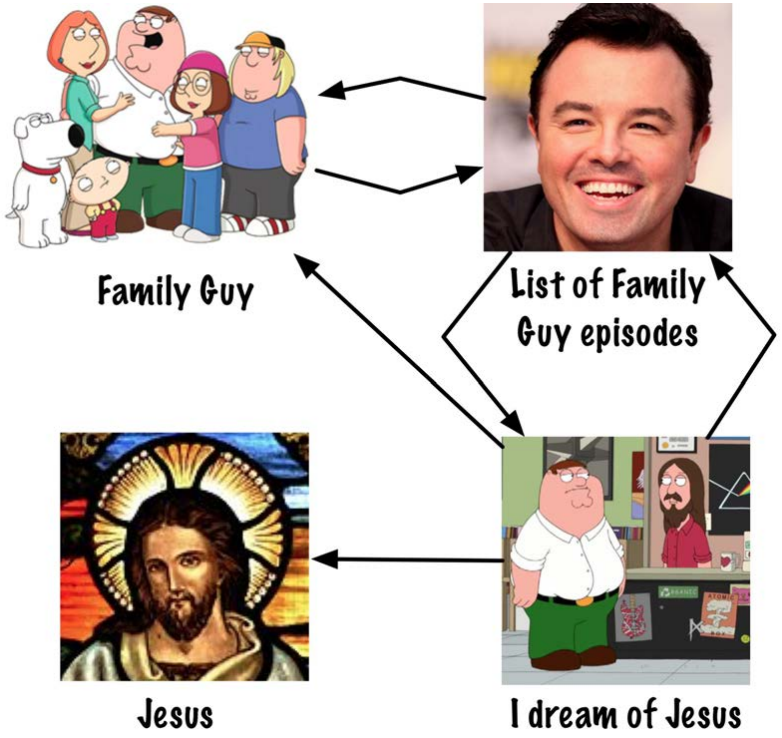
column 1 column 2 column 3 column 4

Row

The first row contains the probabilities of jumping from web page 1 to other web pages.

$$\begin{pmatrix} 0.0375 & 0.8875 & 0.0375 & 0.0375 \end{pmatrix}$$

row order



row, row, row

So, the entire transition matrix becomes:

$$G = \begin{pmatrix} 0.0375 & 0.8875 & 0.0375 & 0.0375 \\ 0.4625 & 0.0375 & 0.4625 & 0.0375 \\ 0.3208 & 0.3208 & 0.0375 & 0.3208 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 \end{pmatrix}$$




Note: The entries of each row sum to 1.

baby steps

- We can then walk through a series of steps.
- Assume we start at *Family Guy*, then

$$\begin{aligned}
\mathbf{v}_0 G &= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0.0375 & 0.8875 & 0.0375 & 0.0375 \\ 0.4625 & 0.0375 & 0.4625 & 0.0375 \\ 0.3208 & 0.3208 & 0.0375 & 0.3208 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 \end{pmatrix} \\
&= \begin{pmatrix} 0.0375 & 0.8875 & 0.0375 & 0.0375 \end{pmatrix} \\
&= \mathbf{v}_1
\end{aligned}$$

The one step

- Since

$$\begin{aligned}
\mathbf{v}_0 G &= \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0.0375 & 0.8875 & 0.0375 & 0.0375 \\ 0.4625 & 0.0375 & 0.4625 & 0.0375 \\ 0.3208 & 0.3208 & 0.0375 & 0.3208 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 \end{pmatrix} \\
&= \begin{pmatrix} 0.0375 & 0.8875 & 0.0375 & 0.0375 \end{pmatrix} \\
&= \mathbf{v}_1
\end{aligned}$$

- We know the probability of being at each web page after one step assuming we start at web page 1.

Step by step

- Where will you be after two steps?

$$\mathbf{v}_1 G = \begin{pmatrix} 0.0375 \\ 0.8875 \\ 0.0375 \\ 0.0375 \end{pmatrix}^T \begin{pmatrix} 0.0375 & 0.8875 & 0.0375 & 0.0375 \\ 0.4625 & 0.0375 & 0.4625 & 0.0375 \\ 0.3208 & 0.3208 & 0.0375 & 0.3208 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 \end{pmatrix}$$

$$= \begin{pmatrix} 0.4333 & 0.0880 & 0.4227 & 0.0561 \end{pmatrix}$$

$$= \mathbf{v}_2$$

- But, how do we find the probability of being at each web page after infinitely many steps?

Iterating

Note that:

$$v_2 = v_1 G = v_0 G^2,$$

$$v_3 = v_2 G = v_0 G^3,$$

$$\vdots$$

$$v_n = v_{n-1} G = v_0 G^n,$$

Lotsa steps

So, let's take many more steps:

$$\mathbf{v}_0 G^{100} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}^T \begin{pmatrix} 0.0375 & 0.8875 & 0.0375 & 0.0375 \\ 0.4625 & 0.0375 & 0.4625 & 0.0375 \\ 0.3208 & 0.3208 & 0.0375 & 0.3208 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 \end{pmatrix}^{100}$$

$$= \begin{pmatrix} 0.2836 & 0.3682 & 0.2210 & 0.1271 \end{pmatrix}$$

$$= \mathbf{v}_{100} = \mathbf{v}_{200} \text{ (to 4 decimal places)}$$

We have converged to the *steady-state vector.*

Steady

- In fact, for this vector, we reach steady state (to 4 decimal places) at the 18th step, which will be very important to Google.
- This gives us the PageRank of these pages:

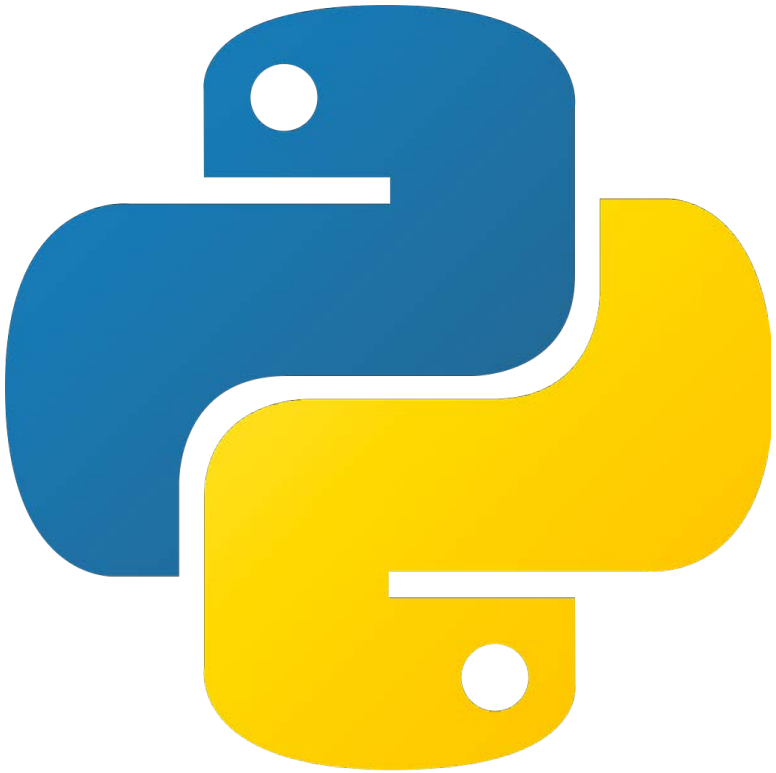
$$\mathbf{v} = \begin{pmatrix} 0.2836 & 0.3682 & 0.2210 & 0.1271 \end{pmatrix}$$




Code it!

Let’s see this in python code.

markovGooglePython.ipynb



Questions!

- Will this process converge for any network of web pages?
- Is there more than one steady-state vector?
- Will this scale up to billions of pages?

Unique solution

- Will the Markov process converge?
- Further, is the solution unique?
- Both are guaranteed for PageRank.

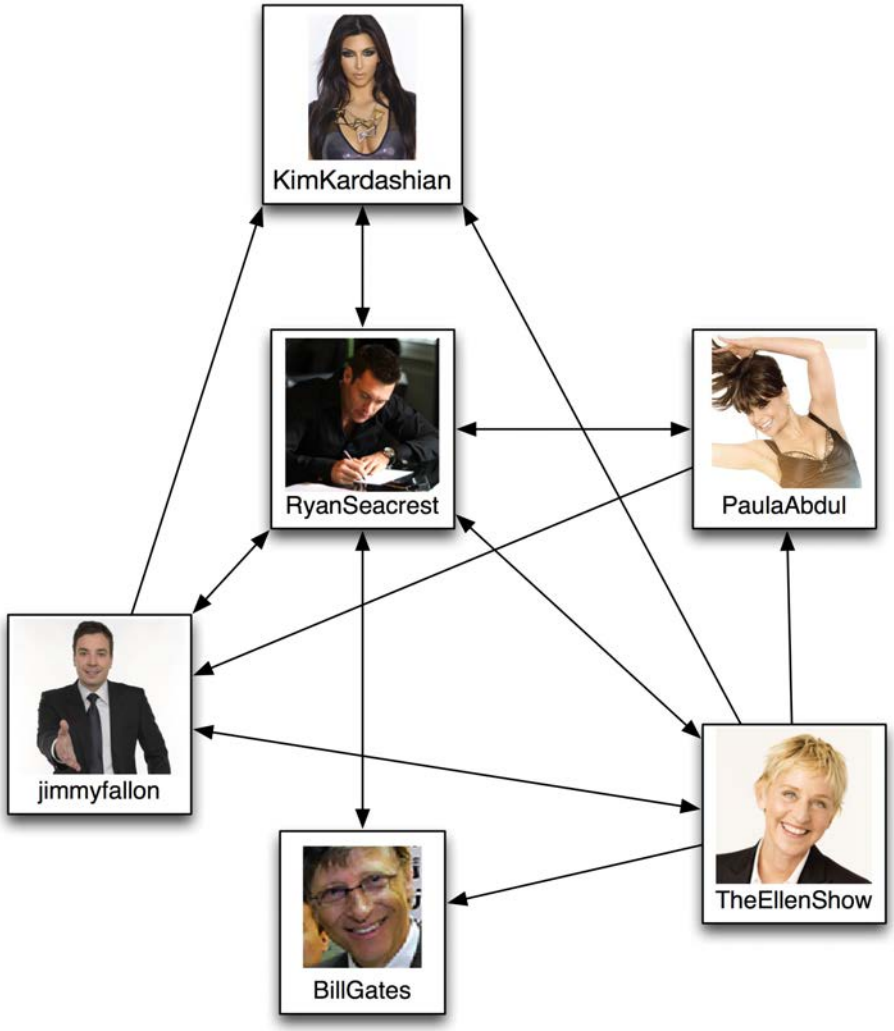
Theorem (Perron) Every real square matrix *P* who entries are all positive has a unique eigenvector with all positive entries, its corresponding eigenvalue has multiplicity one, and it is the dominant eigenvalue, in that every other eigenvalue has strictly smaller magnitude.

Answers!

- Will this process converge for any network of web pages?
- Is there more than one steady-state vector?
- Will this scale up to billions of pages?

Graphic Celebrities

- Here is the graph of connectivity of the celebrities on Twitter from a number of years ago.
- There is an edge from celebrity A to celebrity B if celebrity A follows celebrity B on Twitter.



Google matrix

First, we form the Google matrix:

$$G = \begin{pmatrix} 0.025 & 0.025 & 0.0250 & 0.025 & 0.8750 & 0.0250 \\ 0.025 & 0.025 & 0.3083 & 0.025 & 0.3083 & 0.3083 \\ 0.025 & 0.025 & 0.0250 & 0.025 & 0.8750 & 0.0250 \\ 0.025 & 0.450 & 0.0250 & 0.025 & 0.4500 & 0.0250 \\ 0.195 & 0.195 & 0.1950 & 0.195 & 0.0250 & 0.1950 \\ 0.195 & 0.195 & 0.1950 & 0.195 & 0.1950 & 0.0250 \end{pmatrix}$$

Tweetable PageRank

Iterating:

$$\mathbf{v}_{k+1} = \mathbf{v}_k M,$$

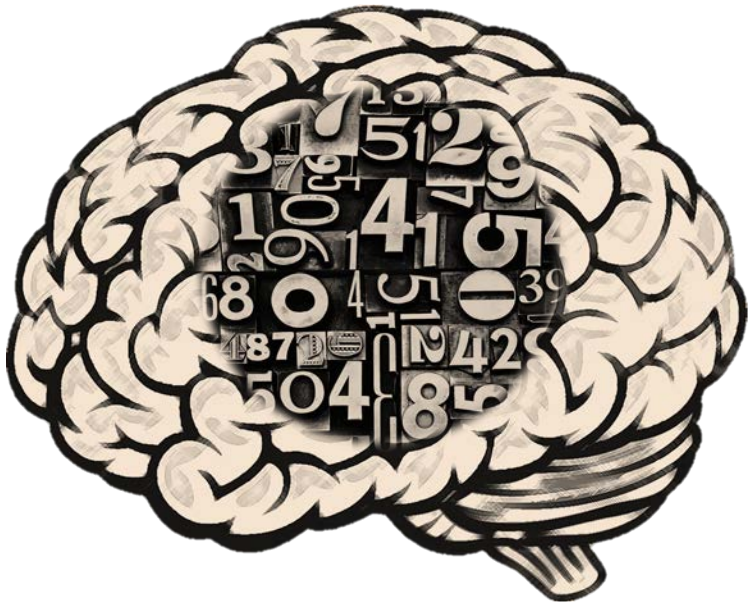
until the elements of $\mathbf{v}_k$ have suitably converged.

$$\mathbf{v} = \begin{pmatrix} 0.1071 & 0.1526 & 0.1503 & 0.1071 & 0.3544 & 0.1285 \end{pmatrix}$$

Keep in mind

- PageRank tends to be unstable in the “tail”.
- With ranking, you often want to show it is a good ranking.
- Note, you need a link/connection between entities for ranking.
- I often think,

  “What’s the game?



Further exploration

- Want to dive further into this topic?
- Here are a few ideas…

Picture credit: http://kevinklau.com/wp-content/uploads/2011/10/explore3.jpg

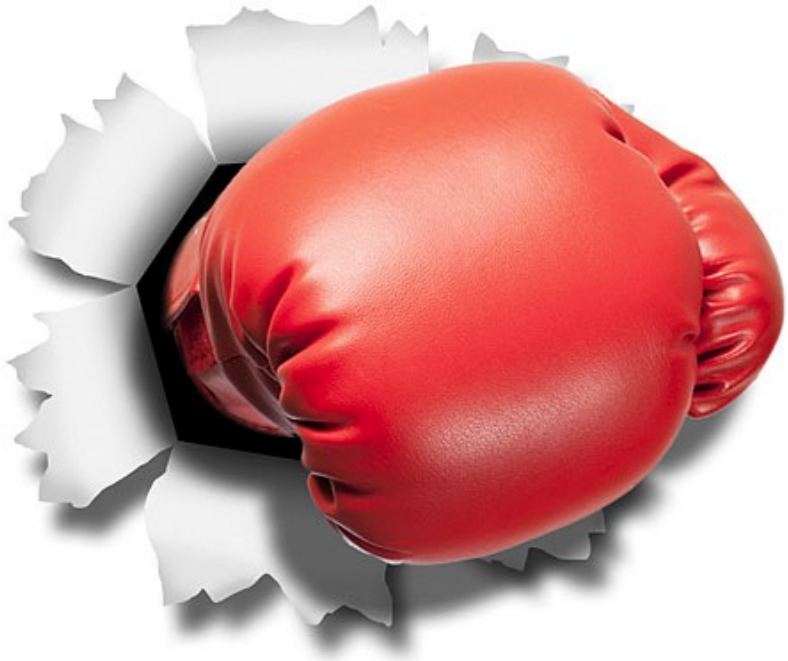
Teleportation

- Earlier, we took the teleportation parameter to equal 0.85.
- Change this value so it is closer to 1. Then, change it so it is closer to 0.
- What impact does this have on convergence? What impact does it have on the ranking?

Picture credit: http://www.blavish.com/wp-content/uploads/2006/10/star-trek-teleport-10-5-2006.jpg

HITS

- HITS is an alternative algorithm for ranking.
- Implement this algorithm, that also uses linear algebra.
- How do the results vary from PageRank?



Picture credit: http://www.photoshopessentials.com/images/photo-effects/punch-through/photoshop-punch-through-image.jpg

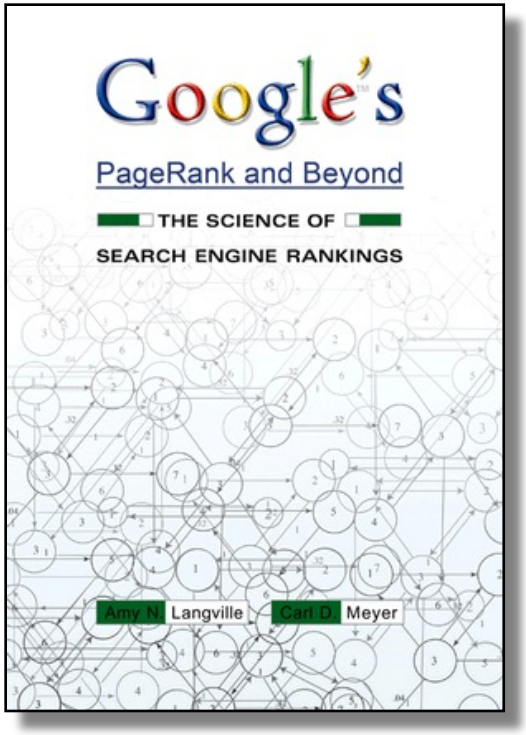
Application

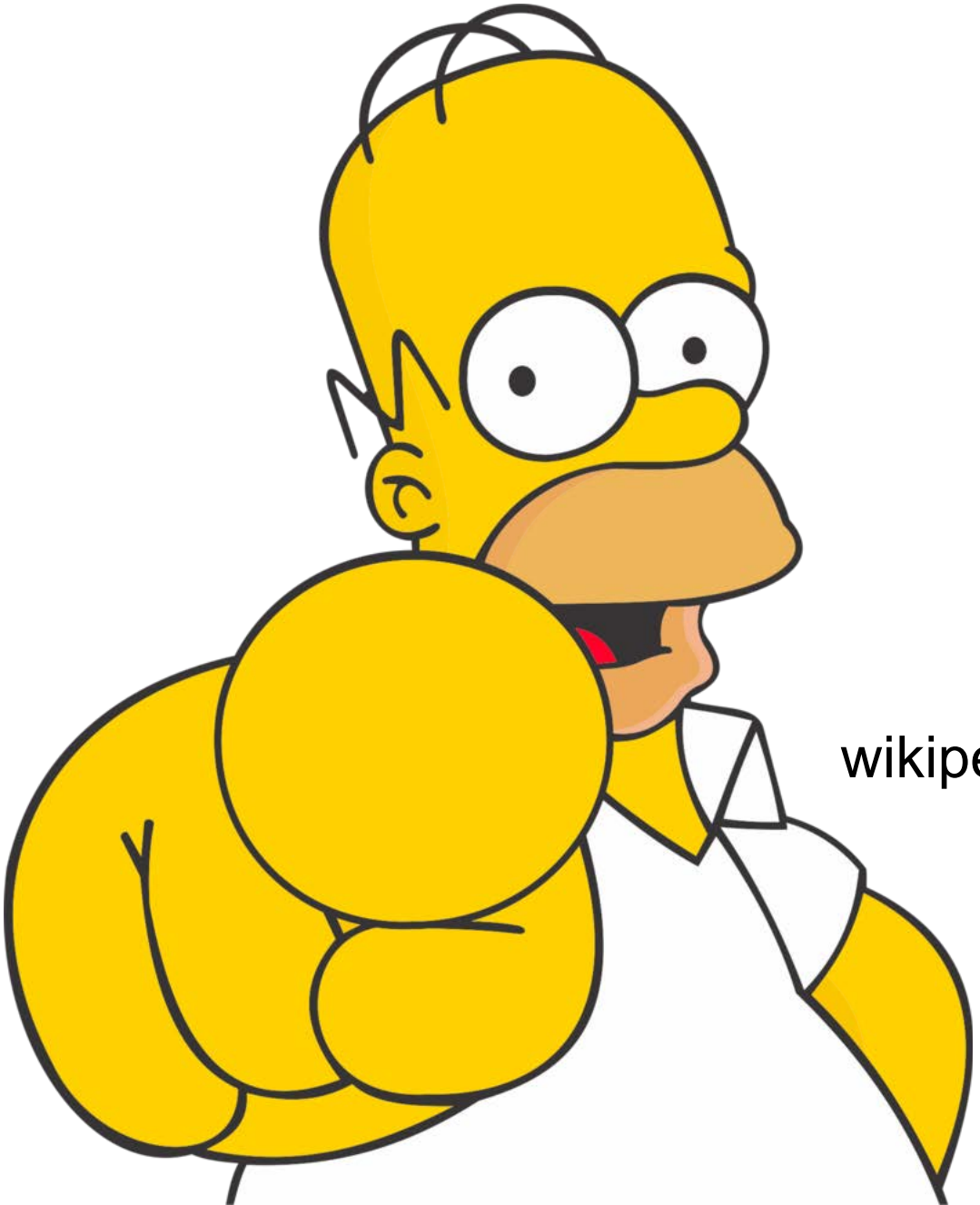
- Applying PageRank to other networks from fields such as biology or archeology.
- Do you have a directed graph? Could PageRank be applied?
- You could even try sports ranking!

Picture credit: http://www.inspirecusd.org/misc_images/000/000/000/170/ApplyNowButton.jpg

Scalability

- How does PageRank scale up to billions of pages?
- A key is expressing the Google matrix in a different form so you only store the (sparse) adjacency matrix and a vector.
- Else, you store an *n* x *n* dense matrix.





Code 4 u

WIKIPEDIA
The Free Encyclopedia

wikipediaPageRankPython.ipynb

Homework – 1

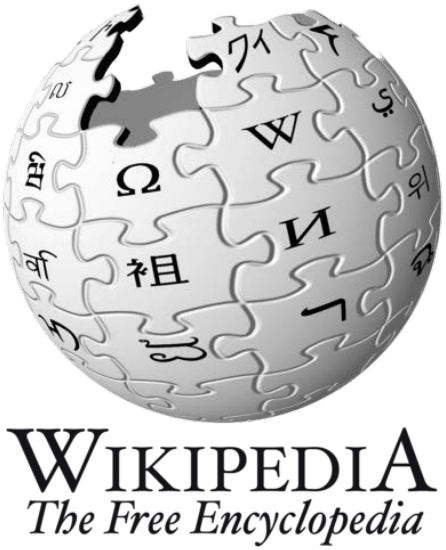
- Reading: *Why High-School Rankings Are Meaningless – and Harmful* (if you don't have access, let me know and I can print it)
- You will be asked for 2-3 paragraphs on your thoughts on this article.
- Rankings are not always helpful.

Homework – 2

- Write the Google matrix for a small network.
- Use our codes to compute the PageRank vector of the network. (Use the Markov version of our codes.)

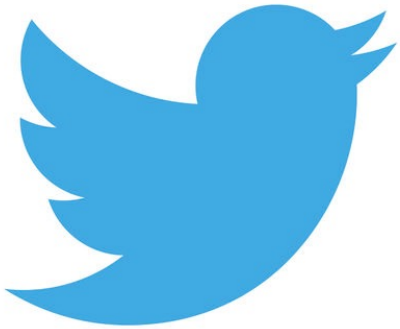
Homework – 3

Create your own Wikipedia example using the codes from the class.



Homework – 4

- Think of an application where you can create a network of at least 5 entities.
- Create the adjacency network and compute the PageRank of your network.



Homework – 5

What are your current thoughts regarding your research? After seeing PageRank, what comes to mind? Do you have questions that will help you think about research into a ranking question? If what, what are your questions?