

INFO-H-509 XML Technologies

Searching and Ranking

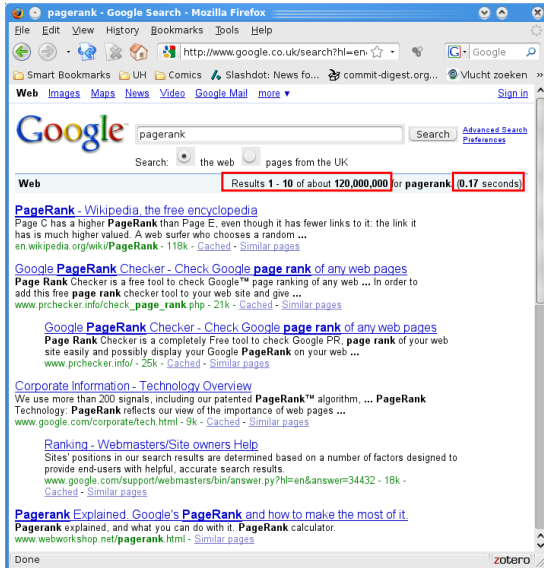
Stijn Vansummeren

May 6, 2010

Navigating the Web

- Directly via a known URL (e.g., `http://www.apple.com/ipad/`)
- By following links from a known web page
- By using a search engine

Searching the web is like searching a needle in a haystack ...




How do they do this?

(show Google video)

A Remark


Search Engines actually only search part of the web!



Search: ☒ the web ☐ pages from Belgium

Web [Show options...](#) Results 1 - 10 of about **102,000,000** for **ipad**. (0.08 seconds)

[Buying an iPad?](#)
[www.MyUS.com](#) We Forward US Items Internationally Ship Quickly, Cheaply & Safely.

[News results for ipad](#)

[BBC News](#)
[Amazon acknowledges threat Apple's iPad poses to Kindle 2](#) - 7 hours ago
It stands to follow then that as orders continue to pour in for the iPad, Apple will use some of that same closed ecosystem mindset to make it equally as
...
[Tainted Green - 738 related articles »](#)
[iPad: the giant iPhone - Technique - 183 related articles »](#)
[Adobe abandons iPhone code tools - BBC News - 377 related articles »](#)

[Apple - iPad - See the web, email, and photos like never before.](#)
The iPad. With a revolutionary, 9.7 inch touch screen, and amazing new apps, it does things no tablet PC, netbook, or e-reader could. Starts at \$499.
[www.apple.com/ipad/](#) - United States - [Cached](#)
[Features](#) [Design](#)
[Gallery](#) [iPad Video](#)
[Tech Specs](#) [Developing iPad Apps](#)
[Apps for iPad](#) [Apple \(United Kingdom\) - iPad](#)
[More results from apple.com »](#)

[Apple - iPad - The best way to see the web, email, and photos.](#)

[De iPad in uw bedrijf ?](#)
iPad ondersteuning, installatie, configuratie. Advies en support
[www.1support.be](#)

[Protect for your NEW iPad](#)
Protect your \$500+ investment against drops and damage.
[www.amazon.com/](#)

[New 3G iPad debuts 4/30](#)
iPad Wi-Fi + 3G go on sale
Get the details here.
[www.DNAinfo.com/Tech](#)

[Nouvelle iPad 16Go Wi-Fi](#)
Vient de lancer Produit authentique
Expédition en France. Prix €572
[www.comeuro.net](#)

[Acheter aux Usa](#)
Profitez du taux du Dollar pour faire votre shopping aux Usa

A Remark

Search Engines actually only search part of the web!

The image shows two overlapping search engine result pages. The background page is Google, and the foreground page is Yahoo.

Google Search Results:

- Search bar: "ipad"
- Buttons: "Search", "Advanced Search"
- Search scope: "the web" (selected), "pages from Belgium"
- Results: "Results 1 - 10 of about **102,000,000** for **ipad**. (0.08 seconds)"
- Sponsored Link: "Buying an iPad? www.MyUS.com We Forward US Items Int..."
- News results for **ipad**:
 - Amazon acknowledges threat. It stands to follow then that as will use some of that same clo...
 - Tainted Green - 738 related articles
 - Adobe abandons iPhone code
- Apple - iPad - See the web, email, and The iPad. With a revolutionary, 9.7 inch touch no tablet PC, netbook, or e-reader could. Start www.apple.com/ipad/ - United States - [Cached](#)
- Links: Features, Design, Gallery, iPad Video, Tech Specs, Developing iPad Apps, Apps for iPad, Apple (United Kingdom)
- More results from apple.com
- Apple - iPad - The best way to see the

Yahoo! Search Results:

- Search bar: "ipad"
- Buttons: "Web", "Images", "Video", "Local", "Shopping", "News", "More"
- Sponsored Results:
 - ipad™ Screen Protector**
Buy the invisibleSHIELD™ Keep your iPad Screen Scratch Free.
www.ZAGG.com/ipad
- IPad - Latest News**
 - News, Photos, Videos, Twitter
 - 26 Percent of Wired's Mobile Traffic Comes From iPad**
- Wired News - 16 minutes ago
Just three weeks after its launch, the iPad already accounts for almost 1 percent of Wired.com's overall traffic -- and more than a quarter of our mobile traffic. [full story](#)
 - Pinnball HD released for iPad** - Macworld.com via Yahoo! News - 10 hours ago
 - The iPad: The World's First Couchtop** - Forbes - Apr 22 10:49am
 - Get multitasking -- with limits -- from Desktop for iPad** - Macworld.com via Yahoo! News - Apr 22 10:15am
 - [more iPad news...](#)
- Search Pad: 2,410,000,000 results for **ipad:**
- Links: Show All, Apple, CNET Reviews, Wikipedia, Engadget, TUAW, Shopping Sites

The Internet in some Numbers

How many websites are there?

- April 2010: 205 million sites (according to netcraft.com)

To compare:

- 205 million seconds = 6.5 years

The Internet in some Numbers

Each website has multiple web pages

- wikipedia (more than 3 million)
 - La libre belgique
 - The BBC website
 - ...
- Google stopped counting in 2004: **8 billion pages**
 - Yahoo (2005): **19.2 billion \simeq 600 years**

Ranking: the general idea



Query



Rank



1



2



3



Goal:

- Return a **list** of web pages matching the search terms ...
- ... such that the **relevant** pages occur within the first 10 results.

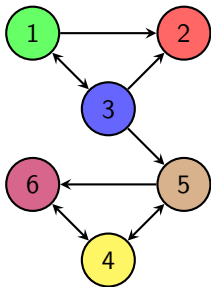
The pre-Google way of ranking

A web page is more relevant w.r.t. a search query if:

- the frequency of a search term in a web page is high;
- a search term occurs in the page's title;
- a search term occurs in bold font;
- ...

By this measure, a web page is important if it says that its important.

Google's idea: Exploit the link structure of the Web



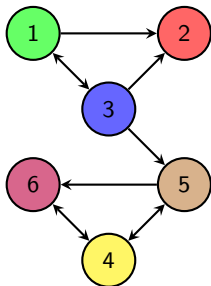
- A link from page *A* to page *B* is a vote by *A* that *B* is “important”
- So a page with many incoming links is more important than a page with few incoming links.
- But votes from “important” pages are more important than votes from “insignificant” pages
- And votes from pages with many outgoing links are less important than votes from pages with few outgoing links.

By this measure, “importance” is a democratic concept, independent of the search term!

PageRank version 1

Definition

The PageRank x_i of a page i is given by $x_i = \sum_{j \in B_i} \frac{x_j}{N_j}$ where B_i is the set of pages linking to i and N_j is the number of outgoing links on page j .



$$\begin{aligned}x_1 &= \frac{1}{3}x_3 \\x_2 &= \frac{1}{2}x_1 + \frac{1}{3}x_3 \\x_3 &= \frac{1}{2}x_1 \\x_4 &= \frac{1}{2}x_5 + \frac{1}{2}x_6 \\x_5 &= \frac{1}{3}x_3 + \frac{1}{2}x_4 \\x_6 &= \frac{1}{2}x_4 + \frac{1}{2}x_5\end{aligned}$$



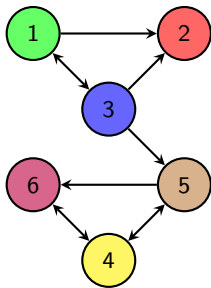
Does such a system always have a **positive solution**?

PageRank version 1 - matrix notation

Definition

Let n be the number of pages in the web graph. The **hyperlink matrix** \mathbf{H} is the $n \times n$ matrix such that

$$\mathbf{H}_{i,j} = \begin{cases} 1/N_j & \text{if } j \text{ links to } i \\ 0 & \text{otherwise} \end{cases}$$



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

$$\mathbf{x} = \mathbf{H} \cdot \mathbf{x}$$

Some linear algebra terminology

Definition

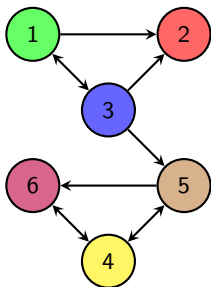
Let \mathbf{A} be a matrix. A number λ and non-zero vector \mathbf{v} satisfying

$$\lambda \mathbf{v} = \mathbf{A} \cdot \mathbf{v}$$

is called an **eigenvalue** and **eigenvector** of \mathbf{A} , respectively.

So searching for a **positive solution** to the equation $\mathbf{x} = \mathbf{H} \cdot \mathbf{x}$ actually means that we're searching for an **eigenvector** of \mathbf{H} with **eigenvalue** 1.

A solution does not always exist

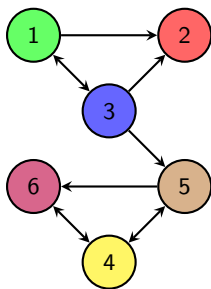


$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

A standard calculation reveals that H 's only eigenvalues are:

$$-\frac{1}{2}, -0.3090169943749474, -0.4082482904638630, \\ 0, 0.4082482904638630, 0.8090169943749474$$

An alternative view on PageRank — Random Surfer



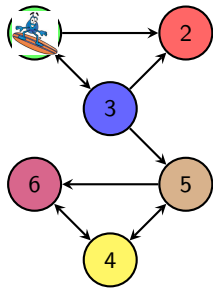
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- The user keeps surfing forever.
- The **probability** p_i that the user visits page i is then given by

$$p_i = \sum_{j \in B_i} \frac{p_j}{N_j}$$

So in essence:

- The equation $\mathbf{x} = H\mathbf{x}$ then asks, for each page, the probability x_i that our random surfer visits page i .
- The higher the probability, the more important the page.

An alternative view on PageRank — Random Surfer



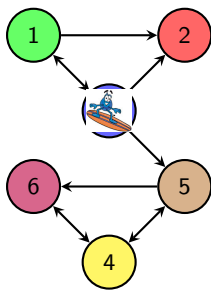
- A user starts surfing by entering a valid URL uniformly at random.
- At each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- The user keeps surfing forever.
- The **probability** p_i that the user visits page i is then given by

$$p_i = \sum_{j \in B_i} \frac{p_j}{N_j}$$

So in essence:

- The equation $\mathbf{x} = H\mathbf{x}$ then asks, for each page, the probability x_i that our random surfer visits page i .
- The higher the probability, the more important the page.

An alternative view on PageRank — Random Surfer



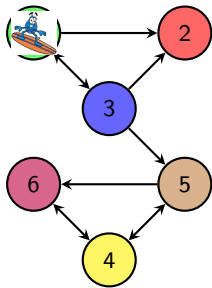
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- The user keeps surfing forever.
- The **probability** p_i that the user visits page i is then given by

$$p_i = \sum_{j \in B_i} \frac{p_j}{N_j}$$

So in essence:

- The equation $\mathbf{x} = H\mathbf{x}$ then asks, for each page, the probability x_i that our random surfer visits page i .
- The higher the probability, the more important the page.

An alternative view on PageRank — Random Surfer



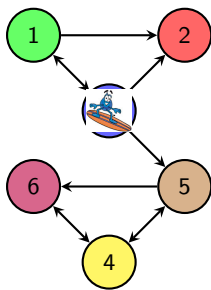
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- The user keeps surfing forever.
- The **probability** p_i that the user visits page i is then given by

$$p_i = \sum_{j \in B_i} \frac{p_j}{N_j}$$

So in essence:

- The equation $\mathbf{x} = H\mathbf{x}$ then asks, for each page, the probability x_i that our random surfer visits page i .
- The higher the probability, the more important the page.

An alternative view on PageRank — Random Surfer



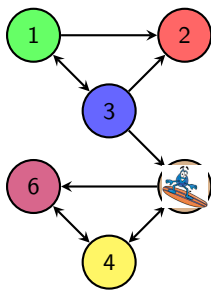
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- The user keeps surfing forever.
- The **probability** p_i that the user visits page i is then given by

$$p_i = \sum_{j \in B_i} \frac{p_j}{N_j}$$

So in essence:

- The equation $\mathbf{x} = H\mathbf{x}$ then asks, for each page, the probability x_i that our random surfer visits page i .
- The higher the probability, the more important the page.

An alternative view on PageRank — Random Surfer



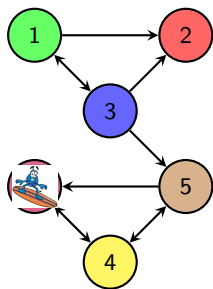
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- The user keeps surfing forever.
- The **probability** p_i that the user visits page i is then given by

$$p_i = \sum_{j \in B_i} \frac{p_j}{N_j}$$

So in essence:

- The equation $\mathbf{x} = H\mathbf{x}$ then asks, for each page, the probability x_i that our random surfer visits page i .
- The higher the probability, the more important the page.

An alternative view on PageRank — Random Surfer



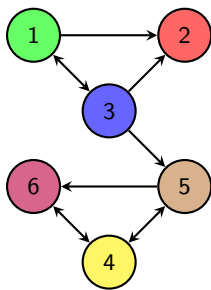
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- The user keeps surfing forever.
- The **probability** p_i that the user visits page i is then given by

$$p_i = \sum_{j \in B_i} \frac{p_j}{N_j}$$

So in essence:

- The equation $\mathbf{x} = H\mathbf{x}$ then asks, for each page, the probability x_i that our random surfer visits page i .
- The higher the probability, the more important the page.

An alternative view on PageRank — Random Surfer



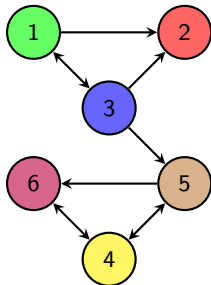
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- The user keeps surfing forever.
- The **probability** p_i that the user visits page i is then given by

$$p_i = \sum_{j \in B_i} \frac{p_j}{N_j}$$

So in essence:

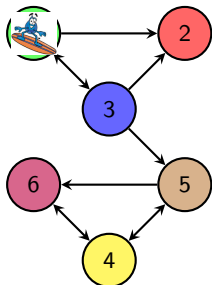
- The equation $\mathbf{x} = H\mathbf{x}$ then asks, for each page, the probability x_i that our random surfer visits page i .
- The higher the probability, the more important the page.

So why is there no solution?



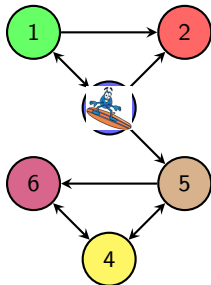
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- But the user gets stuck at “dangling” nodes

So why is there no solution?



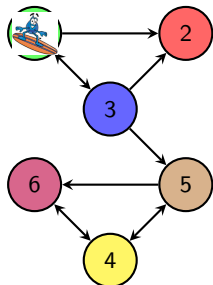
- A user starts surfing by entering a valid URL uniformly at random.
- At each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- But the user gets stuck at “dangling” nodes

So why is there no solution?



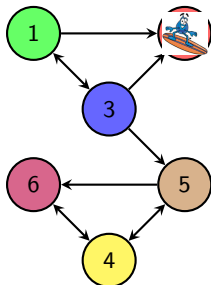
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- But the user gets stuck at “dangling” nodes

So why is there no solution?



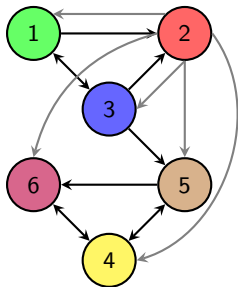
- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- But the user gets stuck at “dangling” nodes

So why is there no solution?



- A user starts surfing by entering a valid URL uniformly at random.
- At a each page, the user moves to a neighboring page by clicking a link on the page uniformly at random.
- **But the user gets stuck at “dangling” nodes**

PageRank version 2: correct for dangling nodes



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & \frac{1}{6} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{6} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{6} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{6} & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{6} & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}}_{\text{corrected hyperlink matrix } S} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

Definition

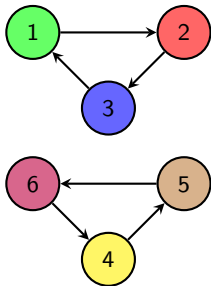
A matrix is **stochastic** if it contains only positive coefficients and all of its columns add to 1

Theorem

Every stochastic square matrix has 1 as eigenvalue

So, the equation $x = S \cdot x$ always has a solution.

Is there only one solution?



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

S has multiple eigenvectors with eigenvalue 1, for instance :

$$\begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \\ \frac{1}{6} \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{p} \\ \frac{1}{p} \\ \frac{1}{p} \\ \frac{1}{q} \\ \frac{1}{q} \\ \frac{1}{q} \end{bmatrix}$$

PageRank - final version



Modify the random surfer model. At each site the surfer:

- visits a random neighbouring site with probability α ;
- or “teleports” to an arbitrary site with probability $1 - \alpha$.

Google takes $\alpha = 85\%$

- **Probability that surfer visits site i :** $x_i = \alpha \sum_{j \in B_i} \frac{x_j}{N_j} + (1 - \alpha) \times \frac{1}{n}$
- In matrix notation:

$$\mathbf{x} = \underbrace{\left(\alpha S + \frac{1 - \alpha}{n} E_{n \times n} \right)}_{\text{Google matrix } G} \cdot \mathbf{x}$$

Theorem

The Google matrix G has a **single** eigenvector with eigenvalue 1. Moreover, this eigenvector is **stochastic**.

Computing the PageRank vector

Theorem

Let \mathbf{v}^0 be an arbitrary column vector that adds to 1. Let $\mathbf{v}^k := \mathbf{G} \cdot \mathbf{v}^{k-1}$. Then the pagerank vector equals $\lim_{k \rightarrow \infty} \mathbf{v}^k$.

Hence, to approximate the pagerank vector with tolerance ε :

- Let n be the total number of web pages
- Initialize \mathbf{v} to $(\frac{1}{n}, \dots, \frac{1}{n})^T$
- Initialize \mathbf{v}' to $(1, \dots, 1)^T$
- Repeat until $\|\mathbf{v}' - \mathbf{v}\| < \varepsilon$:
 - set $\mathbf{v}' := \mathbf{v}$
 - set $\mathbf{v} := \mathbf{G} \cdot \mathbf{v}$

According to Google, this algorithm converges to the PageRank vector within a reasonable tolerance for web graphs of 322 million links after roughly 52 iterations.

Computing the PageRank vector (2)

- Each multiplication $\mathbf{G} \cdot \mathbf{v}$ takes $O(n^2)$ time.
- However, n is HUGE: in 2004 Google reported that it had indexed 8×10^9 web pages.
- Using the fact that on average every page has only 10 outgoing links, one can use results from linear algebra to perform this multiplication in $O(n)$ time.
- Still, Google is reported to calculate the PageRank vector only once a month!