PageRank
aneb lineární algebra za miliony

Tomáš Svoboda, svobodat@fel.cvut.cz
první verze přednášky Ondřej Drbohlav, drbohlav@fel.cvut.cz
http://cw.felk.cvut.cz/doku.php/courses/b4b33rph/start

13. prosince 2023

Notes

slido.com, #1209972

Notes

Major Events of 1997

- Britain cedes Hong Kong (a British colony since 1841) back to China.
- Princess Diana dies in road accident in Paris, France.
- NASA's Mars Pathfinder probe lands on Mars.
- An unmanned Progress cargo-ship crashes into the Mir space station.
- Tiger Woods becomes the youngest golfer to win the Masters Tournament.
- Franklin Delano Roosevelt Memorial dedicated in Washington D.C.

(Podle `http://www.fun4birthdays.com`, odkaz č. 1 z prohledávače google.com) na dotaz 'Major events 1997'

Notes

Vyzkoušejte si své hledání. Co používáte jako hlavní vyhledávač, resp. navigátor na internetu? Napište na slido.

1997 – technika

Typické PC

- 14-ti palcový monitor
- Procesor: Pentium, 1 jádro, 233MHz
- 32MB RAM
- 4.3GB harddisk
- floppy disk

TS home PC: Intel 386SX, 33MHz (turbo!), 2MB RAM, 105 MB hard, černobílý monitor 14″.

Notes

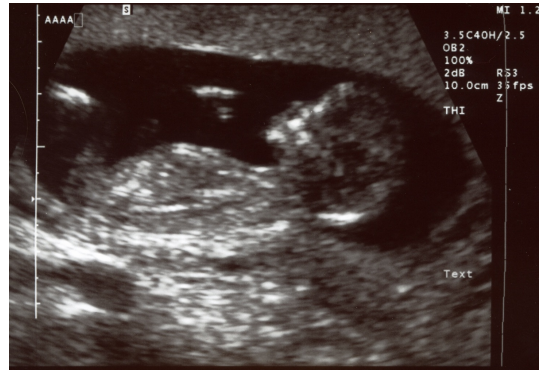
Už jsme si odvykli počítat v MB, není-li pravda?

1997 – budoucí studenti OI, jak šel RPH čas

Notes

Program OI je tu s námi již nějaký čas, takto to bylo na počátku. Ale dnes již nikdo z vás nepamatuje.

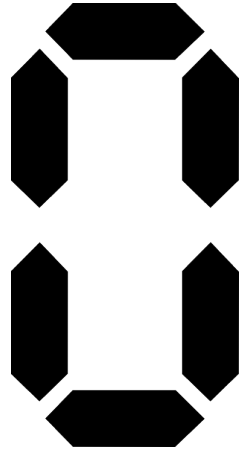
1997 – budoucí studenti OI, jak šel RPH čas



Notes

Program OI je tu s námi již nějaký čas, takto to bylo na počátku. Ale dnes již nikdo z vás nepamatuje.

1997 – budoucí studenti OI, jak šel RPH čas



Notes

Program OI je tu s námi již nějaký čas, takto to bylo na počátku. Ale dnes již nikdo z vás nepamatuje.

1997 – Internet (skoro tak, jak ho známe)

- Věk:

- první WWW prohlížeč Mosaic (1993)
- Netscape (1994)

Vyhledávací stroje:

- Altavista
- viz pár příkladu ze stroje času `http://web.archive.org/`

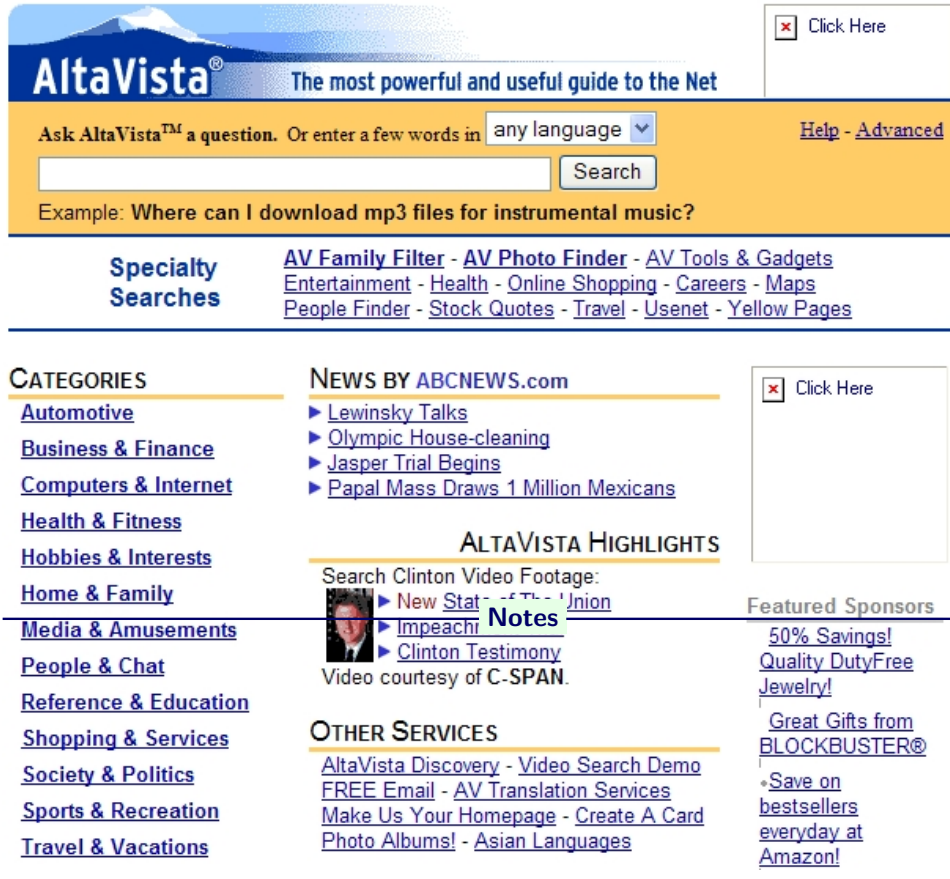
Notes

Dávná historie internetu

- http://web.archive.org/web/19961230033620/http://www.uzdroje.cz/
- http://web.archive.org/web/19961114213839/http://www.seznam.cz/
- http://web.archive.org/web/19970502010243/http://www.feld.cvut.cz/
- http://web.archive.org/web/19970501133354/http://cmp.felk.cvut.cz/
- http://web.archive.org/web/19981111184551/http://google.com/
- http://web.archive.org/web/19981111183552/http://google.stanford.edu/
- http://web.archive.org/web/19990125084553/http://alpha.google.com/
- http://web.archive.org/web/19961020022754/http://www9.yahoo.com/
- http://cmp.felk.cvut.cz/~svoboda/Lodenice/

Notes

Příklad: Altavista



Notes

Příklad: Hledání

Hledej: Apple

1. . . . byla jsem za Petrou a ta mě požádala o recept na skvělý babiččin koláč. Tady je: jedno **jablko**, dvě vejce, . . .
2. . . .

⋮

105. **Apple**. Chystáme se na výrobu iPodů. Až vyrostete, kupujte to!

⋮

217. **Jablko** patří mezi ovoce. Příbuzným druhem je hruška . . .

⋮

Notes

1997 - vlastnosti hledání

- velice často *nerelevantní* výsledky
- nic se nedá najít a nebo to zabere spoustu času
- v dokumentu se slova sice vyskytují, ale dokument sám je na poměrně obskurních stránkách
- $\longrightarrow$ bookmarks/záložky

Příčiny? Jak se hledalo?

Notes

Odskok: jak se to dělá s vědeckými, inženýrskými články, ale vlastně i například s knihami či jiným tvůrčím dílem obecně:

- Peer review.
- Citační ohlas článku, resp. kolik lidí s knihu koupí.
- Zkouška časem. Je obsah stále relevantní?

1997 - vlastnosti hledání

- velice často *nerelevantní* výsledky
- nic se nedá najít a nebo to zabere spoustu času
- v dokumentu se slova sice vyskytují, ale dokument sám je na poměrně obskurních stránkách
- $\longrightarrow$ bookmarks/záložky

Příčiny? Jak se hledalo?

- za všemi vyhledávacími stroji jsou stovky lidí, kteří procházejí rodící se internet. Odhadují důležitost stránek
- problém je v tom, že Internet rychle přerostl možnosti ručního procházení: jednak počtem stránek, jednak rychlostí obměny obsahu.

Notes

Odskok: jak se to dělá s vědeckými, inženýrskými články, ale vlastně i například s knihami či jiným tvůrčím dílem obecně:

- Peer review.
- Citační ohlas článku, resp. kolik lidí s knihu koupí.
- Zkouška časem. Je obsah stále relevantní?

Obecně: Jak hodnotit tvůrčí díla?

- Peer review, recenze, ...
  - Záleží od koho recenze nebo peer review je?
  - Kolik recenzí, shodnou se na něčem?
  - Je review/recenze „blind“?

Notes

- Slovem „neškáluje“ rozumíme, že nová díla, resp. weby přibývají rychleji, než je lze ručně organizovat či hodnotit.
- Ostatně podobný problém nastává i u vědeckých článků. Příliš mnoho autorů na příliš málo kvalitích oponentů.

Obecně: Jak hodnotit tvůrčí díla?

- Peer review, recenze, ...
  - Záleží od koho recenze nebo peer review je?
  - Kolik recenzí, shodnou se na něčem?
  - Je review/recenze „blind“?
- Citační ohlas článku, resp. kolik lidí s knihu koupí.
  - Cena, dostupnost, ...
  - Reklama!
  - Osobnost autor*ky.

Notes

- Slovem „neškáluje“ rozumíme, že nová díla, resp. weby přibývají rychleji, než je lze ručně organizovat či hodnotit.
- Ostatně podobný problém nastává i u vědeckých článků. Příliš mnoho autorů na příliš málo kvalitích oponentů.

Obecně: Jak hodnotit tvůrčí díla?

- Peer review, recenze, ...
  - Záleží od koho recenze nebo peer review je?
  - Kolik recenzí, shodnou se na něčem?
  - Je review/recenze „blind“?
- Citační ohlas článku, resp. kolik lidí s knihu koupí.
  - Cena, dostupnost, ...
  - Reklama!
  - Osobnost autor*ky.
- Zkouška časem. Je obsah stále relevantní?
  - Stále se kupuje? Re-edice, ...
  - Inspiruje jinou oblast? Film, divadlo, ...
  - Citační ohlas.

Notes

- Slovem „neškáluje“ rozumíme, že nová díla, resp. weby přibývají rychleji, než je lze ručně organizovat či hodnotit.
- Ostatně podobný problém nastává i u vědeckých článků. Příliš mnoho autorů na příliš málo kvalitích oponentů.

Obecně: Jak hodnotit tvůrčí díla?

- Peer review, recenze, ...
  - Záleží od koho recenze nebo peer review je?
  - Kolik recenzí, shodnou se na něčem?
  - Je review/recenze „blind“?
- Citační ohlas článku, resp. kolik lidí s knihu koupí.
  - Cena, dostupnost, ...
  - Reklama!
  - Osobnost autor*ky.
- Zkouška časem. Je obsah stále relevantní?
  - Stále se kupuje? Re-edice, ...
  - Inspiruje jinou oblast? Film, divadlo, ...
  - Citační ohlas.

Ruční hodnocení stále nejlepší, ale *neškáluje*!

Notes

- Slovem „neškáluje“ rozumíme, že nová díla, resp. weby přibývají rychleji, než je lze ručně organizovat či hodnotit.
- Ostatně podobný problém nastává i u vědeckých článků. Příliš mnoho autorů na příliš málo kvalitích oponentů.

Co musí vyhledávací stroj umět?

Notes

Co musí vyhledávací stroj umět?

- **pracovat plně automaticky**
- procházet Internet
- indexovat stránky (uchovávat jejich obsah ve formě vhodné k rychlému vyhledání při zadání klíčového slova)
- hodnotit důležitost stránek

Notes

Co musí vyhledávací stroj umět?

- pracovat plně automaticky
- **procházet Internet**
- indexovat stránky (uchovávat jejich obsah ve formě vhodné k rychlému vyhledání při zadání klíčového slova)
- hodnotit důležitost stránek

Jak procházet web a nezacyklit se?

Notes

Co musí vyhledávací stroj umět?

- pracovat plně automaticky
- procházet Internet
- **indexovat stránky (uchovávat jejich obsah ve formě vhodné k rychlému vyhledání při zadání klíčového slova)**
- hodnotit důležitost stránek

Jaké struktury dat zvolit pro indexování? Jak provádět efektivní vyhledávání dokumentů s výskytem klíčového slova?

Notes

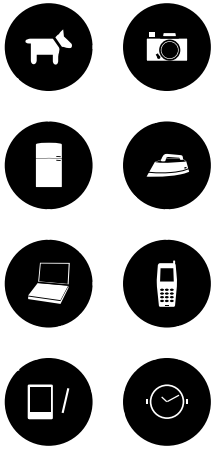
Co musí vyhledávací stroj umět?

- pracovat plně automaticky
- procházet Internet
- indexovat stránky (uchovávat jejich obsah ve formě vhodné k rychlému vyhledání při zadání klíčového slova)
- **hodnotit důležitost stránek**

- Je přesně to, co vyhledávače v roce 1997 neuměly.
- Přímo souvisí s pozorovaným problémem s nerelevantností výsledků hledání.
- Očekávané zlepšení výsledků vyhledávání je velké. Většina textu na internetu se obsahuje cca 10.000 slov; dnešní velikost internetu – miliardy stránek $\Rightarrow$ seřazení stránek podle důležitosti je opravdu potřebné!

Notes

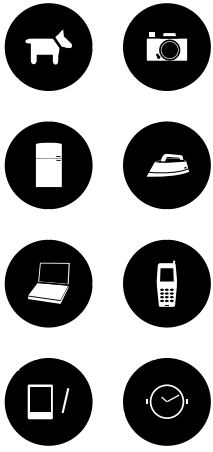
Jak automaticky zjistit důležitost stránek?



Notes

Otázka na slido.

Jak automaticky zjistit důležitost stránek?

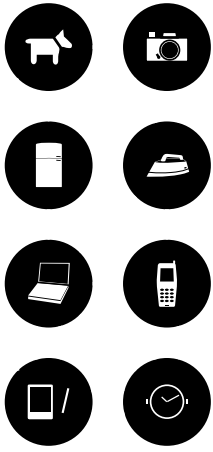


- velikost stránky?
- použití spisovného jazyka?
- kritéria založená na rozpoznávání smyslu textu?
- ...

Notes

Otázka na slido.

Jak automaticky zjistit důležitost stránek?



- velikost stránky?
- použití spisovného jazyka?
- kritéria založená na rozpoznávání smyslu textu?
- ...

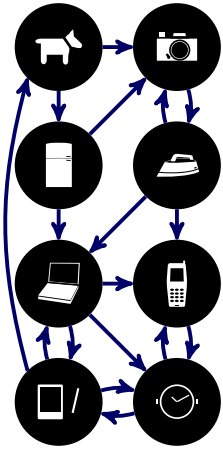
Možné problémy:

- částečně by to mohlo fungovat, ale ...
- jednoduchá kritéria je "snadné" splnit: *Velikost*: dosáhnout je to podstatně jednodušší než např. ve světě firem [Update2023: pomněte ChatGPT a spol.]
- *smysl textu*: extrémně těžké, neumí se to
- leccos je v zásadě snadné zfalšovat

Notes

Otázka na slido.

Jak automaticky zjistit důležitost stránek?



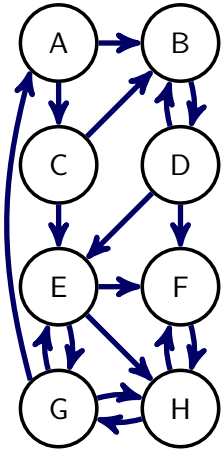
- web nejsou jen stránky, ale i odkazy mezi nimi
- mohl bych je použít pro spočítání důležitosti stránek?

Úvahy:

- pokud O doporučuje (odkazuje) mnoho lidí, O je věrohodná
- velká výhoda bude založit skóre stránky na odkazech **na** tuto stránku, tj. **zpětných** odkazech, protože je to něco, co vlastník stránky nemá v moci.

Notes

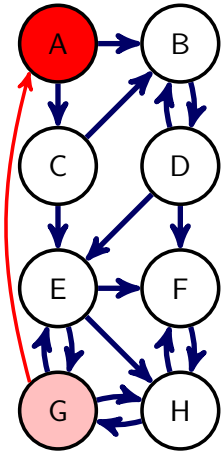
Nápad 1: počítání zpětných odkazů



- začněme co nejjednoduššeji: co spočítat počet zpětných linků?

Notes

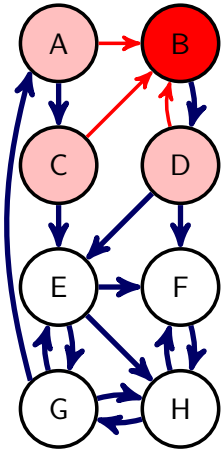
Nápad 1: počítání zpětných odkazů



- začněme co nejjednoduššeji: co spočítat počet zpětných linků?
- $W_A = 1$

Notes

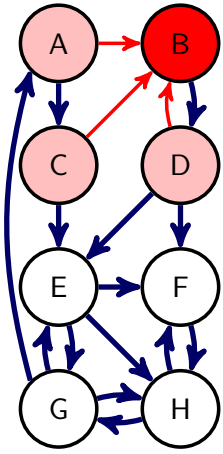
Nápad 1: počítání zpětných odkazů



- začněme co nejjednoduššeji: co spočítat počet zpětných linků?
- $W_A = 1$
- $W_B = 3$

Notes

Nápad 1: počítání zpětných odkazů

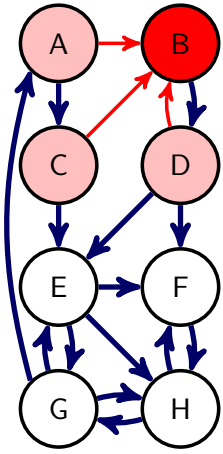


- začněme co nejjednoduššeji: co spočítat počet zpětných linků?
- $W_A = 1$
- $W_B = 3$

  ...
- $W = [1, 3, 1, 1, 3, 3, 2, 3]$

Notes

Nápad 1: počítání zpětných odkazů

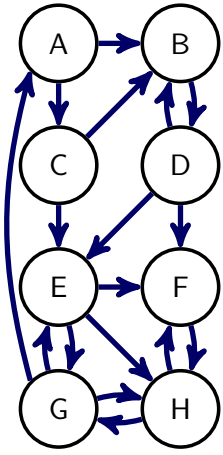


- začněme co nejjednoduššeji: co spočítat počet zpětných linků?
- $W_A = 1$
- $W_B = 3$

  ...
- $W = [1, 3, 1, 1, 3, 3, 2, 3]$
- jako první přiblížení je to fajn, ale je samotný počet zpětných odkazů jednak nepopisuje v podstatě vůbec strukturu sítě, jednak je snadno manipulovatelný (návod: pan Zlobivý si koupí 10 domén a z každé z nich odkáže svou hlavní doménu)

Notes

Nápad 1.1: Váhování zpětných odkazů I



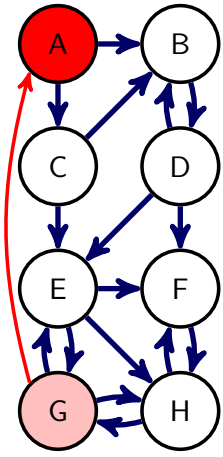
Úvaha:

- Není jedno, kdo na mě dává odkaz. Odkaz od důležité stránky má větší váhu než víceméně náhodný odkaz.
- Zkusme tedy sčítat ne *počet* odkazů, ale jejich *váhy*:

Notes

Podobná úvaha jako při ručním hodnocení. Osoba recenzenta je důležitá.

Nápad 1.1: Váhování zpětných odkazů I



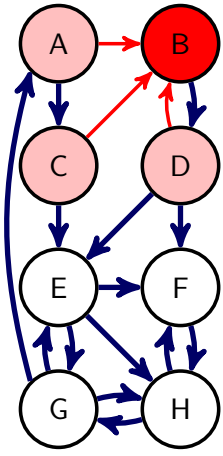
Úvaha:

- Není jedno, kdo na mě dává odkaz. Odkaz od důležité stránky má větší váhu než víceméně náhodný odkaz.
- Zkusme tedy sčítat ne *počet* odkazů, ale jejich *váhy*:
- $W_A = W_G$

Notes

Podobná úvaha jako při ručním hodnocení. Osoba recenzenta je důležitá.

Nápad 1.1: Váhování zpětných odkazů I



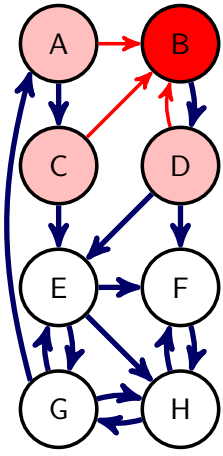
Úvaha:

- Není jedno, kdo na mě dává odkaz. Odkaz od důležité stránky má větší váhu než víceméně náhodný odkaz.
- Zkusme tedy sčítat ne *počet* odkazů, ale jejich *váhy*:
- $W_A = W_G$
- $W_B = W_A + W_C + W_D$

Notes

Podobná úvaha jako při ručním hodnocení. Osoba recenzenta je důležitá.

Nápad 1.1: Váhování zpětných odkazů I



Úvaha:

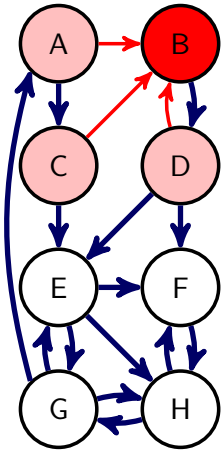
- Není jedno, kdo na mě dává odkaz. Odkaz od důležité stránky má větší váhu než víceméně náhodný odkaz.
- Zkusme tedy sčítat ne *počet* odkazů, ale jejich *váhy*:
- $W_A = W_G$
- $W_B = W_A + W_C + W_D$
- $W_D = W_B$

  ⋮

Notes

Podobná úvaha jako při ručním hodnocení. Osoba recenzenta je důležitá.

Nápad 1.1: Váhování zpětných odkazů I



Úvaha:

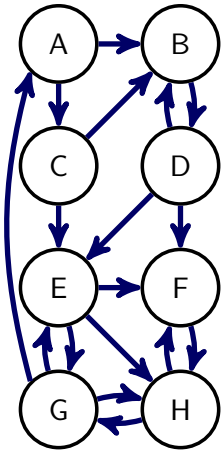
- Není jedno, kdo na mě dává odkaz. Odkaz od důležité stránky má větší váhu než víceméně náhodný odkaz.
- Zkusme tedy sčítat ne *počet* odkazů, ale jejich *váhy*:
- $W_A = W_G$
- $W_B = W_A + W_C + W_D$
- $W_D = W_B$

  ⋮
- To vypadá rozumně, navíc  má trochu ztíženou práci: musí myslet na to, že svým „pomocným“ doménám musí zajistit kredibilitu.

Notes

Podobná úvaha jako při ručním hodnocení. Osoba recenzenta je důležitá.

Nápad 1.1: Váhování zpětných odkazů II

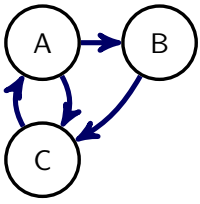


Úvaha:

- máme ale problém, uvažme jednoduchou síť

Notes

Nápad 1.1: Váhování zpětných odkazů II

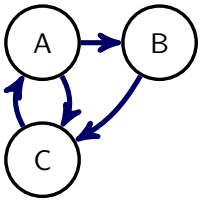


Úvaha:

- máme ale problém, uvažme jednoduchou síť

Notes

Nápad 1.1: Váhování zpětných odkazů II



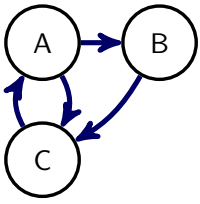
Úvaha:

- máme ale problém, uvažme jednoduchou síť

$$W_A = W_C$$
$$W_B = W_A$$
$$W_C = W_A + W_B$$

Notes

Nápad 1.1: Váhování zpětných odkazů II



Úvaha:

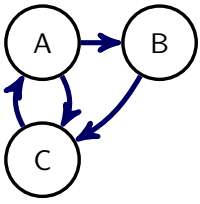
- máme ale problém, uvažme jednoduchou síť

$$W_A = W_C$$
$$W_B = W_A$$
$$W_C = W_A + W_B$$

- to není možné splnit

Notes

Nápad 1.1: Váhování zpětných odkazů II



Úvaha:

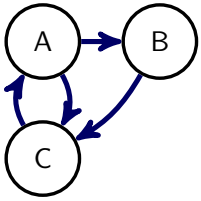
- máme ale problém, uvažme jednoduchou síť

$$W_A = W_C$$
$$W_B = W_A$$
$$W_C = W_A + W_B$$

- to není možné splnit
- možná není správné, že stránky mohou hlasovat pro libovolný počet ostatních stránek
- co kdybychom tedy každému dali jen jeden hlas, který může spravedlivě dělit mezi všechny stránky, které doporučuje?

Notes

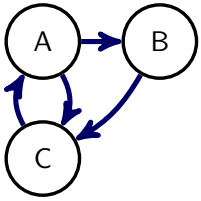
Nápad 1.2: Poměrné hlasování



- označme n_A počet odkazů vedoucích z webu A
- $n_A = 2$, $n_B = 1$, $n_C = 1$

Notes

Nápad 1.2: Poměrné hlasování

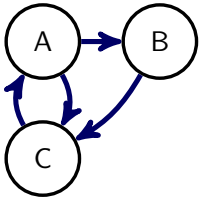


- označme n_A počet odkazů vedoucích z webu A
- $n_A = 2$, $n_B = 1$, $n_C = 1$
- s použitím poměrného hlasování tedy bude

$$\begin{aligned} W_A &= \frac{W_C}{n_C} = W_C \\ W_B &= \frac{W_A}{n_A} = \frac{1}{2} W_A \\ W_C &= \frac{W_A}{n_A} + \frac{W_B}{n_B} = \frac{1}{2} W_A + W_B \end{aligned}$$

Notes

Nápad 1.2: Poměrné hlasování



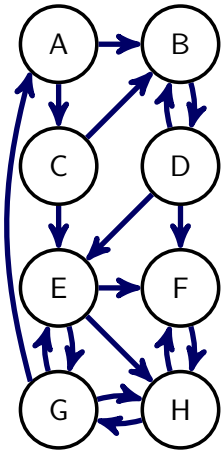
- označme n_A počet odkazů vedoucích z webu A
- $n_A = 2$, $n_B = 1$, $n_C = 1$
- s použitím poměrného hlasování tedy bude

$$\begin{aligned} W_A &= \frac{W_C}{n_C} = W_C \\ W_B &= \frac{W_A}{n_A} = \frac{1}{2} W_A \\ W_C &= \frac{W_A}{n_A} + \frac{W_B}{n_B} = \frac{1}{2} W_A + W_B \end{aligned}$$

- to je konzistentní!

Notes

Nápad 1.2: Poměrné hlasování, jak řešit?



- Zkusme hodnotit důležitost webů takto:

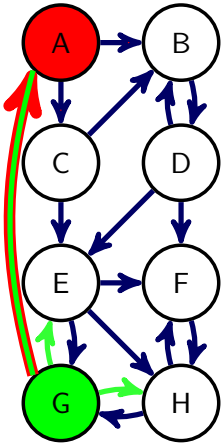
$$W_k = \sum_{b \in \{\to k\}} \frac{W_b}{n_b},$$

kde $\{\to k\}$ je množina webů, které odkazují na web k.

Notes

Neznámé máme v soustavě rovnic na obou stranách! Soustava je konzistetní, ale stále nevíme, jak vyřešit. Odložme ten bolehlav na chvilku.

Nápad 1.2: Poměrné hlasování, jak řešit?



- Zkusme hodnotit důležitost webů takto:

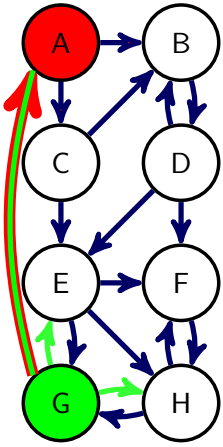
$$W_k = \sum_{b \in \{\to k\}} \frac{W_b}{n_b},$$

  kde $\{\to k\}$ je množina webů, které odkazují na web k.
- Příklad: $W_A = \frac{W_G}{3}$

Notes

Neznámé máme v soustavě rovnic na obou stranách! Soustava je konzistetní, ale stále nevíme, jak vyřešit. Odložme ten bolehlav na chvilku.

Nápad 1.2: Poměrné hlasování, jak řešit?



- Zkusme hodnotit důležitost webů takto:

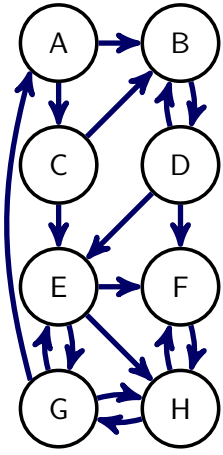
$$W_k = \sum_{b \in \{\to k\}} \frac{W_b}{n_b},$$

  kde $\{\to k\}$ je množina webů, které odkazují na web k.
- Příklad: $W_A = \frac{W_G}{3}$
- Jak takovou soustavu rovnic vyřešit?
- Jde to vůbec? (pro obecný web)?
- Je řešení jen jedno, nebo víc?

Notes

Neznámé máme v soustavě rovnic na obou stranách! Soustava je konzistetní, ale stále nevíme, jak vyřešit. Odložme ten bolehlav na chvilku.

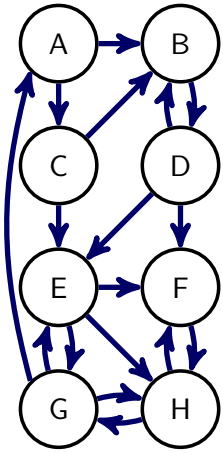
Nápad 2: Náhodná procházka



- co kdybychom nemuseli počítat skoro nic?
- máme strukturu webu k dispozici – jak bychom toho mohli využít
- nechme robota, aby procházel web následujícím způsobem:
  - začni na náhodné stránce
  - na této stránce je X odkazů. Náhodně se rozhodni pro jeden z nich
  - opakuj
  - počítej, kolikrát se dostaneš na jednotlivé stránky
- to celkem dává smysl, a vystihuje to konektivitu sítě

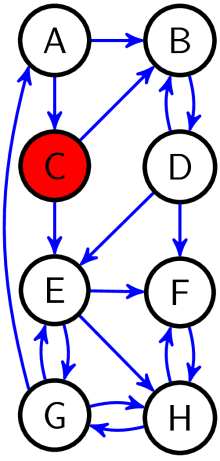
Notes

Nápad 2: Náhodná procházka



Notes

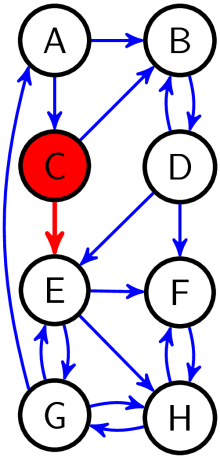
Nápad 2: Náhodná procházka



- začnu na C; 2 možná pokračování

Notes

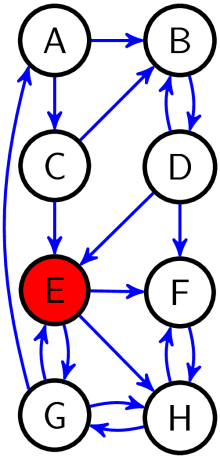
Nápad 2: Náhodná procházka



- začnu na C; 2 možná pokračování
- náhodně vyberu

Notes

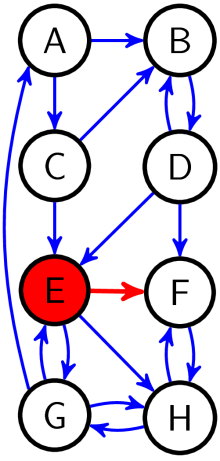
Nápad 2: Náhodná procházka



- začnu na C; 2 možná pokračování
- náhodně vyberu
- dostanu se do E; 3 cesty

Notes

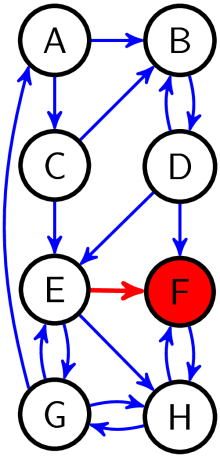
Nápad 2: Náhodná procházka



- začnu na C; 2 možná pokračování
- náhodně vyberu
- dostanu se do E; 3 cesty
- náhodně vyberu

Notes

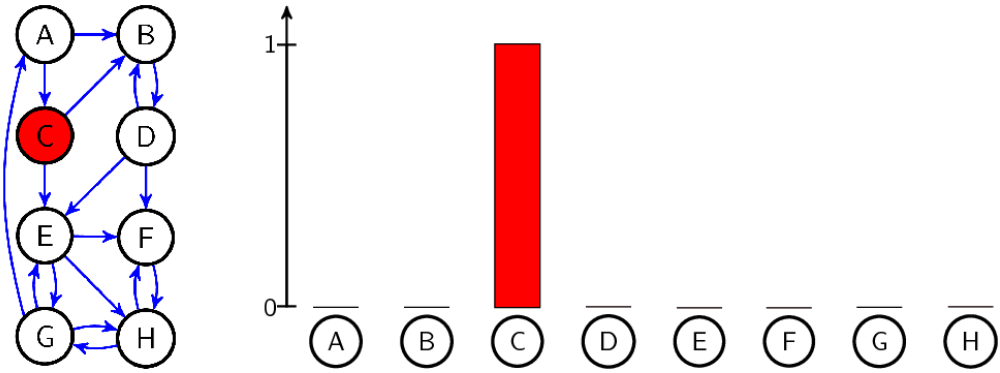
Nápad 2: Náhodná procházka



- začnu na C; 2 možná pokračování
- náhodně vyberu
- dostanu se do E; 3 cesty
- náhodně vyberu
- dostanu se do F
- . . . a tak dále

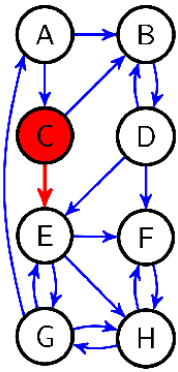
Notes

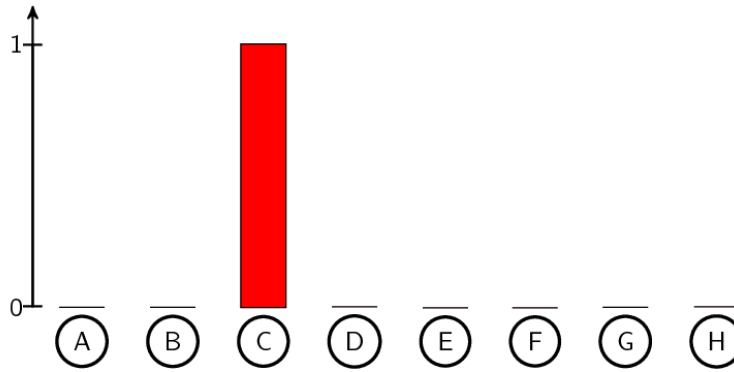
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)



Notes

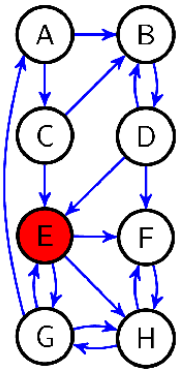
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

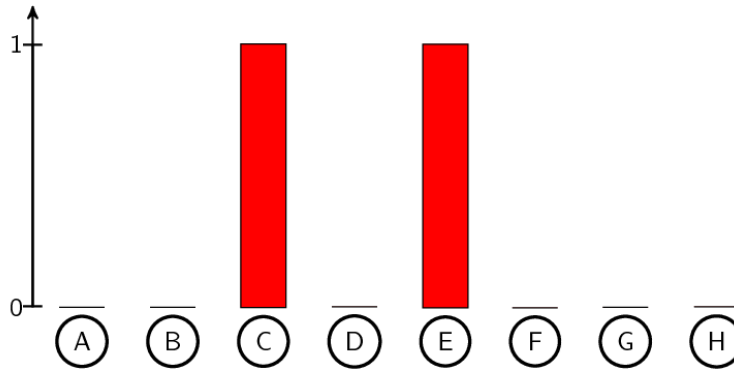




Notes

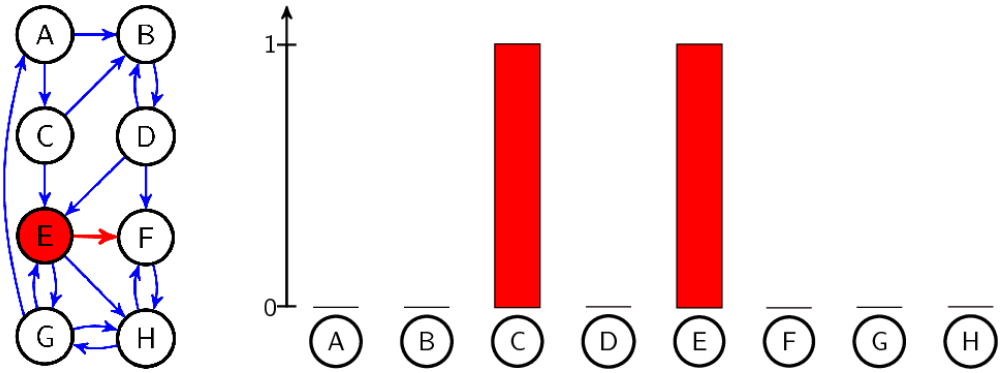
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)





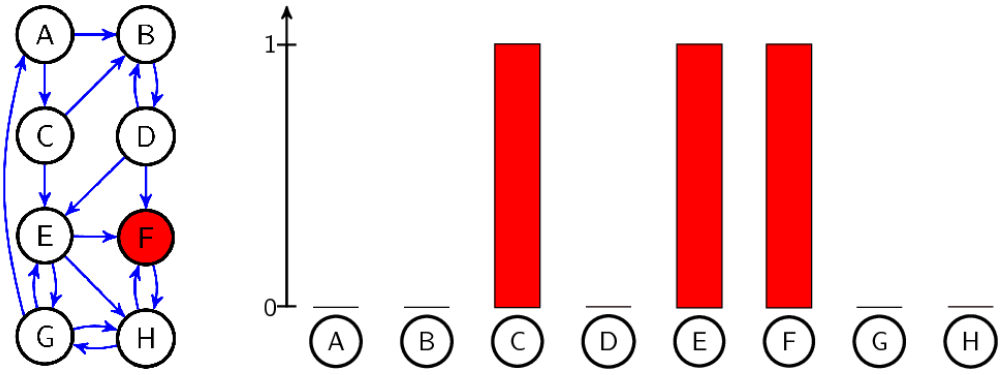
Notes

Nápad 2: Náhodná procházka, počítání průchodů (návštěv)



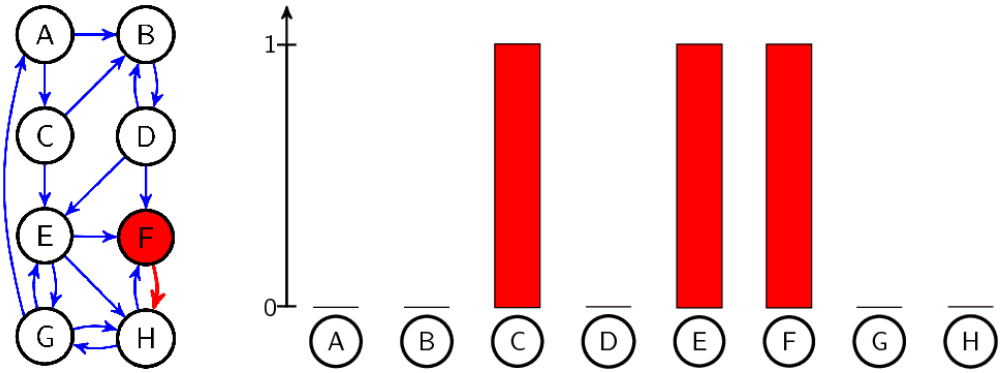
Notes

Nápad 2: Náhodná procházka, počítání průchodů (návštěv)



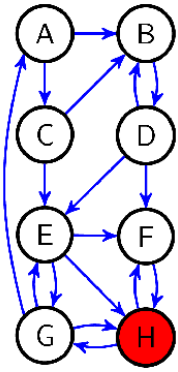
Notes

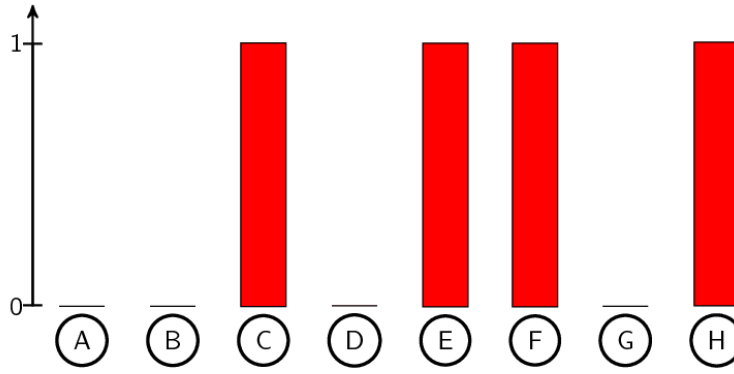
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)



Notes

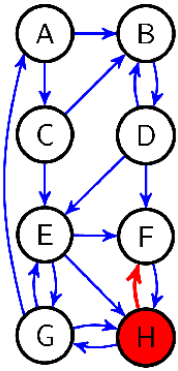
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

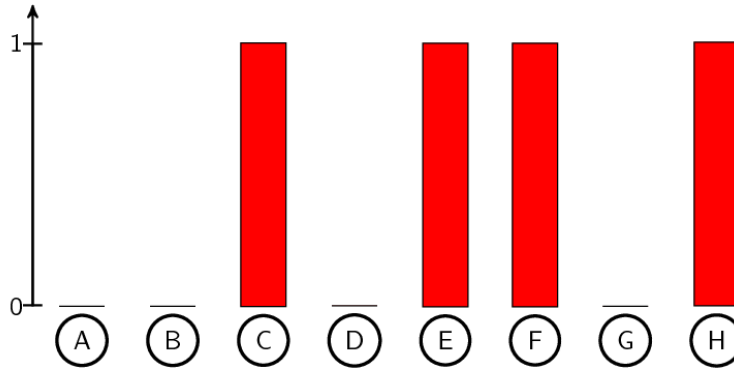




Notes

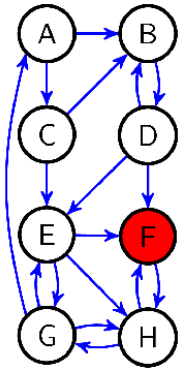
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

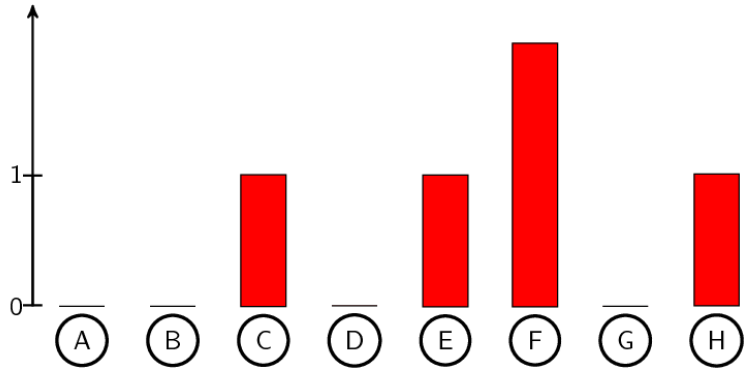




Notes

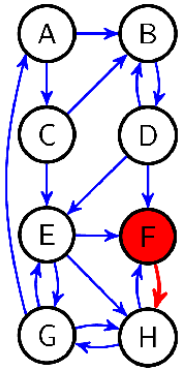
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

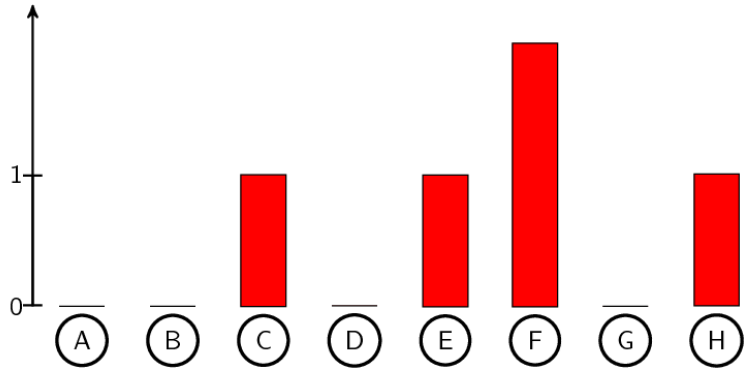




Notes

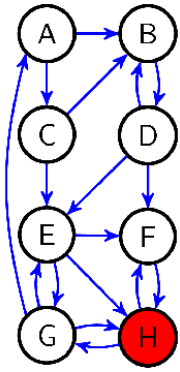
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

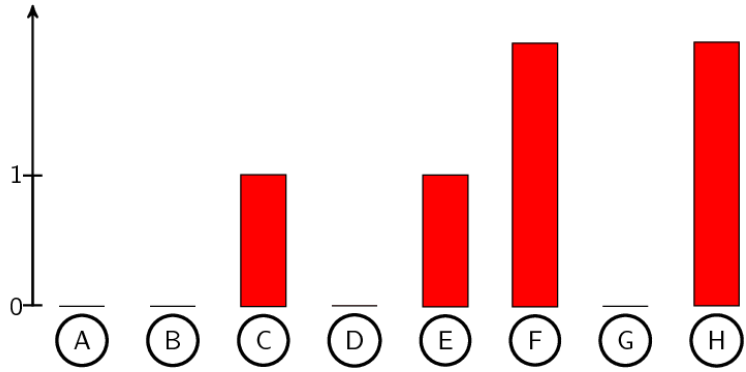




Notes

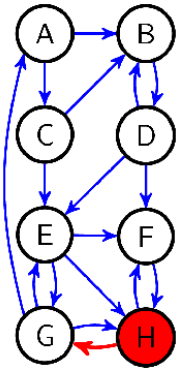
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

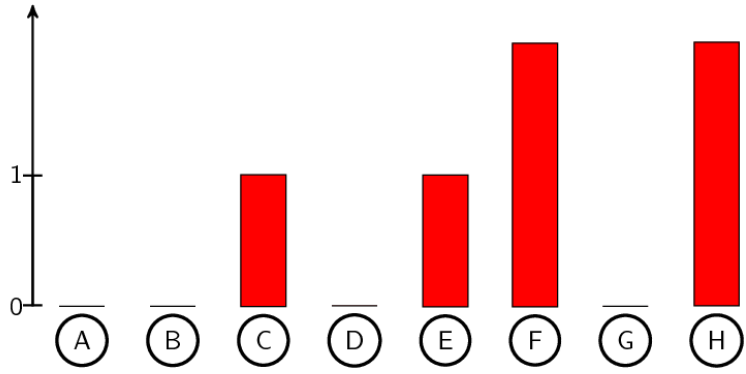




Notes

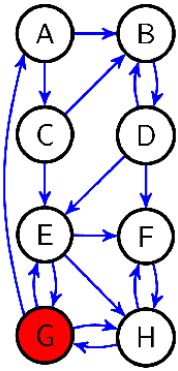
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

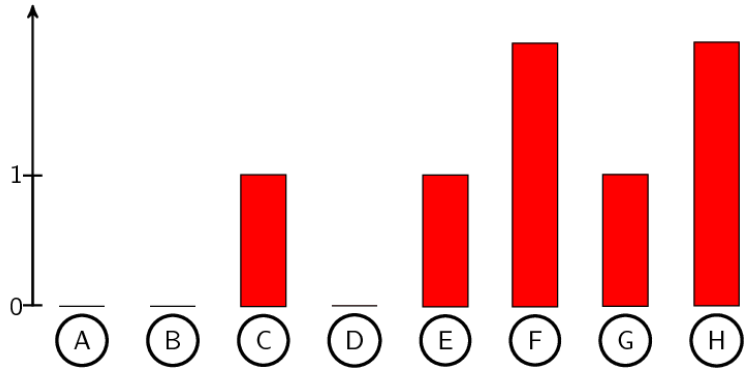




Notes

Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

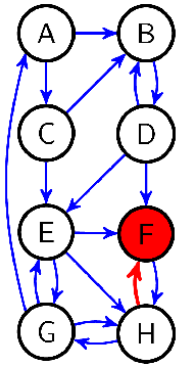


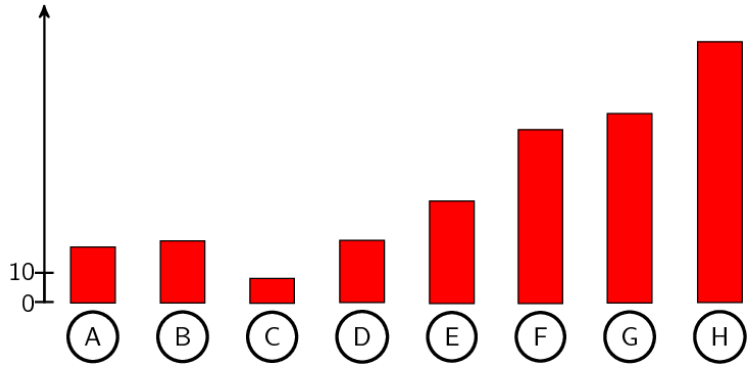


Notes

Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

300 iterací





$T = (18, 20, 8, 20, 33, 56, 61, 84)$

Notes

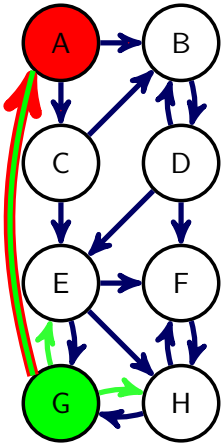
Nápad 2: Náhodná procházka, počítání průchodů (návštěv)

$$T = (18, 20, 8, 20, 33, 56, 61, 84)$$

- dobře, z náhodné procházky dostaneme vždy počty průchodů
- co ta čísla jsou, je jasné. Mohu nějak vysvětlit, jak jednotlivá T souvisejí?

Notes

Nápad 2: Náhodná procházka, počítání průchodů (návštěv)



$$T = (18, 20, 8, 20, 33, 56, 61, 84)$$

- dobře, z náhodné procházky dostaneme vždy počty průchodů
- co ta čísla jsou, je jasné. Mohu nějak vysvětlit, jak jednotlivá T souvisejí?
- ano: pokud jsem například strávil na A čas T_A, mohlo se to stát jen proto, že jsem tam odněkud přišel, v tomto případě z G. Z G se chodí do A v 1/3 případů.
- $T_A \approx \frac{T_G}{3}$
- hmmm, to je zajímavé!

Notes

Naše dva nápady: Ekvivalence!

Nápad 1: Poměrné hlasování

- odvozeno na základě velmi jednoduchých úvah
- máme podmínky na to, jak mají spolu váhy W_k jednotlivých webů souviset
- zatím nevíme, jak tyto podmínky splnit a tak najít řešení

Nápad 2: Náhodná procházka

- přirozená úvaha při absenci dalších znalostí
- dokážeme modelovat problém, získat časy strávené na jednotlivých stránkách T_k
- teprve po krátké analýze se ukázalo, že $T_k = W_k$ a tedy Nápad 2 řeší soustavu rovnic z Nápadu 1!

Notes

Systém $W_k = \sum A_{kb} W_b$

Chceme, aby platilo

$$W_k = \sum_{b \in \{\to k\}} \frac{W_b}{n_b}$$

Notes

Systém $W_k = \sum A_{kb} W_b$

Chceme, aby platilo

$$W_k = \sum_{b \in \{\to k\}} \frac{W_b}{n_b} = \sum_b A_{kb} W_b \,,$$

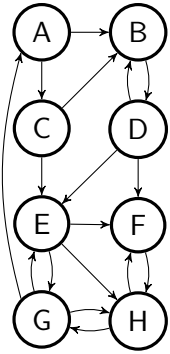
což je ekvivaletní maticovému zápisu

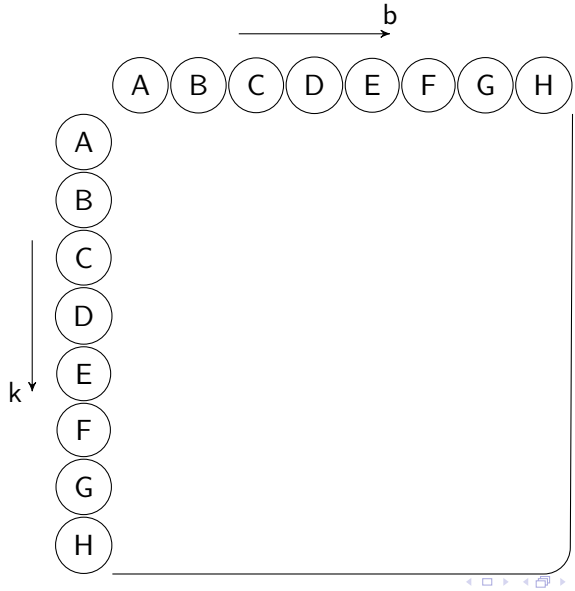
$$\mathbf{w} = \mathtt{A}\mathbf{w} \,.$$

- A_{kb} je nenulové tam, kde existuje odkaz $b \to k$.
- je rovno $1/$(počet odkazů z b)

Notes

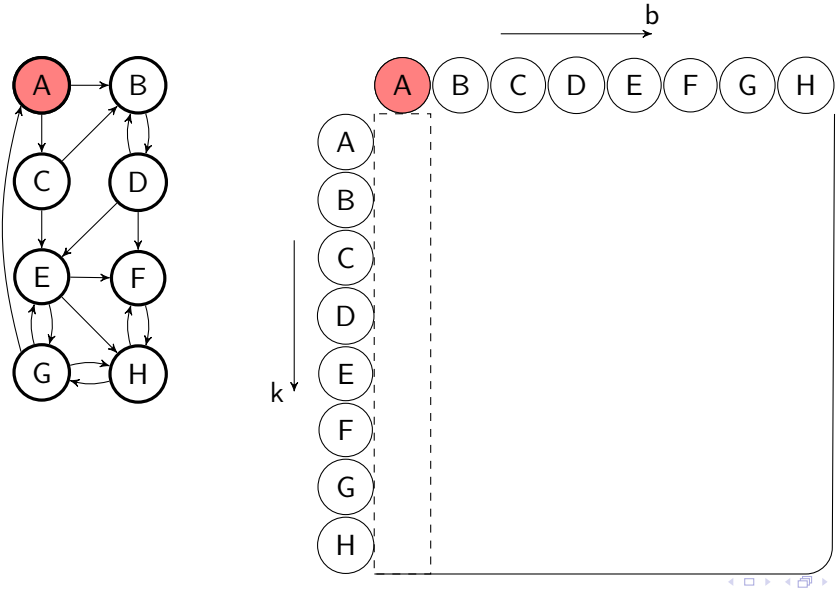
Konstrukce matice A, matice sousednosti (Adjacency)





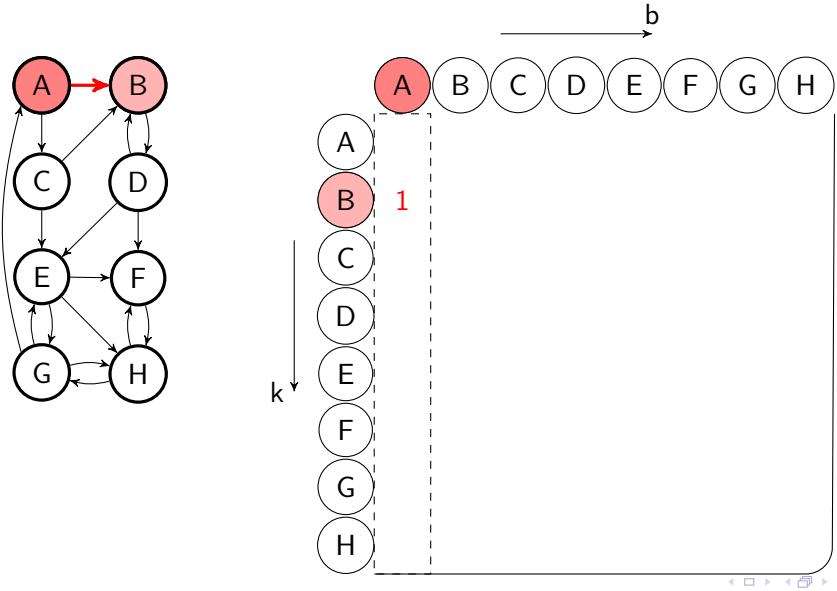
Notes

Konstrukce matice A, matice sousednosti (Adjacency)



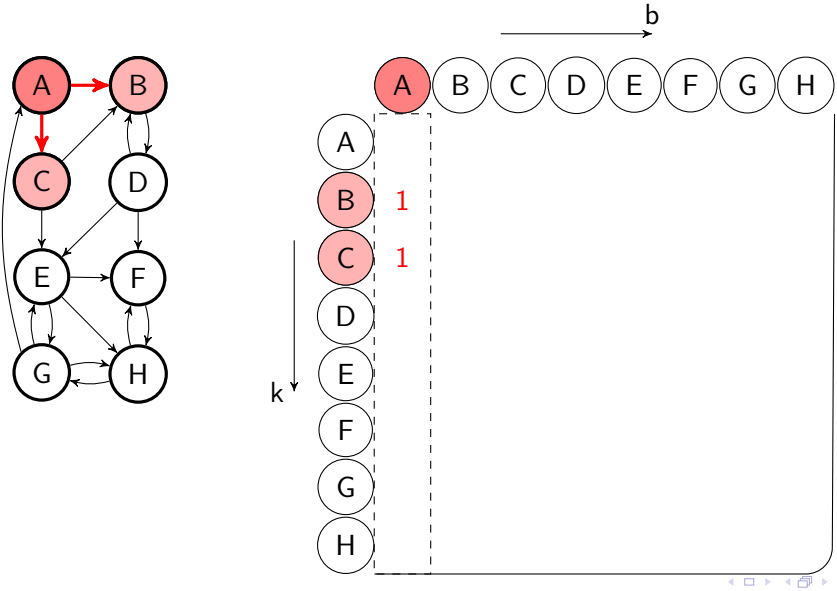
Notes

Konstrukce matice A, matice sousednosti (Adjacency)



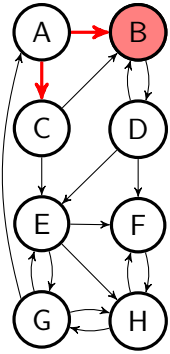
Notes

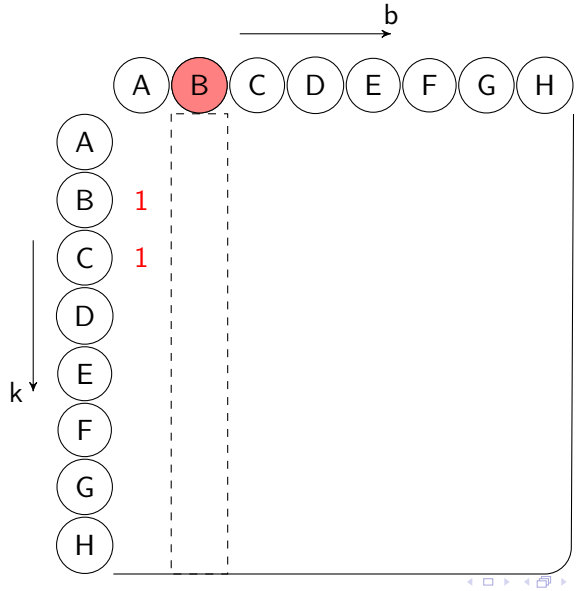
Konstrukce matice A, matice sousednosti (Adjacency)



Notes

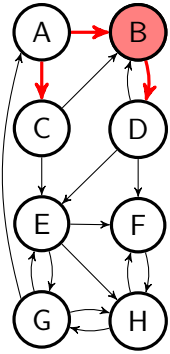
Konstrukce matice A, matice sousednosti (Adjacency)

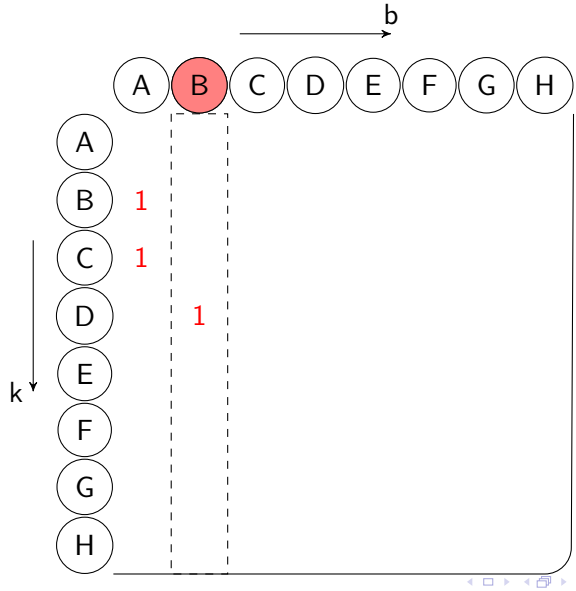




Notes

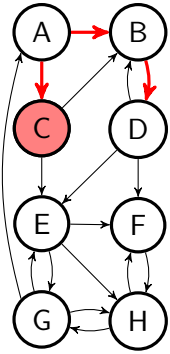
Konstrukce matice A, matice sousednosti (Adjacency)

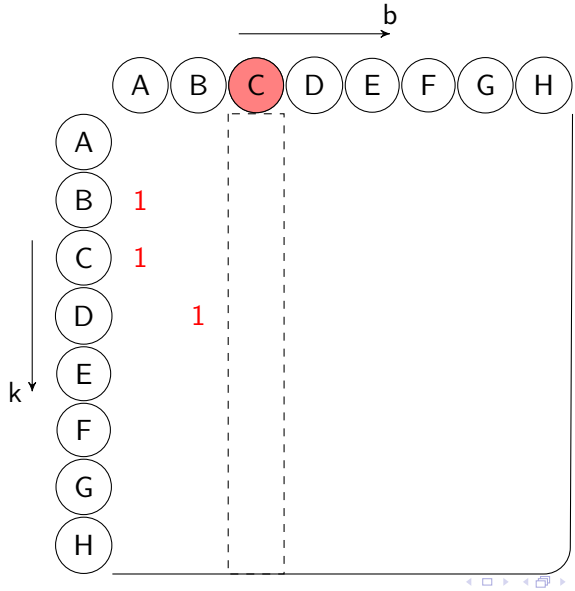




Notes

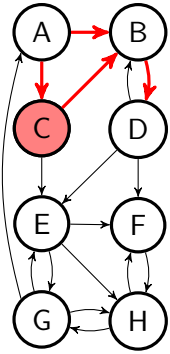
Konstrukce matice A, matice sousednosti (Adjacency)

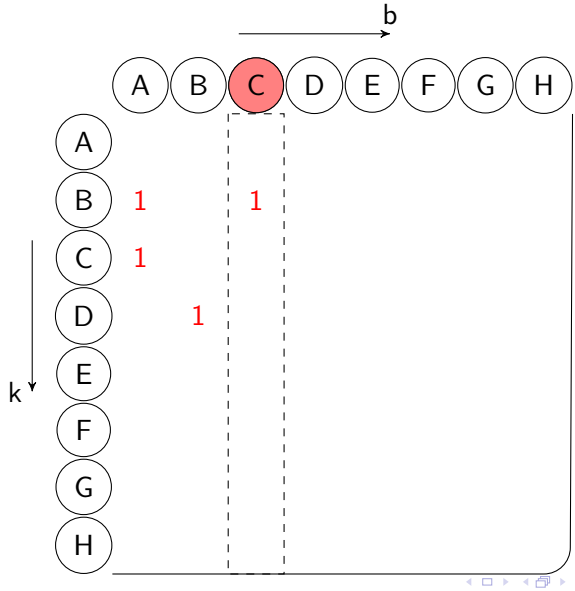




Notes

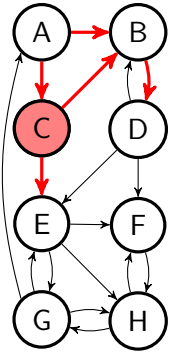
Konstrukce matice A, matice sousednosti (Adjacency)

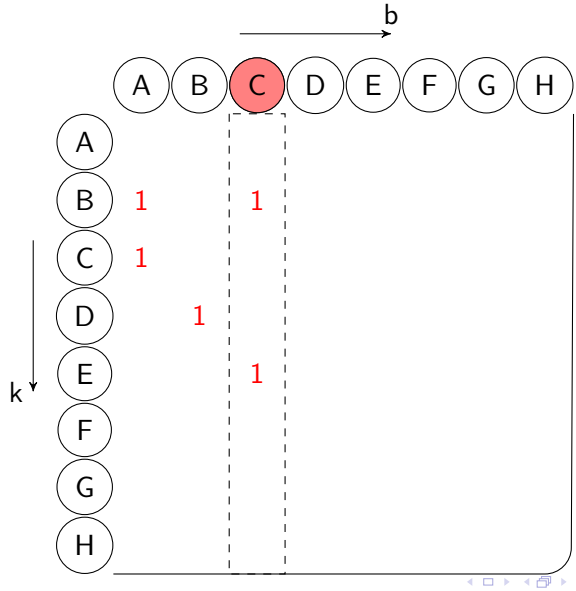




Notes

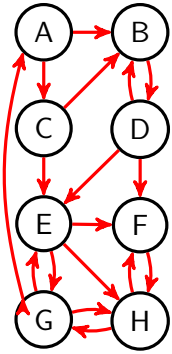
Konstrukce matice A, matice sousednosti (Adjacency)

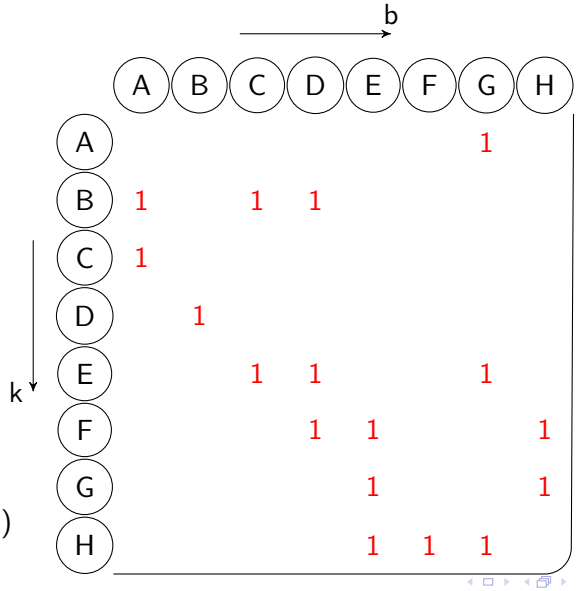




Notes

Konstrukce matice A, matice sousednosti (Adjacency)

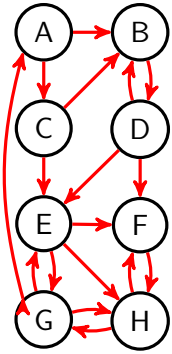


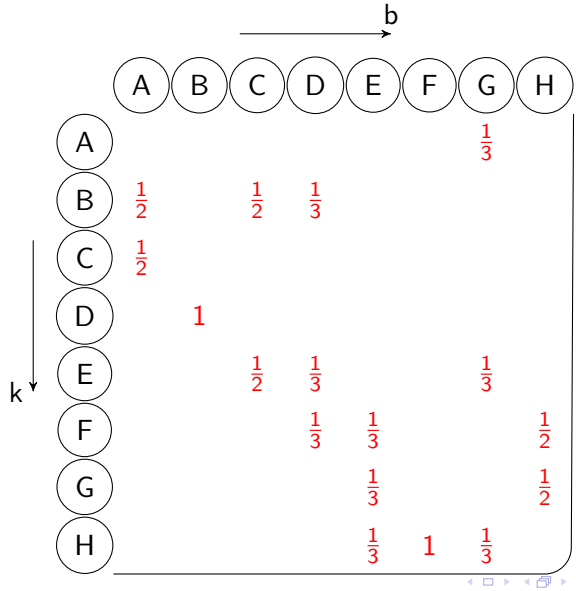


adjacency matrix
(matice sousednosti)

Notes

Konstrukce matice P, normování





Notes

Čtvercové matici, kde je součet prvků v každém sloupci roven 1 se říká *stochastická* matice, říká je i matice přechodů, ...

Systém $W_k = \sum A_{kb} W_b$ ($\mathbf{w} = \mathtt{A}\mathbf{w}$): řešení

$$\mathbf{w} = \mathtt{A}\mathbf{w}$$

Notes

Systém $W_k = \sum A_{kb} W_b$ ($\mathbf{w} = \mathtt{A}\mathbf{w}$): řešení

$$\mathbf{w} = \mathtt{A}\mathbf{w}$$

- co je to za rovnici?

Notes

Systém $W_k = \sum A_{kb} W_b$ ($\mathbf{w} = \mathtt{A}\mathbf{w}$): řešení

$$\lambda \mathbf{w} = \mathtt{A}\mathbf{w}$$

- co je to za rovnici?
- a teď?

Notes

Systém $W_k = \sum A_{kb} W_b$ ($\mathbf{w} = \mathtt{A}\mathbf{w}$): řešení

$$\lambda \mathbf{w} = \mathtt{A}\mathbf{w}$$

- co je to za rovnici?
- a teď?
- ale vždyť to vypadá jako rovnice pro vlastní vektory

Notes

Systém $W_k = \sum A_{kb} W_b$ ($\mathbf{w} = \mathtt{A}\mathbf{w}$): řešení

$$\lambda \mathbf{w} = \mathtt{A}\mathbf{w}$$

- co je to za rovnici?
- a teď?
- ale vždyť to vypadá jako rovnice pro vlastní vektory ($\lambda = 1$)!

Notes

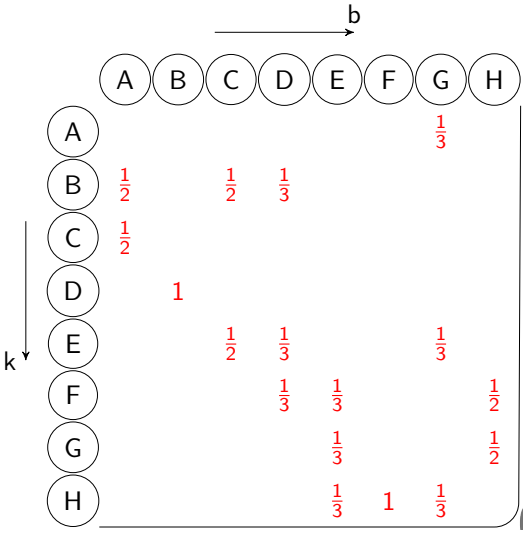
Systém $W_k = \sum A_{kb} W_b$ ($\mathbf{w} = \mathtt{A}\mathbf{w}$): řešení

$$\lambda \mathbf{w} = \mathtt{A}\mathbf{w}$$

- co je to za rovnici?
- a teď?
- ale vždyť to vypadá jako rovnice pro vlastní vektory ($\lambda = 1$)!
- existuje však vlastní vektor pro $\lambda = 1$?

Notes

$\lambda\mathbf{w} = \mathtt{A}\mathbf{w}$, existuje vlastní vektor pro $\lambda = 1$?



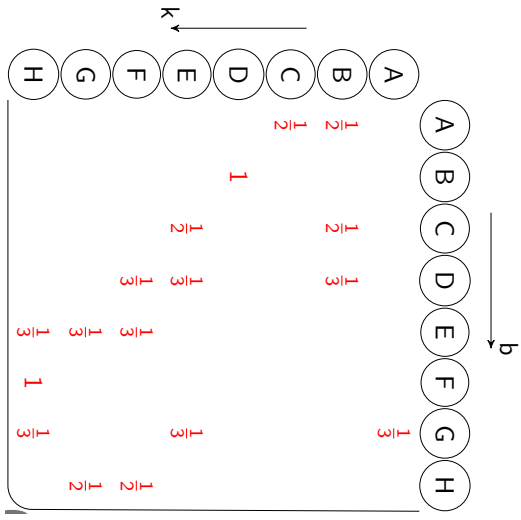
Notes

Víme, že vlastní čísla nalezneme jako řešení rovnice

$$\det(\mathtt{A} - \lambda\mathtt{I}) = 0$$

Determinant matice je stejný pro transponovanou matici.
Další diskuse např. :
https://khanovaskola.cz/video/441/3655-determinant-transponovane-matice

$\lambda\mathbf{w} = \mathtt{A}\mathbf{w}$, existuje vlastní vektor pro $\lambda = 1$?



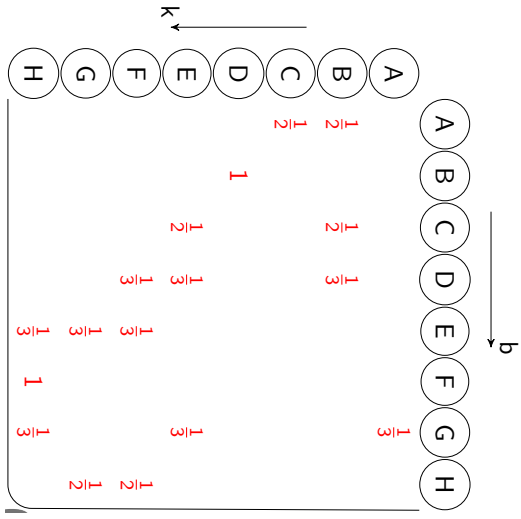
Notes

Víme, že vlastní čísla nalezneme jako řešení rovnice

$$\det(\mathtt{A} - \lambda\mathtt{I}) = 0$$

Determinant matice je stejný pro transponovanou matici.
Další diskuse např. :
https://khanovaskola.cz/video/441/3655-determinant-transponovane-matice

$\lambda\mathbf{w} = \mathtt{A}\mathbf{w}$, existuje vlastní vektor pro $\lambda = 1$?



$$\forall b\colon \sum_k A_{kb} = 1$$

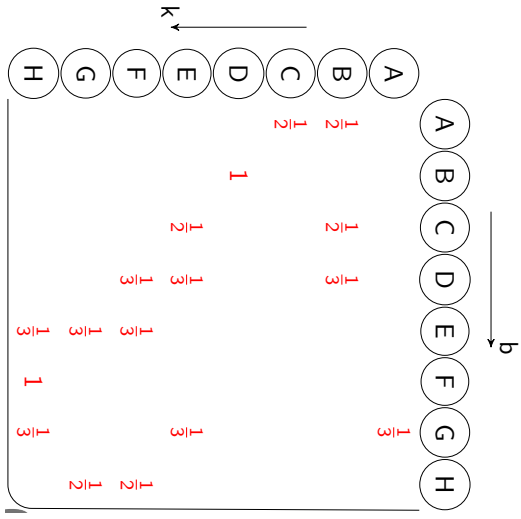
Notes

Víme, že vlastní čísla nalezneme jako řešení rovnice

$$\det(\mathtt{A} - \lambda\mathtt{I}) = 0$$

Determinant matice je stejný pro transponovanou matici.
Další diskuse např. :
https://khanovaskola.cz/video/441/3655-determinant-transponovane-matice

$\lambda\mathbf{w} = \mathtt{A}\mathbf{w}$, existuje vlastní vektor pro $\lambda = 1$?



$$\forall b\colon \sum_k A_{kb} = 1$$

to je však totéž, co

$$\mathtt{A}^\top \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

a vlastní čísla matic a transponovaných matic jsou stejná.

Notes

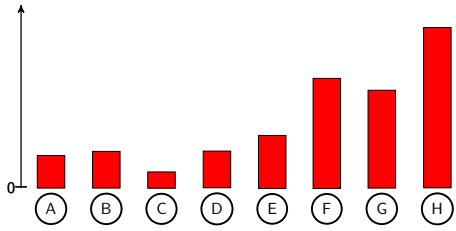
Víme, že vlastní čísla nalezneme jako řešení rovnice

$$\det(\mathtt{A} - \lambda\mathtt{I}) = 0$$

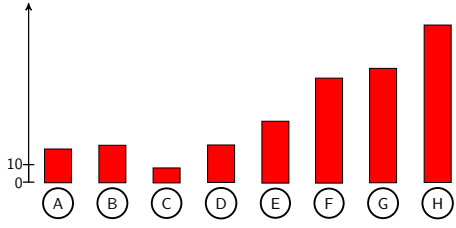
Determinant matice je stejný pro transponovanou matici.
Další diskuse např. :
https://khanovaskola.cz/video/441/3655-determinant-transponovane-matice

Vlastní vektor $\mathbf{v}_1$

- vypočtený vektor $\mathbf{v}_1$



- srovnání s četností navštívení jednotlivých webů (náhodná procházka, 300 iterací)



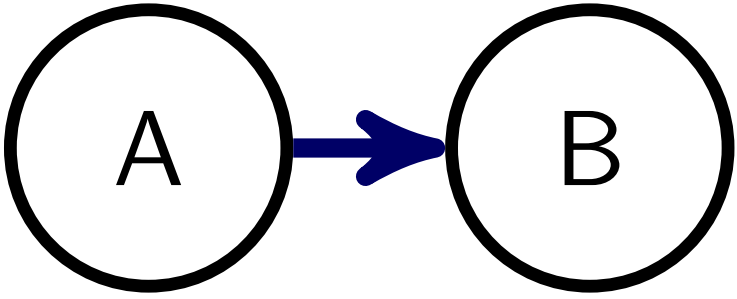
Notes

Máme hotovo?

- Bude fungovat vždy?

Notes

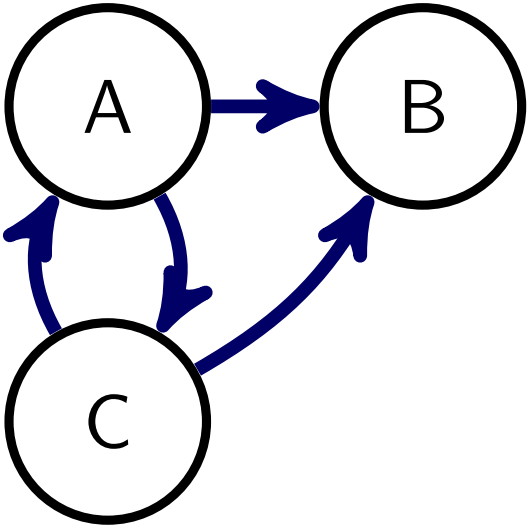
Problém 1 - dangling nodes



Notes

Zkuste si napsat matici sousednosti `A` nebo si odkrokujte náhodnou procházku.

Problém 1a - dangling (sink) nodes; Je něco špatně s A?



Notes

Zkuste si napsat matici sousednosti A nebo si odkrokujte náhodnou procházku.

Google matice G

$$\mathtt{G} = \mathtt{A} + \mathtt{D}$$

D je nulová matice, s výjimkou sloupců, kde je v A nulový sloupec.

Notes

- V podstatě říkám, že se z izolovaného uzlu (stránky) mohu náhodně ocitnout v libovolném jiném uzlu.

Google matice G

$$\mathtt{G} = \mathtt{A} + \mathtt{D}$$

D je nulová matice, s výjimkou sloupců, kde je v A nulový sloupec.

G je vždy stochastická. Máme už vyhráno?

Notes

- V podstatě říkám, že se z izolovaného uzlu (stránky) mohu náhodně ocitnout v libovolném jiném uzlu.

Jak se to ale spočítá?

`A` je velmi velká, n uzlů v našem webu, `eig(A)` má složitost $\mathcal{O}(n^3)$.

Notes

Předpokládeme, že existuje báze vlastních vektorů matice `A`, $\mathbf{v}_j$. Pak můžeme začátek iterace napsat pomocí

$$\mathbf{v}^0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 + \cdots + c_n\mathbf{v}_n$$

Tedy $\mathbf{v}^0 = [c_1, c_2, c_3, \cdots, c_n]^\top$. Pro vlastní vektory platí $\mathtt{A}\mathbf{v}_j = \lambda_j\mathbf{v}_j$. Pak:

$$\begin{aligned}
\mathbf{v}^1 &= \mathtt{A}\mathbf{v}^0 = c_1\mathbf{v}_1 + c_2\lambda_2\mathbf{v}_2 + c_3\lambda_3\mathbf{v}_3 + \cdots + c_n\lambda_n\mathbf{v}_n \\
\mathbf{v}^2 &= \mathtt{A}\mathbf{v}^1 = c_1\mathbf{v}_1 + c_2\lambda_2^2\mathbf{v}_2 + c_3\lambda_3^3\mathbf{v}_3 + \cdots + c_n\lambda_n^2\mathbf{v}_n \\
&\vdots \qquad\quad \vdots \\
\mathbf{v}^k &= \mathtt{A}\mathbf{v}^{k-1} = c_1\mathbf{v}_1 + c_2\lambda_2^k\mathbf{v}_2 + c_3\lambda_3^k\mathbf{v}_3 + \cdots + c_n\lambda_n^k\mathbf{v}_n
\end{aligned}$$

Připomeňme, že

$$1 = \lambda_1 > |\lambda_2| > |\lambda_3| > |\lambda_4| > \cdots > |\lambda_n|$$

Jak se to ale spočítá?

A je velmi velká, n uzlů v našem webu, eig(A) má složitost $\mathcal{O}(n^3)$.
Naštěstí je také řídká!

Notes

Předpokládeme, že existuje báze vlastních vektorů matice A, $\mathbf{v}_j$. Pak můžeme začátek iterace napsat pomocí

$$\mathbf{v}^0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 + \cdots + c_n\mathbf{v}_n$$

Tedy $\mathbf{v}^0 = [c_1, c_2, c_3, \cdots, c_n]^\top$. Pro vlastní vektory platí $\mathtt{A}\mathbf{v}_j = \lambda_j\mathbf{v}_j$. Pak:

$$\begin{aligned}
\mathbf{v}^1 &= \mathtt{A}\mathbf{v}^0 = c_1\mathbf{v}_1 + c_2\lambda_2\mathbf{v}_2 + c_3\lambda_3\mathbf{v}_3 + \cdots + c_n\lambda_n\mathbf{v}_n \\
\mathbf{v}^2 &= \mathtt{A}\mathbf{v}^1 = c_1\mathbf{v}_1 + c_2\lambda_2^2\mathbf{v}_2 + c_3\lambda_3^3\mathbf{v}_3 + \cdots + c_n\lambda_n^2\mathbf{v}_n \\
&\vdots \qquad\quad \vdots \\
\mathbf{v}^k &= \mathtt{A}\mathbf{v}^{k-1} = c_1\mathbf{v}_1 + c_2\lambda_2^k\mathbf{v}_2 + c_3\lambda_3^k\mathbf{v}_3 + \cdots + c_n\lambda_n^k\mathbf{v}_n
\end{aligned}$$

Připomeňme, že

$$1 = \lambda_1 > |\lambda_2| > |\lambda_3| > |\lambda_4| > \cdots > |\lambda_n|$$

Jak se to ale spočítá?

$\mathtt{A}$ je velmi velká, n uzlů v našem webu, $\mathtt{eig}(\mathtt{A})$ má složitost $\mathcal{O}(n^3)$.
Naštěstí je také řídká!

- Metoda Power iteration:
  - začneme s náhodným jednotkovým vektorem $\mathbf{v}$
  - $\mathbf{v} \leftarrow \mathit{normalize}(\mathtt{A}\mathbf{v})$
  - iterujeme

Notes

Předpokládeme, že existuje báze vlastních vektorů matice $\mathtt{A}$, $\mathbf{v}_j$. Pak můžeme začátek iterace napsat pomocí

$$\mathbf{v}^0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 + \cdots + c_n\mathbf{v}_n$$

Tedy $\mathbf{v}^0 = [c_1, c_2, c_3, \cdots, c_n]^\top$. Pro vlastní vektory platí $\mathtt{A}\mathbf{v}_j = \lambda_j\mathbf{v}_j$. Pak:

$$\begin{aligned}
\mathbf{v}^1 &= \mathtt{A}\mathbf{v}^0 = c_1\mathbf{v}_1 + c_2\lambda_2\mathbf{v}_2 + c_3\lambda_3\mathbf{v}_3 + \cdots + c_n\lambda_n\mathbf{v}_n \\
\mathbf{v}^2 &= \mathtt{A}\mathbf{v}^1 = c_1\mathbf{v}_1 + c_2\lambda_2^2\mathbf{v}_2 + c_3\lambda_3^3\mathbf{v}_3 + \cdots + c_n\lambda_n^2\mathbf{v}_n \\
&\vdots \qquad\quad \vdots \\
\mathbf{v}^k &= \mathtt{A}\mathbf{v}^{k-1} = c_1\mathbf{v}_1 + c_2\lambda_2^k\mathbf{v}_2 + c_3\lambda_3^k\mathbf{v}_3 + \cdots + c_n\lambda_n^k\mathbf{v}_n
\end{aligned}$$

Připomeňme, že

$$1 = \lambda_1 > |\lambda_2| > |\lambda_3| > |\lambda_4| > \cdots > |\lambda_n|$$

Jak se to ale spočítá?

$\mathtt{A}$ je velmi velká, n uzlů v našem webu, $\mathtt{eig}(\mathtt{A})$ má složitost $\mathcal{O}(n^3)$.
Naštěstí je také řídká!

- Metoda Power iteration:
  - začneme s náhodným jednotkovým vektorem $\mathbf{v}$
  - $\mathbf{v} \leftarrow \mathit{normalize}(\mathtt{A}\mathbf{v})$
  - iterujeme
- toto funguje, protože hledáme vektor příslušející největšímu vlastnímu číslu

$$1 = \lambda_1 > |\lambda_2| > |\lambda_3| > |\lambda_4| > \cdots > |\lambda_n|$$

- konvergence je tím rychlejší, čím má druhé největší vlastní číslo větší odstup od prvního
- velice jednoduchý algoritmus, efektivní pro obrovské a řídké matice

Notes

Předpokládeme, že existuje báze vlastních vektorů matice $\mathtt{A}$, $\mathbf{v}_j$. Pak můžeme začátek iterace napsat pomocí

$$\mathbf{v}^0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + c_3\mathbf{v}_3 + \cdots + c_n\mathbf{v}_n$$

Tedy $\mathbf{v}^0 = [c_1, c_2, c_3, \cdots, c_n]^\top$. Pro vlastní vektory platí $\mathtt{A}\mathbf{v}_j = \lambda_j\mathbf{v}_j$. Pak:

$$\begin{aligned}
\mathbf{v}^1 &= \mathtt{A}\mathbf{v}^0 = c_1\mathbf{v}_1 + c_2\lambda_2\mathbf{v}_2 + c_3\lambda_3\mathbf{v}_3 + \cdots + c_n\lambda_n\mathbf{v}_n \\
\mathbf{v}^2 &= \mathtt{A}\mathbf{v}^1 = c_1\mathbf{v}_1 + c_2\lambda_2^2\mathbf{v}_2 + c_3\lambda_3^3\mathbf{v}_3 + \cdots + c_n\lambda_n^2\mathbf{v}_n \\
&\vdots \qquad\quad \vdots \\
\mathbf{v}^k &= \mathtt{A}\mathbf{v}^{k-1} = c_1\mathbf{v}_1 + c_2\lambda_2^k\mathbf{v}_2 + c_3\lambda_3^k\mathbf{v}_3 + \cdots + c_n\lambda_n^k\mathbf{v}_n
\end{aligned}$$

Připomeňme, že

$$1 = \lambda_1 > |\lambda_2| > |\lambda_3| > |\lambda_4| > \cdots > |\lambda_n|$$

Power Iteration

$$\mathbf{w}^{k+1} = \mathtt{A}\mathbf{w}^{k}$$

Notes

Power Iteration

$$\mathbf{w}^{k+1} = \mathtt{A}\mathbf{w}^{k}$$

- Konverguje sekvence $\mathbf{w}^{k}$ vždy?

Notes

Power Iteration

$$\mathbf{w}^{k+1} = \mathtt{A}\mathbf{w}^{k}$$

- Konverguje sekvence $\mathbf{w}^k$ vždy?
- Je výsledný stacionární vektor nezávislý na $\mathbf{w}^0$?

Notes

Power Iteration

$$\mathbf{w}^{k+1} = \mathtt{A}\mathbf{w}^{k}$$

- Konverguje sekvence $\mathbf{w}^{k}$ vždy?
- Je výsledný stacionární vektor nezávislý na $\mathbf{w}^{0}$?
- Obsahuje výsledný stacionární vektor vždy to, co chci?

Notes

Power Iteration

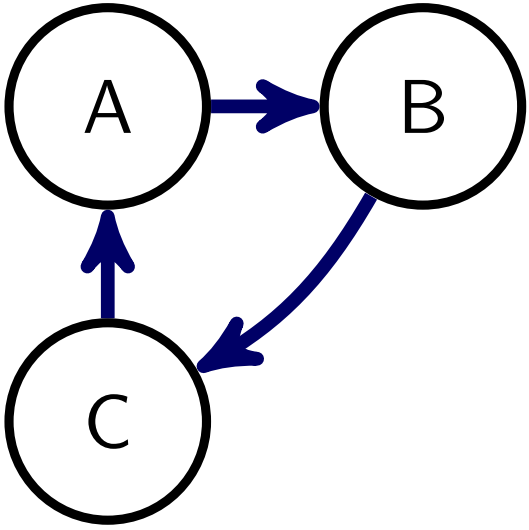
$$\mathbf{w}^{k+1} = \mathtt{A}\mathbf{w}^{k}$$

- Konverguje sekvence $\mathbf{w}^k$ vždy?
- Je výsledný stacionární vektor nezávislý na $\mathbf{w}^0$?
- Obsahuje výsledný stacionární vektor vždy to, co chci?

$3 \times$ Ne!

Notes

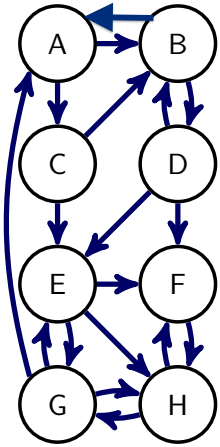
Problém 2 - Konvergence

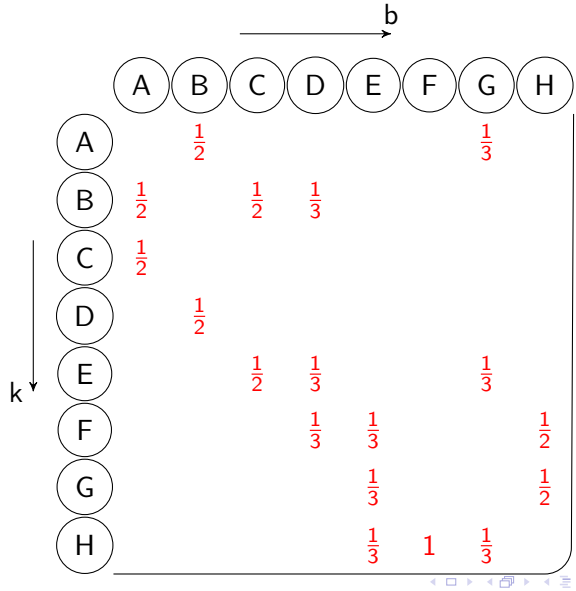


Notes

Vlastní vektor je ok, ale metoda power iter nekonverguje.
`>> page_rank(cycle_matrix)`

Problém 3 - Stacionární vektor





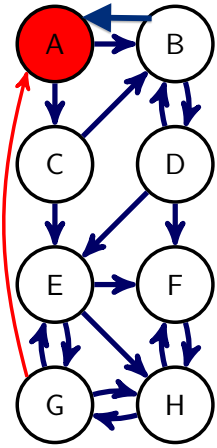
Notes

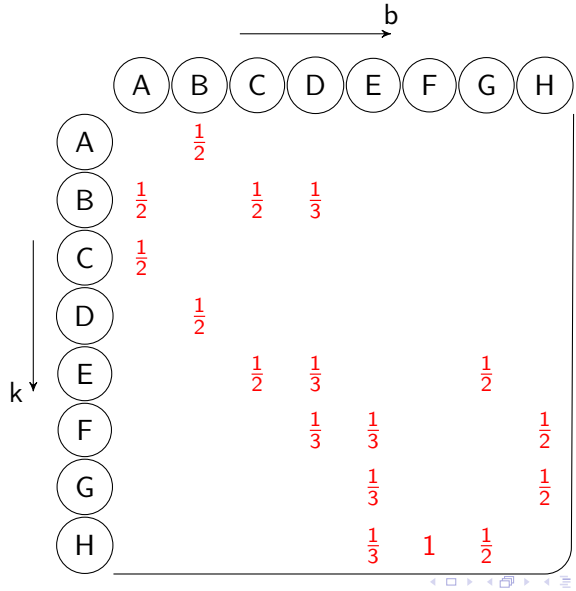
Zkonverguje, i vlastní vektor vypadá ok, ale je to hledané řešení?

```
>> page_rank(adjacency_matrix_deficient)
```

Matice `A` je rozložitelná (reducibilní/reducible)

Problém 3 - Stacionární vektor





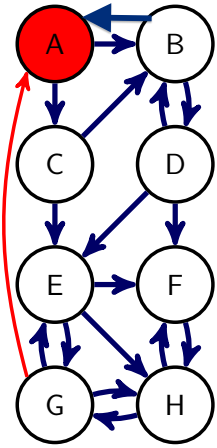
Notes

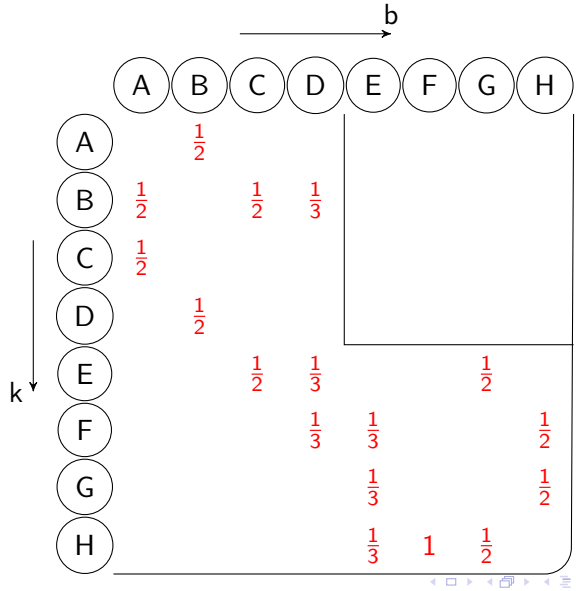
Zkonverguje, i vlastní vektor vypadá ok, ale je to hledané řešení?

```
>> page_rank(adjacency_matrix_deficient)
```

Matice `A` je rozložitelná (reducibilní/reducible)

Problém 3 - Stacionární vektor





Notes

Zkonverguje, i vlastní vektor vypadá ok, ale je to hledané řešení?

```
>> page_rank(adjacency_matrix_deficient)
```

Matice `A` je rozložitelná (reducibilní/reducible)

Google matice

$$\mathtt{G} = \alpha(\mathtt{A} + \mathtt{D}) + (1 - \alpha)\frac{1}{n}\mathtt{I}$$

Notes

$$\mathtt{G}\mathbf{w} = \alpha\mathtt{A}\mathbf{w} + \alpha\mathtt{D}\mathbf{w} + (1 - \alpha)\frac{1}{n}\mathtt{I}\mathbf{w}$$

Matice $\mathtt{A}$ je řídká, to už víme. $\mathtt{D}$ má nulové a nenulové sloupce. Nenulové jsou všechny stejné! Při násobení jednotkovou maticí potřebujeme násobit jen jednou, resp. jedním řádkem matice $(1 - \alpha)\frac{1}{n}\mathtt{I}$. In literature[1], $\alpha = 0.85$ could be found.

[1] David Austin: How Google Finds Your Needle in the Web's Haystack

Google matice

$$\mathtt{G} = \alpha(\mathtt{A} + \mathtt{D}) + (1 - \alpha)\frac{1}{n}\mathtt{I}$$

G už není řídká, přesto je power iteration stále velmi efektivní. Proč?

Notes

$$\mathtt{G}\mathbf{w} = \alpha\mathtt{A}\mathbf{w} + \alpha\mathtt{D}\mathbf{w} + (1 - \alpha)\frac{1}{n}\mathtt{I}\mathbf{w}$$

Matice A je řídká, to už víme. D má nulové a nenulové sloupce. Nenulové jsou všechny stejné! Při násobení jednotkovou maticí potřebujeme násobit jen jednou, resp. jedním řádkem matice $(1 - \alpha)\frac{1}{n}\mathtt{I}$. In literature[2], $\alpha = 0.85$ could be found.

[2] David Austin: How Google Finds Your Needle in the Web's Haystack

Co plyne z odvození PageRanku?

- úžasný efekt mají mnohonásobné pohledy na věc. Jedno může poháněť druhé, analýza mohutně profituje.
- jednoduchá myšlenka, přístupná každému, která změnila svět
- neznamená to ale, že na ni každý přijde (známý efekt ,,no jistě!", který se dostavuje po odhalení řešení). Dnes nám přijde velice přirozené, že všechny souřadné soustavy by měly být rovnocenné (Einsteinova teorie relativity)
- jednoduché věci (většinou) fungují
- je to potvrzení toho, že je třeba analyzovat problém postupně a postupovat od principielně jednoduššího řešení ke složitějšímu

Notes

Google story

- 1995: Larry Page a Sergey Brin se potkali na Stanfordu (bylo jim kolem 21 let)
- 1996-1997: odvození, implementace hledacího stroje, vymyšlení názvu Google
- 1998: 'lepší než cokoli jiného'
- 2004: IPO na Wall St, \$85/akcie (market cap \$25 miliard)
- od počátku skutečný leader v inovacích

Notes

Reference

- wikipedia
- David Austin: How Google Finds Your Needle in the Web's Haystack
- KURT BRYAN AND TANYA LEISE: The $25,000,000,000 Eigenvector: The Linear Algebra Behind Google

Notes