

Ako ďaleko je od teórie k praxi?

(exkurzia do teoretickej informatiky)

Michal „Mišof“ Forišek

Department of Theoretical Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University
Bratislava, Slovakia

7. novembra 2012

Čo robí teoretický informatik?

- **Aplikovaný výskum:** príde niekto z praxe s problémom, skúmame, čo s ním vieme robiť.
- **Základný výskum:** „hej, toto vyzerá zaujímavo!“
- Má základný výskum zmysel?

Má základný výskum zmysel?

Zmysel základného výskumu závisí od témy.
(Posúdiť vhodnosť témy je ťažké.)

Výstup dobrého základného výskumu:

- hlbšie poznanie problematiky (napr. NP-úplnosť)
- príprava na budúcnosť (napr. kvantová faktorizácia)

Čo robí teoretický informatik?

- **Aplikovaný výskum:** príde niekto z praxe s problémom, skúmame, čo s ním vieme robiť.
- **Základný výskum:** „hej, toto vyzerá zaujímavo!“
- Má základný výskum zmysel?

Má základný výskum zmysel?

Zmysel základného výskumu závisí od témy.
(Posúdiť vhodnosť témy je ťažké.)

Výstup dobrého základného výskumu:

- hlbšie poznanie problematiky (napr. NP-úplnosť)
- príprava na budúcnosť (napr. kvantová faktorizácia)

Vyhľadávanie v bežnom texte

Bežná webstránka má do 10 kB *textu*.

Ctrl-F (find) v prehliadači funguje takmer okamžite.

Kniha *Isaac Asimov: Ja, robot* má 425 kB.

Vyhľadanie všetkých výskytov
slova „robot“: 0.007 s.

Kým máme jedného človeka a málo textu,
všetko funguje ako má.

Vyhľadávanie v bežnom texte

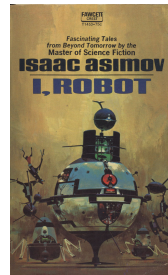
Bežná webstránka má do 10 kB *textu*.

Ctrl-F (find) v prehliadači funguje takmer okamžite.

Kniha *Isaac Asimov: Ja, robot* má 425 kB.

Vyhľadanie všetkých výskytov
slova „robot“: 0.007 s.

Kým máme jedného človeka a málo textu,
všetko funguje ako má.



Vyhľadávanie v bežnom texte

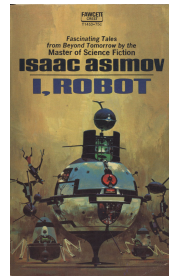
Bežná webstránka má do 10 kB *textu*.

Ctrl-F (find) v prehliadači funguje takmer okamžite.

Kniha *Isaac Asimov: Ja, robot* má 425 kB.

Vyhľadanie všetkých výskytov
slova „robot“: 0.007 s.

Kým máme jedného človeka a málo textu,
všetko funguje ako má.



Množstvo informácie rastie

- Disketa: posledné bežné mali **1.44 MB**



- CD: typicky do **700 MB**
(plné auto kníh)
- DVD: najrozšírenejšie majú **4.7 GB**

Množstvo informácie rastie

- Disketa: posledné bežné mali **1.44 MB**
- CD: typicky do **700 MB**
(plné auto kníh)
- DVD: najrozšírenejšie majú **4.7 GB**
- BluRay disky: bežne **50 GB**
Video v kvalite HD: **28 Mb/s**

Inými slovami: každú sekundu 3.5 megabajtu údajov.
(Každé 4 desatiny sekundy meniť disketu?)

- Nový pevný disk: **1 TB**

Množstvo informácie rastie

- Disketa: posledné bežné mali **1.44 MB**
- CD: typicky do **700 MB**
(plné auto kníh)
- DVD: najrozšírenejšie majú **4.7 GB**
- BluRay disky: bežne **50 GB**
Video v kvalite HD: **28 Mb/s**

Inými slovami: každú sekundu 3.5 megabajtu údajov.
(Každé 4 desatiny sekundy meniť disketu?)

- Nový pevný disk: **1 TB**

Množstvo informácie rastie

- Disketa: posledné bežné mali **1.44 MB**
- CD: typicky do **700 MB**
(plné auto kníh)
- DVD: najrozšírenejšie majú **4.7 GB**
- BluRay disky: bežne **50 GB**
Video v kvalite HD: **28 Mb/s**

Inými slovami: každú sekundu 3.5 megabajtu údajov.
(Každé 4 desatiny sekundy meniť disketu?)

- Nový pevný disk: **1 TB**

Množstvo informácie rastie

- Disketa: posledné bežné mali **1.44 MB**
- CD: typicky do **700 MB**
(plné auto kníh)
- DVD: najrozšírenejšie majú **4.7 GB**
- BluRay disky: bežne **50 GB**
Video v kvalite HD: **28 Mb/s**

Inými slovami: každú sekundu 3.5 megabajtu údajov.
(Každé 4 desatiny sekundy meniť disketu?)

- Nový pevný disk: **1 TB**

Množstvo informácie rastie

- Disketa: posledné bežné mali **1.44 MB**
- CD: typicky do **700 MB**
(plné auto kníh)
- DVD: najrozšírenejšie majú **4.7 GB**
- BluRay disky: bežne **50 GB**
Video v kvalite HD: **28 Mb/s**

Inými slovami: každú sekundu 3.5 megabajtu údajov.
(Každé 4 desatiny sekundy meniť disketu?)

- Nový pevný disk: **1 TB**

Množstvo informácie rastie ešte viac

americká Library of Congress: **50 TB** informácií



Množstvo informácie rastie ešte viac

špeciálne efekty pre film *Avatar*: **1 PB** = 1000 TB



Množstvo informácie rastie ešte viac

hra *World of Warcraft*: **1.3 PB**



Množstvo informácie rastie ešte viac

fyzikálny experiment LHC: získame **15 PB** dát za rok



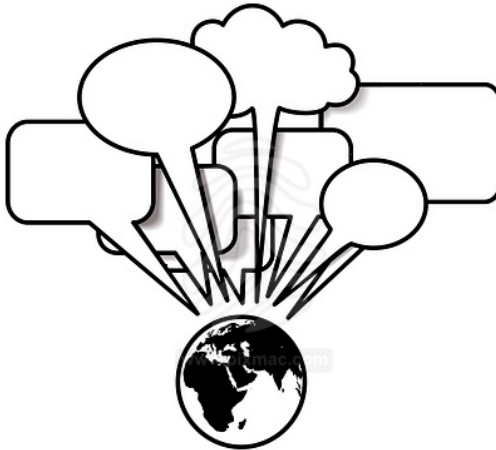
Množstvo informácie rastie ešte viac

Google: spracuje **24 PB** dát denne
(celé dáta používané pri vyhľadávaní majú vyše 100 PB)



Množstvo informácie rastie ešte viac

Všetky slová, ktoré kto kedy povedal: **5 EB = 5000 PB**



Web rastie

- 1998: Založené Google, pozná 26 miliónov stránok
- 2008: Google pozná **bilión** (10^{12}) rôznych URL,
zodpovedajú odhadom **20 miliárdám** unikátnych webstránok.

Ako dlho by trvalo nájsť výskyty slova „robot“ na celom webe?

rátajme spolu:

20 miliárd stránok $\times$ trebárs 1 kB = 20 TB textu.

to je zhruba 50 miliónov krát viac ako Asimovova kniha

bude to trvať 50 miliónov krát dlhšie

$\Rightarrow$ niekoľko dní

Web rastie

- 1998: Založené Google, pozná 26 miliónov stránok
- 2008: Google pozná **bilión** (10^{12}) rôznych URL, zodpovedajú odhadom **20 miliárdám** unikátnych webstránok.

Ako dlho by trvalo nájsť výskyty slova „robot“ na celom webe?

rátajme spolu:

20 miliárd stránok $\times$ trebárs 1 kB = 20 TB textu.

to je zhruba 50 miliónov krát viac ako Asimovova kniha

bude to trvať 50 miliónov krát dlhšie

$\Rightarrow$ niekoľko dní

Web rastie

- 1998: Založené Google, pozná 26 miliónov stránok
- 2008: Google pozná **bilión** (10^{12}) rôznych URL, zodpovedajú odhadom **20 miliárdám** unikátnych webstránok.

Ako dlho by trvalo nájsť výskyty slova „robot“ na celom webe?

rátajme spolu:

20 miliárd stránok $\times$ trebárs 1 kB = 20 TB textu.

to je zhruba 50 miliónov krát viac ako Asimovova kniha

bude to trvať 50 miliónov krát dlhšie

$\Rightarrow$ niekoľko dní

Web rastie

- 1998: Založené Google, pozná 26 miliónov stránok
- 2008: Google pozná **bilión** (10^{12}) rôznych URL, zodpovedajú odhadom **20 miliárdám** unikátnych webstránok.

Ako dlho by trvalo nájsť výskyty slova „robot“ na celom webe?

rátajme spolu:

20 miliárd stránok $\times$ trebárs 1 kB = 20 TB textu.

to je zhruba 50 miliónov krát viac ako Asimovova kniha

bude to trvať 50 miliónov krát dlhšie

$\Rightarrow$ niekoľko dní

Web rastie ešte viac

Realita: **0.25 sekundy**

Niekoľko dní čakať na výsledok nechceme!

Riešenie: paralelizácia

„nech si Google nakúpi viac počítačov :–)“

Na čo zabúdame?

Na ostatných ľudí!

Denne Google spracuje **300 miliónov** vyhľadávaní.

To by bolo 1 a pol miliardy dní výpočtu.

Ak by to chceli za deň stihnúť, treba miliardu a pol počítačov.

Web rastie ešte viac

Realita: **0.25 sekundy**

Niekoľko dní čakať na výsledok nechceme!

Riešenie: paralelizácia

„nech si Google nakúpi viac počítačov :-)“

Na čo zabúdame?

Na ostatných ľudí!

Denne Google spracuje **300 miliónov** vyhľadávaní.

To by bolo 1 a pol miliardy dní výpočtu.

Ak by to chceli za deň stihnúť, treba miliardu a pol počítačov.

Web rastie ešte viac

Realita: **0.25 sekundy**

Niekoľko dní čakať na výsledok nechceme!

Riešenie: paralelizácia

„nech si Google nakúpi viac počítačov :-)“

Na čo zabúdame?

Na ostatných ľudí!

Denne Google spracuje **300 miliónov** vyhľadávaní.

To by bolo 1 a pol miliardy dní výpočtu.

Ak by to chceli za deň stihnúť, treba miliardu a pol počítačov.

Potrebuje lepšie algoritmy

Prvý nápad: index

pre každé slovo budeme mať zoznam stránok, ktoré ho obsahujú.

Na jednej strane žiadna výhra (veľa pamäte; čo zložitejšie otázky?).
Na druhej strane ani toto nestačí.

Potrebujeme lepšie algoritmy

Prvý nápad: index

pre každé slovo budeme mať zoznam stránok, ktoré ho obsahujú.

Na jednej strane žiadna výhra (veľa pamäte; čo zložitejšie otázky?).

Na druhej strane ani toto nestačí.

Potrebuje lepšie algoritmy

Prvý nápad: index

pre každé slovo budeme mať zoznam stránok, ktoré ho obsahujú.

Na jednej strane žiadna výhra (veľa pamäte; čo zložitejšie otázky?).
Na druhej strane ani toto nestačí.

Problém?



Ktoré z nich ukázať užívateľovi?

Ako spoznať, ktoré sú relevantné?

PageRank

Nový algoritmus pri zrode Googlu.

Autori Larry Page (podľa neho sa volá) a Sergej Brin.

Vyhodnocuje, ktoré stránky sú dôležité a ktoré nie.

Myšlienka: dôležité stránky sú tie, ktoré ľudia navštevujú.

Problém: ale my nevieme, ktoré to sú...

Riešenie: oni nám to povedia – spravia linky!

PageRank

Nový algoritmus pri zrode Googlu.

Autori Larry Page (podľa neho sa volá) a Sergej Brin.

Vyhodnocuje, ktoré stránky sú dôležité a ktoré nie.

Myšlienka: dôležité stránky sú tie, ktoré ľudia navštevujú.

Problém: ale my nevieme, ktoré to sú...

Riešenie: oni nám to povedia – spravia linky!

PageRank

Nový algoritmus pri zrode Googlu.

Autori Larry Page (podľa neho sa volá) a Sergej Brin.

Vyhodnocuje, ktoré stránky sú dôležité a ktoré nie.

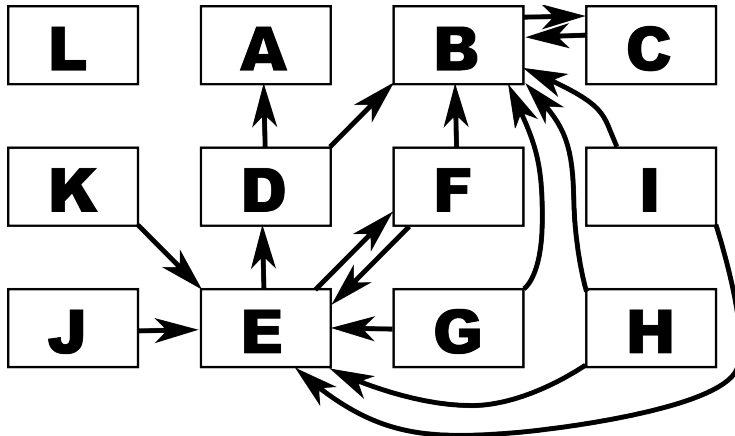
Myšlienka: dôležité stránky sú tie, ktoré ľudia navštevujú.

Problém: ale my nevieme, ktoré to sú...

Riešenie: oni nám to povedia – spravia linky!

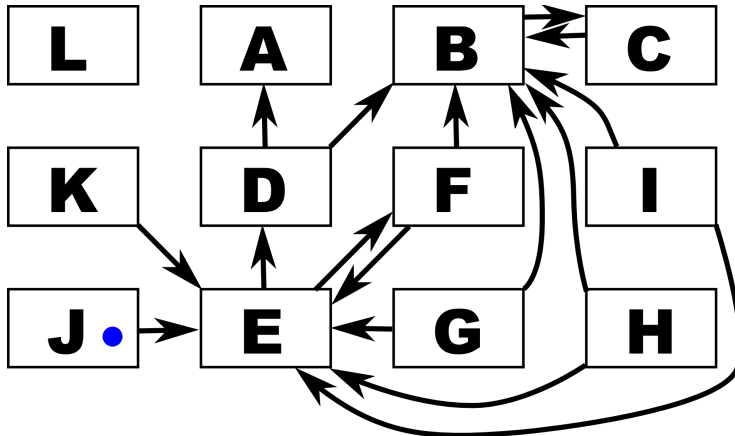
Základná myšlienka PageRanku

Človek náhodne kliká po linkách.



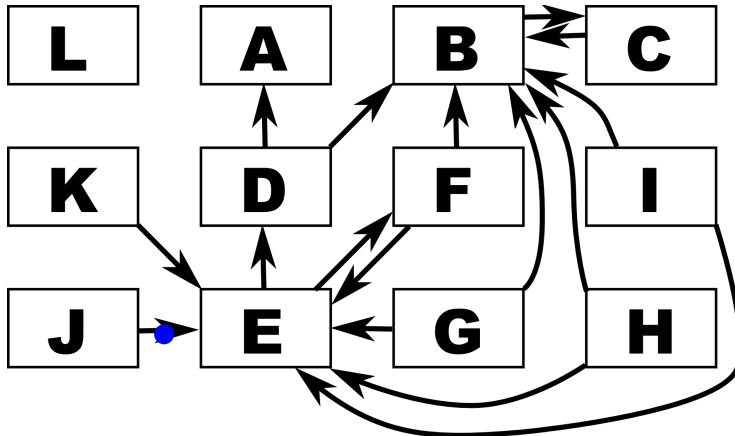
Základná myšlienka PageRanku

Človek náhodne kliká po linkách.



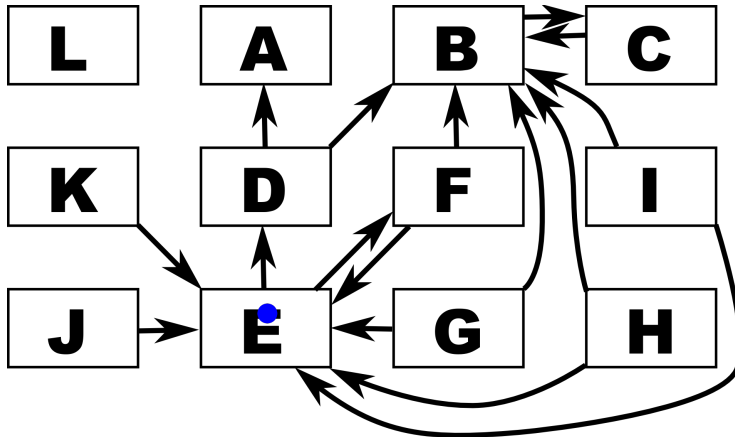
Základná myšlienka PageRanku

Človek náhodne kliká po linkách.



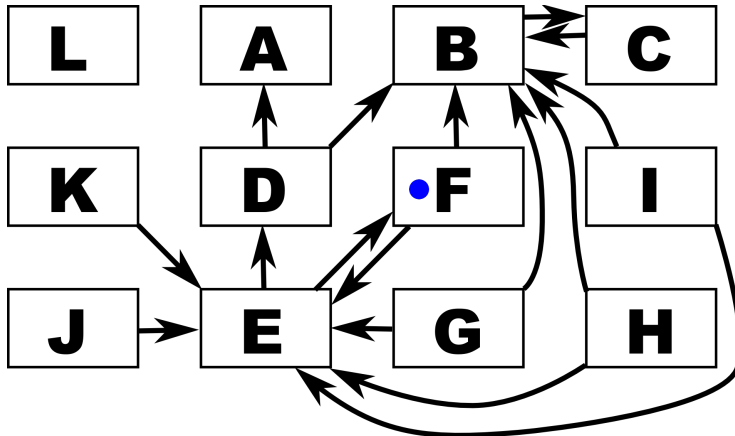
Základná myšlienka PageRanku

Človek náhodne kliká po linkách.



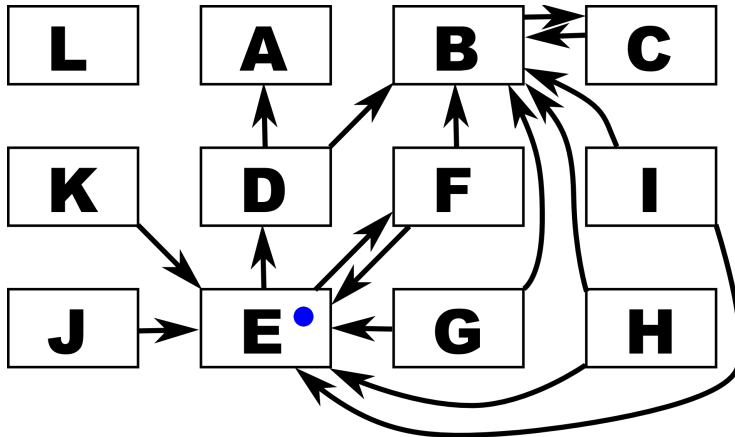
Základná myšlienka PageRanku

Človek náhodne kliká po linkách.



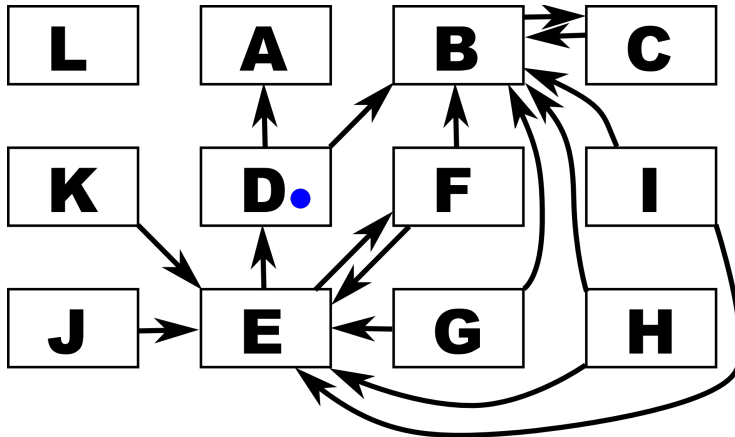
Základná myšlienka PageRanku

Človek náhodne kliká po linkách.



Základná myšlienka PageRanku

Človek náhodne kliká po linkách.



Nie je náhodné klikanie blbosť?

Človek predsa nekliká po linkách *náhodne*!

Jeden človek nie – ale všetci dokopy áno!

Presnejší model správania sa ľudí:

- 85% prípadov: klikne na linku
- 15% prípadov: odíde na nesúvisiacu stránku

Nie je náhodné klikanie blbosť?

Človek predsa nekliká po linkách *náhodne*!

Jeden človek nie – ale všetci dokopy áno!

Presnejší model správania sa ľudí:

- 85% prípadov: klikne na linku
- 15% prípadov: odíde na nesúvisiacu stránku

Nie je náhodné klikanie blbosť?

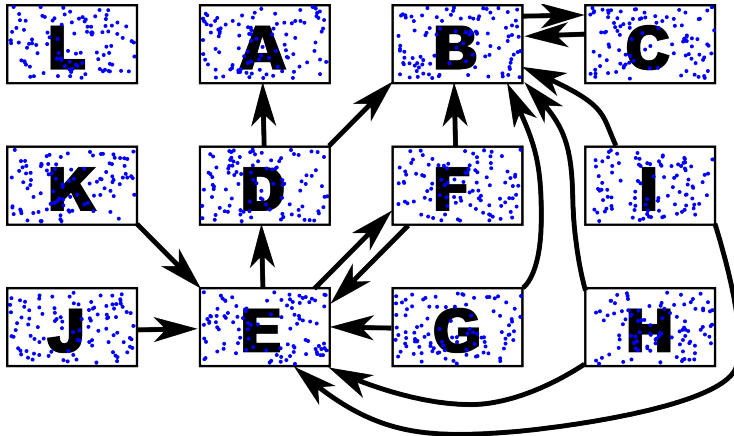
Človek predsa nekliká po linkách *náhodne*!

Jeden človek nie – ale všetci dokopy áno!

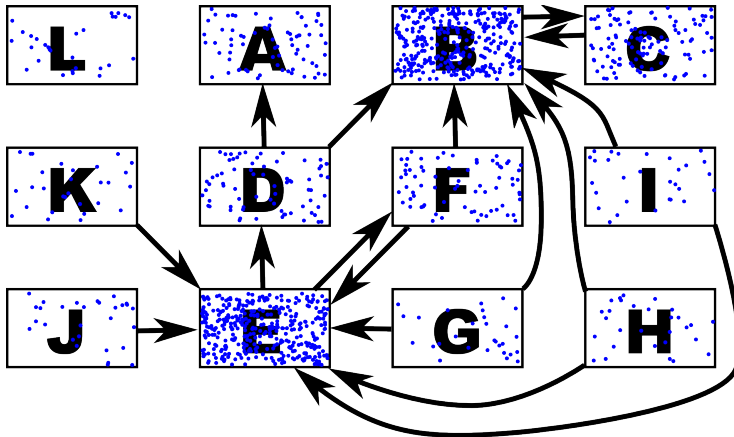
Presnejší model správania sa ľudí:

- 85% prípadov: klikne na linku
- 15% prípadov: odíde na nesúvisiacu stránku

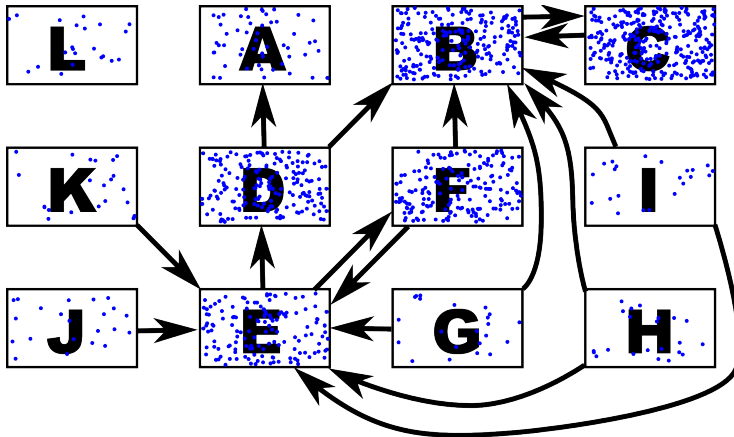
Simulácia pre veľa ľudí



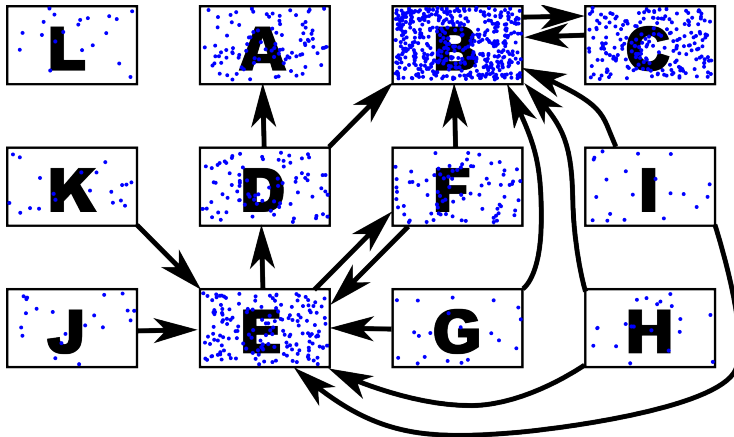
Simulácia pre veľa ľudí



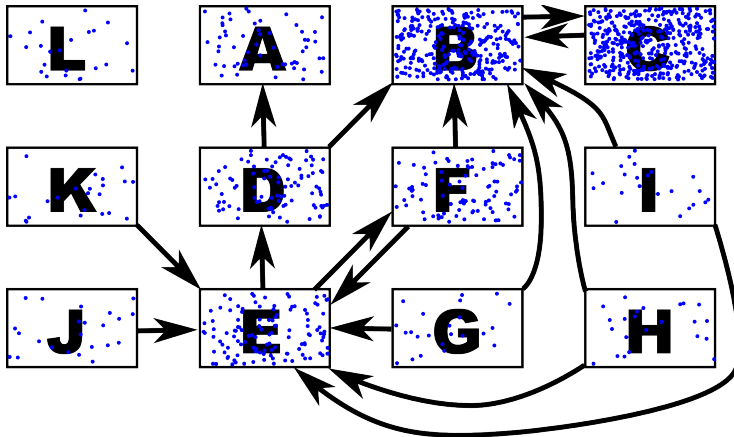
Simulácia pre veľa ľudí



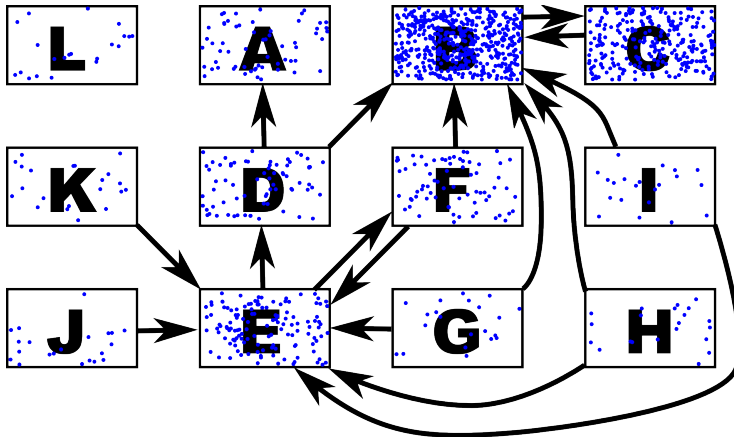
Simulácia pre veľa ľudí



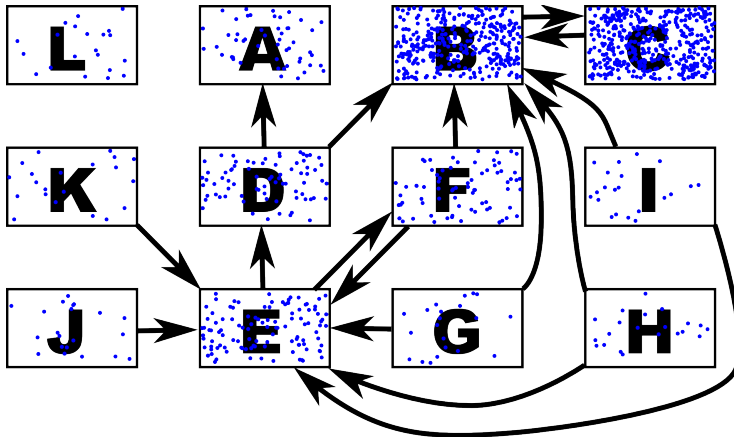
Simulácia pre veľa ľudí



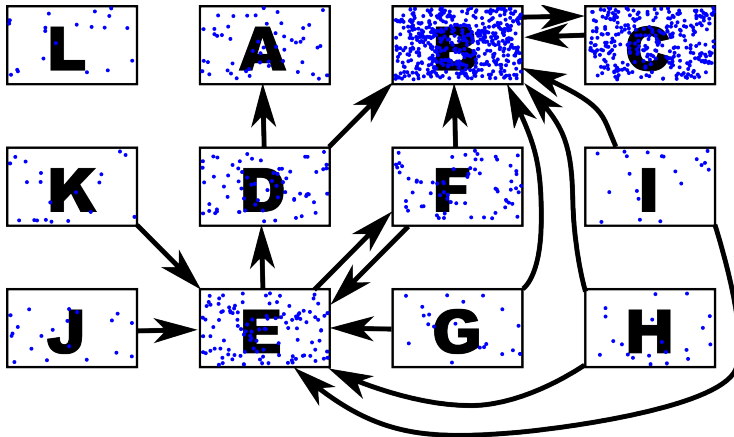
Simulácia pre veľa ľudí



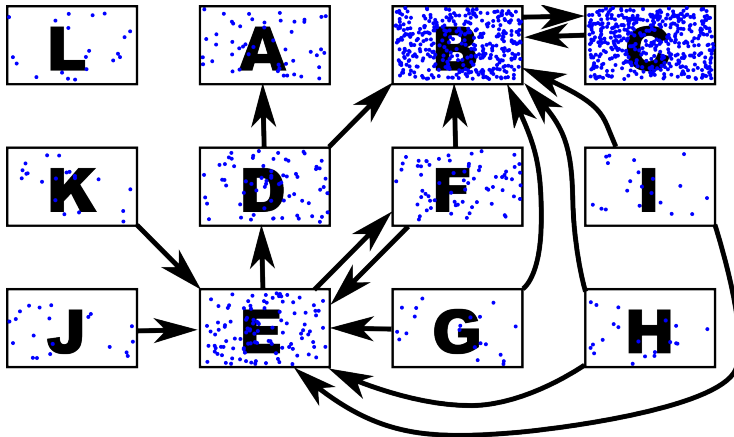
Simulácia pre veľa ľudí



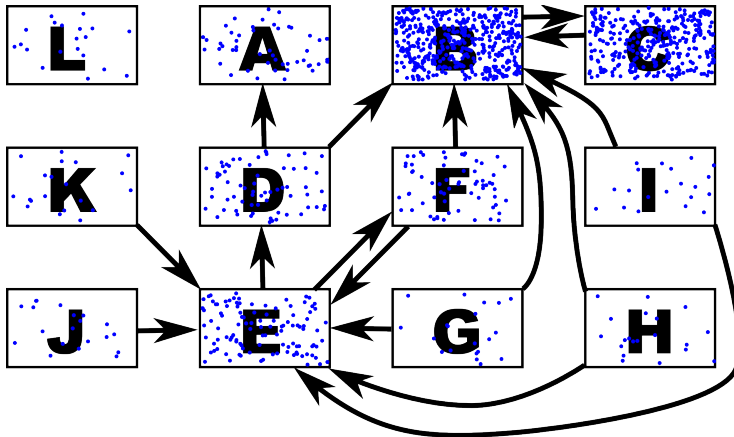
Simulácia pre veľa ľudí



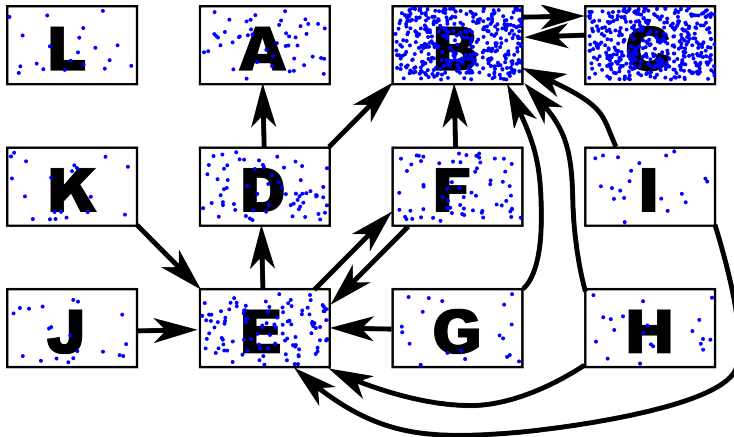
Simulácia pre veľa ľudí



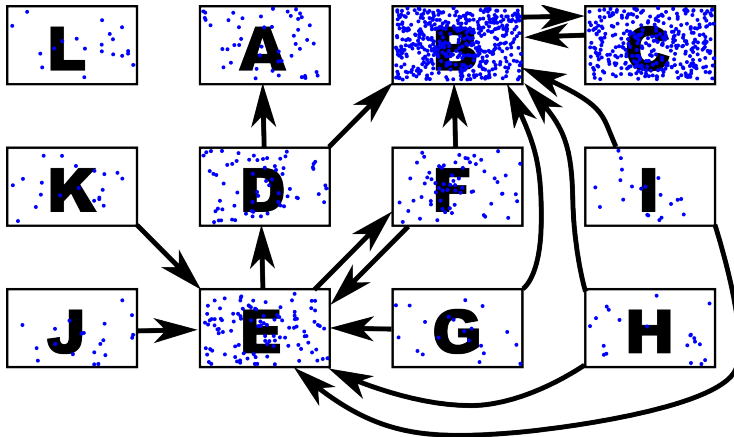
Simulácia pre veľa ľudí



Simulácia pre veľa ľudí

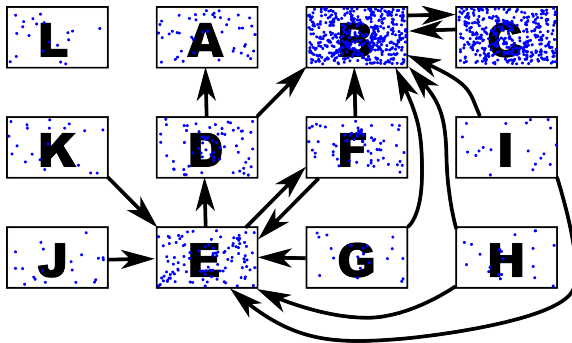


Simulácia pre veľa ľudí



Sústava lineárnych rovníc

Ako zistiť, k čomu to celé smeruje?

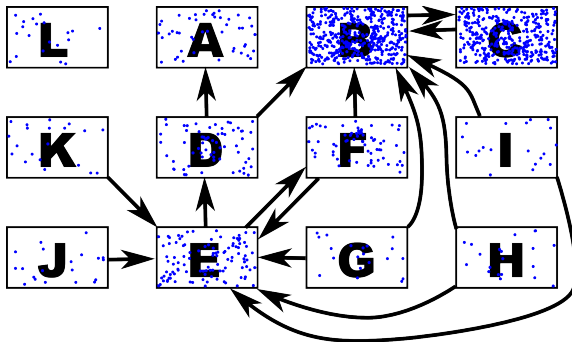


Rovnice!

$$\text{rank}_E = 0.15 \cdot \frac{1}{12} + 0.85 \cdot \left(\frac{\text{rank}_F}{2} + \frac{\text{rank}_G}{2} + \dots + \text{rank}_K \right)$$

Sústava lineárnych rovníc

Ako zistiť, k čomu to celé smeruje?



Rovnice!

$$\text{rank}_E = 0.15 \cdot \frac{1}{12} + 0.85 \cdot \left(\frac{\text{rank}_F}{2} + \frac{\text{rank}_G}{2} + \dots + \text{rank}_K \right)$$

Sústava lineárnych rovníc

Gaussova eliminačná metóda:

$$\begin{array}{rrcr} x & + & 2y & - & z & = & -1 \\ -x & - & 3y & + & 5z & = & 10 \\ 3x & + & y & + & 4z & = & 16 \end{array}$$

Sústava lineárnych rovníc

Gaussova eliminačná metóda:

$$\begin{array}{rclcl} x + & 2y - & z = & -1 \\ & -y + & 4z = & 9 \\ & -5y + & 7z = & 19 \end{array}$$

Sústava lineárnych rovníc

Gaussova eliminačná metóda:

$$\begin{array}{rclcl} x + & 2y - & z = & -1 \\ & -y + & 4z = & 9 \\ & & 13z = & 26 \end{array}$$

a postupným dosadzovaním máme $z = 2$, $y = -1$, $x = 3$

Sústava lineárnych rovníc

Gaussova eliminačná metóda:

$$\begin{array}{rclcl} x + & 2y - & z = & -1 \\ & -y + & 4z = & 9 \\ & & 13z = & 26 \end{array}$$

a postupným dosadzovaním máme $z = 2$, $y = -1$, $x = 3$

Sústava lineárnych rovníc

Gaussova eliminačná metóda je algoritmus.

n premenných $\Rightarrow$ rádovo n^3 operácií

Problém:

miliardy webstránok $\Rightarrow$ miliardy premenných

miliardy premenných $\Rightarrow$ miliardy miliárd miliárd operácií

$\Rightarrow$ stotisíc miliárd rokov výpočtu?!

A samotnú sústavu rovníc tvorí miliarda miliárd čísel

$\Rightarrow$ exabajty pamäte?!

Sústava lineárnych rovníc

Gaussova eliminačná metóda je algoritmus.

n premenných $\Rightarrow$ rádovo n^3 operácií

Problém:

miliardy webstránok $\Rightarrow$ miliardy premenných

miliardy premenných $\Rightarrow$ miliardy miliárd miliárd operácií

$\Rightarrow$ stotisíc miliárd rokov výpočtu?!

A samotnú sústavu rovníc tvorí miliarda miliárd čísel

$\Rightarrow$ exabajty pamäte?!

Sústava lineárnych rovníc

Gaussova eliminačná metóda je algoritmus.

n premenných $\Rightarrow$ rádovo n^3 operácií

Problém:

miliardy webstránok $\Rightarrow$ miliardy premenných

miliardy premenných $\Rightarrow$ miliardy miliárd miliárd operácií

$\Rightarrow$ stotisíc miliárd rokov výpočtu?!

A samotnú sústavu rovníc tvorí miliarda miliárd čísel

$\Rightarrow$ exabajty pamäte?!

Sústava lineárnych rovníc

Gaussova eliminačná metóda je algoritmus.

n premenných $\Rightarrow$ rádovo n^3 operácií

Problém:

miliardy webstránok $\Rightarrow$ miliardy premenných

miliardy premenných $\Rightarrow$ miliardy miliárd miliárd operácií

$\Rightarrow$ stotisíc miliárd rokov výpočtu?!

A samotnú sústavu rovníc tvorí miliarda miliárd čísel

$\Rightarrow$ exabajty pamäte?!

Sústava lineárnych rovníc

Riešenie problému: nové algoritmy.

Šikovne využijeme to, že naša sústava rovníc je „riedka“:
Každá webstránka ukazuje len na pár iných.

Prototyp Googlu: jeden počítač za niekoľko hodín spočítal
PageRank pre „celý internet“ (vtedy pár miliónov stránok)

Súčasnosť: PageRank je jedným z **mnohých** faktorov stále
používaných Googlom.

Sústava lineárnych rovníc

Riešenie problému: nové algoritmy.

Šikovne využijeme to, že naša sústava rovníc je „riedka“:
Každá webstránka ukazuje len na pár iných.

Prototyp Googlu: jeden počítač za niekoľko hodín spočítal
PageRank pre „celý internet“ (vtedy pár miliónov stránok)

Súčasnoscť: PageRank je jedným z **mnohých** faktorov stále
používaných Googlom.

Sústava lineárnych rovníc

Riešenie problému: nové algoritmy.

Šikovne využijeme to, že naša sústava rovníc je „riedka“:
Každá webstránka ukazuje len na pár iných.

Prototyp Googlu: jeden počítač za niekoľko hodín spočítal
PageRank pre „celý internet“ (vtedy pár miliónov stránok)

Súčasnoscť: PageRank je jedným z **mnohých** faktorov stále
používaných Googlom.

Úloha o majoritnom prvku

Zadanie

Na vstupe je postupnosť n prvkov.

Jeden z nich sa v tejto postupnosti vyskytuje viac ako $(n/2)$ -krát.

Nájdite ho.

- Naivné riešenie: pre každý prvok spočítaj výskyty $\Theta(n^2)$.
- Lepšie: usporiadaj pole, rovnaké budú za sebou $\Theta(n \log n)$.
- Ešte lepšie: stačí nájsť medián $\Theta(n)$.

Úloha o majoritnom prvku

Zadanie

Na vstupe je postupnosť n prvkov.

Jeden z nich sa v tejto postupnosti vyskytuje viac ako $(n/2)$ -krát.

Nájdite ho.

- Naivné riešenie: pre každý prvok spočítaj výskyty $\Theta(n^2)$.
- Lepšie: usporiadaj pole, rovnaké budú za sebou $\Theta(n \log n)$.
- Ešte lepšie: stačí nájsť medián $\Theta(n)$.

Úloha o majoritnom prvku

Zadanie

Na vstupe je postupnosť n prvkov.

Jeden z nich sa v tejto postupnosti vyskytuje viac ako $(n/2)$ -krát.
Nájdite ho.

- Naivné riešenie: pre každý prvok spočítaj výskyty $\Theta(n^2)$.
- Lepšie: usporiadaj pole, rovnaké budú za sebou $\Theta(n \log n)$.
- Ešte lepšie: stačí nájsť medián $\Theta(n)$.

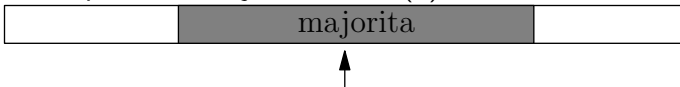
Úloha o majoritnom prvku

Zadanie

Na vstupe je postupnosť n prvkov.

Jeden z nich sa v tejto postupnosti vyskytuje viac ako $(n/2)$ -krát.
Nájdite ho.

- Naivné riešenie: pre každý prvok spočítaj výskyty $\Theta(n^2)$.
- Lepšie: usporiadať pole, rovnaké budú za sebou $\Theta(n \log n)$.
- Ešte lepšie: stačí nájsť medián $\Theta(n)$.



Úloha o majoritnom prvku a jej lepšie riešenia

Nevýhody hľadania mediánu

Algoritmy vo worst-case $\Theta(n)$ sú komplikované.
Potreba pamäte $\Theta(n)$ – nedá sa zlepšiť?

Nové pozorovanie

Keď nájdem dva rôzne prvky, môžem *oba* zahodiť.
(Zahodím ≥ 1 zlý, ≤ 1 dobrý, majorita sa zachová.)

Do dvojíc! (Boyer, Moore 1981)

Rozdelíme všetky prvky do dvojíc (a jedného navyše pre nepárne n).
Dvojice v ktorých sú prvky rôzne zahodíme.
Teraz nutne máme viac „dobrých“ ako „zlých“ dvojíc.
Z každej si necháme jeden prvok, majorita ostáva zachovaná.

Úloha o majoritnom prvku a jej lepšie riešenia

Nevýhody hľadania mediánu

Algoritmy vo worst-case $\Theta(n)$ sú komplikované.
Potreba pamäte $\Theta(n)$ – nedá sa zlepšiť?

Nové pozorovanie

Keď nájdem dva rôzne prvky, môžem *oba* zahodiť.
(Zahodím ≥ 1 zlý, ≤ 1 dobrý, majorita sa zachová.)

Do dvojíc! (Boyer, Moore 1981)

Rozdelíme všetky prvky do dvojíc (a jedného navyše pre nepárne n).
Dvojice v ktorých sú prvky rôzne zahodíme.
Teraz nutne máme viac „dobrých“ ako „zlých“ dvojíc.
Z každej si necháme jeden prvok, majorita ostáva zachovaná.

Úloha o majoritnom prvku a jej lepšie riešenia

Nevýhody hľadania mediánu

Algoritmy vo worst-case $\Theta(n)$ sú komplikované.
Potreba pamäte $\Theta(n)$ – nedá sa zlepšiť?

Nové pozorovanie

Keď nájdem dva rôzne prvky, môžem *oba* zahodiť.
(Zahodím ≥ 1 zlý, ≤ 1 dobrý, majorita sa zachová.)

Do dvojíc! (Boyer, Moore 1981)

Rozdelíme všetky prvky do dvojíc (a jedného navyše pre nepárne n).
Dvojice v ktorých sú prvky rôzne zahodíme.

Teraz nutne máme viac „dobrých“ ako „zlých“ dvojíc.
Z každej si necháme jeden prvok, majorita ostáva zachovaná.

Úloha o majoritnom prvku a jej lepšie riešenia

Nevýhody hľadania mediánu

Algoritmy vo worst-case $\Theta(n)$ sú komplikované.
Potreba pamäte $\Theta(n)$ – nedá sa zlepšiť?

Nové pozorovanie

Keď nájdem dva rôzne prvky, môžem *oba* zahodiť.
(Zahodím ≥ 1 zlý, ≤ 1 dobrý, majorita sa zachová.)

Do dvojíc! (Boyer, Moore 1981)

Rozdelíme všetky prvky do dvojíc (a jedného navyše pre nepárne n).
Dvojice v ktorých sú prvky rôzne zahodíme.
Teraz nutne máme viac „dobrých“ ako „zlých“ dvojíc.
Z každej si necháme jeden prvok, majorita ostáva zachovaná.

Úloha o majoritnom prvku a jej lepšie riešenia

Turnaj dvojíc pokračuje

Postup opakujeme až kým neskončíme s jediným prvkom.

Worst case: Zakaždým zmenšíme počet prvkov $\sim$ na polovicu.

$$n + n/2 + n/4 + \dots = 2n$$

Časová aj pamäťová zložitosť $\Theta(n)$ ale ľahká implementácia

Optimálne riešenie

Vylepšíme predchádzajúce riešenie tak, aby stačila pamäť $O(1)$:

```
kandidat, pocet = None, 0
for p in postupnost_prvkov:
    if pocet == 0:
        kandidat, pocet = p, 1
    else:
        if p == kandidat: pocet += 1
        else: pocet -= 1
```

Úloha o majoritnom prvku a jej lepšie riešenia

Turnaj dvojíc pokračuje

Postup opakujeme až kým neskončíme s jediným prvkom.

Worst case: Zakaždým zmenšíme počet prvkov $\sim$ na polovicu.

$$n + n/2 + n/4 + \dots = 2n$$

Časová aj pamäťová zložitosť $\Theta(n)$ ale ľahká implementácia

Optimálne riešenie

Vylepšíme predchádzajúce riešenie tak, aby stačila pamäť $O(1)$:

```
kandidat, pocet = None, 0
for p in postupnost_prvkov:
    if pocet == 0:
        kandidat, pocet = p, 1
    else:
        if p == kandidat: pocet += 1
        else: pocet -= 1
```

Všeobecnejšia verzia

Viac častých prvkov?

Na vstupe je postupnosť n prvkov a číslo k .

Nájdite všetky prvky, ktoré sa vyskytujú viac ako (n/k) -krát.

Veľmi podobné riešenie

Udržiavame množinu $k - 1$ kandidátov.

Na konci sú v nej zaručene všetky časté prvky.

(Možno aj nejaké iné, môže byť treba dodatočnú kontrolu.)

Všeobecnejšia verzia

Viac častých prvkov?

Na vstupe je postupnosť n prvkov a číslo k .

Nájdite všetky prvky, ktoré sa vyskytujú viac ako (n/k) -krát.

Veľmi podobné riešenie

Udržiavame množinu $k - 1$ kandidátov.

Na konci sú v nej zaručene všetky časté prvky.

(Možno aj nejaké iné, môže byť treba dodatočnú kontrolu.)

Praktické aplikácie a súvisiace témy

Načo je to dobré?

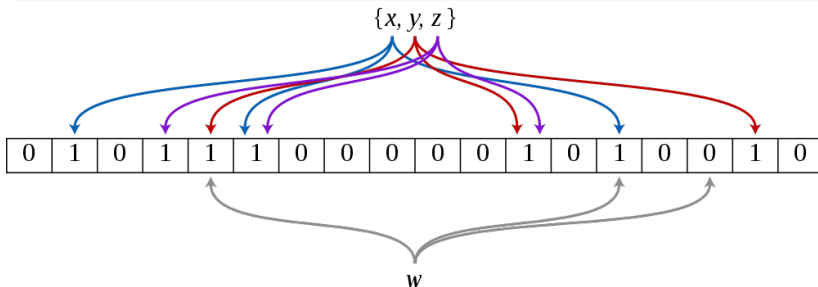
Lacná analýza veľkého prúdu dát:

- ktoré stránky sú najnavštevovanejšie?
- nerobí niekto DoS na môj router?

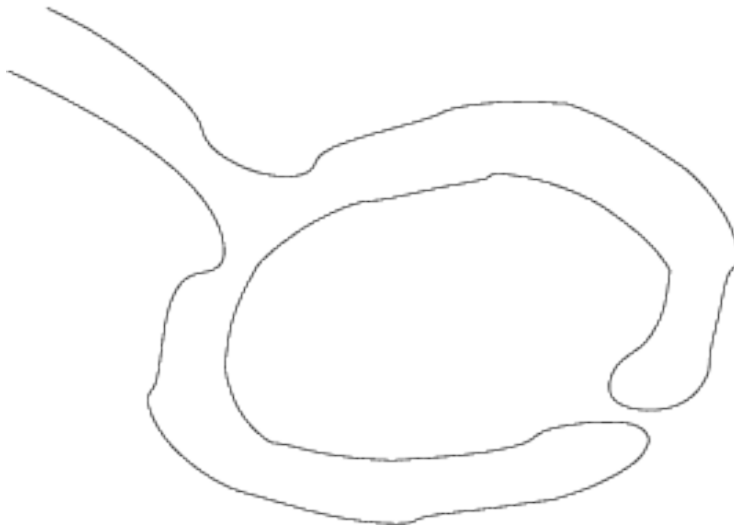
(to celé v konštantnej pomocnej pamäti)

Čo s viac pamäťou? Bloomov filter

„Hešovacia tabuľka“, no nepamätáme si ani len samotné prvky.
Napriek tomu vieme skoro s istotou vkladať a testovať.
Stačí konštantný počet bitov na záznam.
(použitý napr. v Google BigTable)



Alibaba a kúzelná jaskyňa



Stále zľahka

Máte pred sebou zadanie ťažkého sudoku.
Na papieri si ho vyriešite. Váš kamarát ho vyriešiť nevie.
Ako mu dokázať, že poznáte riešenie?
... ale neprezradiť nič z toho riešenia?

Challenge-response protokol

Rozložím na zadanie 81 lístočkov s číslami (face down).
Kamarát má na výber:
a) skontroluje, či tie dané v zadaní sú OK
b) vyberie blok, zamiešame lístky z neho, skontroluje 1-9

Stále zľahka

Máte pred sebou zadanie ťažkého sudoku.
Na papieri si ho vyriešite. Váš kamarát ho vyriešiť nevie.
Ako mu dokázať, že poznáte riešenie?
... ale neprezradiť nič z toho riešenia?

Challenge-response protokol

Rozložím na zadanie 81 lístočkov s číslami (face down).
Kamarát má na výber:

- a) skontroluje, či tie dané v zadaní sú OK
- b) vyberie blok, zamiešame lístky z neho, skontroluje 1-9

Serióznejší príklad

Hamiltonovská kružnica

Daný je graf G . Alenka v ňom pozná Hamiltonovskú kružnicu. Ako toto dokáže Borisovi?

Challenge-response protokol

Alenka vyrobí graf H izomorfný s G (spermutuje vrcholy). Každú hranu H napíše na iný papierik, rozloží ich (face down). Boris má na výber:

- a) chce vidieť kružnicu: Alenka odkryje n lístočkov
- b) chce vidieť všetko: Alenka odkryje všetko a povie izomorfizmus.

Serióznejší príklad

Hamiltonovská kružnica

Daný je graf G . Alenka v ňom pozná Hamiltonovskú kružnicu. Ako toto dokáže Borisovi?

Challenge-response protokol

Alenka vyrobí graf H izomorfný s G (spermutuje vrcholy). Každú hranu H napíše na iný papierik, rozloží ich (face down). Boris má na výber:

- a) chce vidieť kružnicu: Alenka odkryje n lístočkov
- b) chce vidieť všetko: Alenka odkryje všetko a povie izomorfizmus.

Seriózne zero knowledge schémy

Bit commitment

Pri implementácii potrebujeme ekvivalent „papierika textom dole“.

Použitie

Autentifikácia!

1. ja poznám heslo
2. chcem o tom presvedčiť server
3. nik kto počúva sa nesmie nič dozvedieť

Seriózne zero knowledge schémy

Bit commitment

Pri implementácii potrebujeme ekvivalent „papierika textom dole“.

Použitie

Autentifikácia!

1. ja poznám heslo
2. chcem o tom presvedčiť server
3. nik kto počúva sa nesmie nič dozvedieť

Koniec

Ďakujem za pozornosť!