Koľko študentov sa zmestí do tridsaťdeviny?

Michal „mišof“ Forišek

Katedra informatiky
Univerzita Komenského
Bratislava, Slovensko

Akadémia Trojstenu, december 2007

Indukcia zlyháva

Základný problém kompresie

Čo máme

disk a na disku veľký súbor

Čo chceme

disk a na disku menší súbor

Sprosté riešenie

Riešenie

umažeme polovicu súboru

Problém

prišli sme o (väčšinou asi dôležitú) polovicu súboru

Čo presnejšie chceme

disk a na disku menší súbor, **z ktorého vieme späť zostrojiť pôvodný súbor**

Sprosté riešenie

Riešenie

umažeme polovicu súboru

Problém

prišli sme o (väčšinou asi dôležitú) polovicu súboru

Čo presnejšie chceme

disk a na disku menší súbor, **z ktorého vieme späť zostrojiť pôvodný súbor**

Sprosté riešenie

Riešenie

umažeme polovicu súboru

Problém

prišli sme o (väčšinou asi dôležitú) polovicu súboru

Čo **presnejšie** chceme

disk a na disku menší súbor, **z ktorého vieme späť zostrojiť pôvodný súbor**

Skromný cieľ

Kompresný program

Program, ktorý vie vyrábať z väčších súborov menšie, a z tých menších potom späť zostrojiť väčšie.

Aký dobrý program chceme?

Na úvod skromný cieľ: vedieť ľubovoľný 1 MB súbor zmenšiť aspoň o 10 B.

Problém 2

Súborov s veľkosťou 1 MB je

1 204 203 453 131 759 529 492 479-krát **viac**

ako súborov s veľkosťou o 10 B menšou (a menších).

Skromný cieľ

Kompresný program

Program, ktorý vie vyrábať z väčších súborov menšie, a z tých menších potom späť zostrojiť väčšie.

Aký dobrý program chceme?

Na úvod skromný cieľ: vedieť ľubovoľný 1 MB súbor zmenšiť aspoň o 10 B.

Problém 2

Súborov s veľkosťou 1 MB je

1 204 203 453 131 759 529 492 479-krát **viac**

ako súborov s veľkosťou o 10 B menšou (a menších).

Skromný cieľ

Kompresný program

Program, ktorý vie vyrábať z väčších súborov menšie, a z tých menších potom späť zostrojiť väčšie.

Aký dobrý program chceme?

Na úvod skromný cieľ: vedieť ľubovoľný 1 MB súbor zmenšiť aspoň o 10 B.

Problém 2

Súborov s veľkosťou 1 MB je

1 204 203 453 131 759 529 492 479-krát **viac**

ako súborov s veľkosťou o 10 B menšou (a menších).

Záver?

Dôsledok

99.999 999 999 999 999 999 999 917% súborov nevieme zmenšiť ani len o 10 bajtov.
Kompresia nemôže fungovať.

Prax je sviňa

V praxi to ale funguje. Kde je pes zakopaný?

Záver?

Dôsledok

99.999 999 999 999 999 999 999 917% súborov nevieme zmenšiť ani len o 10 bajtov.
Kompresia nemôže fungovať.

Prax je sviňa

V praxi to ale funguje. Kde je pes zakopaný?

Akčná vsuvka

Potrebujem niekoľko (skupín) dobrovoľníkov vybavených kockami, mincami, ...

Úloha

- Zobrať list papiera (A5).
- Napísať naň hore jednoslovné „heslo“, podľa ktorého si ho spoznáte.
- Vygenerovať čo najrýchlejšie náhodnú postupnosť tak 200 núl a jednotiek.

znak=0, hlava=1

Akčnejšia vsuvka

Potrebujem ešte niekoľko dobrovoľníkov

Úloha

- Zobrať list papiera (A5).
- Napísať naň hore jednoslovné „heslo“,
  podľa ktorého si ho spoznáte.
- Čo najrýchlejšie naň napísať
  náhodnú postupnosť tak 200 núl a jednotiek.

Vyhodnotenie

Závery

Dokonca aj v situáciách, kedy to nevidíme, dáta, s ktorými robíme, nie sú náhodné.
Pravidelnosti v dátach budeme môcť využiť.

Vyhodnotenie

Závery

Dokonca aj v situáciách, kedy to nevidíme, dáta, s ktorými robíme, nie sú náhodné.
Pravidelnosti v dátach budeme môcť využiť.

Tá istá informácia sa dá uložiť rôzne

Menej dobre

„0101010101010101010101010101010101010101010101010101“

Lepšie

„25-krát 01“, prípadne „25[01]"

Menej dobre

„0110101000101000101000100000101000001000101000...00“

Lepšie

„reťazec dĺžky 100 000 s jednotkami na prvočíselných pozíciách“

Tá istá informácia sa dá uložiť rôzne

Menej dobre

„0101010101010101010101010101010101010101010101010101“

Lepšie

„25-krát 01“, prípadne „25[01]"

Menej dobre

„0110101000101000101000100000101000001000101000...00“

Lepšie

„reťazec dĺžky 100 000 s jednotkami na prvočíselných pozíciách“

Prečo kompresia funguje?

- Informácie kódujeme neefektívne.
- Kompresia = nájdenie efektívneho zápisu.
- Niekde je hranica – nedá sa všetko zakódovať 1 bitom.
- Tou hranicou je množstvo informácie v súbore.

Prečo kompresia funguje?

- Informácie kódujeme neefektívne.
- Kompresia = nájdenie efektívneho zápisu.
- Niekde je hranica – nedá sa všetko zakódovať 1 bitom.
- Tou hranicou je množstvo informácie v súbore.

Ako merať množstvo informácií?

Hra: Uhádni slovenskú vetu:

Veta

?????????????? (14 znakov)

Môžete sa pýtať áno/nie otázky.
Otázniky môžu byť veľké písmená a medzery.

Ako merať množstvo informácií?

Pýtať sa dalo rôznymi spôsobmi – lepšími aj horšími.

Množstvo informácie

= počet otázok, ktoré potrebujeme, ak sa pýtame „najlepšie“.

Každá odpoveď áno/nie = 1 bit informácie.

Čo je „najlepšie“?

Intuitívne: pýtať sa tak, aby pravdepodobnosť oboch odpovedí bola zhruba 1/2.

Ako merať množstvo informácií?

Pýtať sa dalo rôznymi spôsobmi – lepšími aj horšími.

Množstvo informácie

= počet otázok, ktoré potrebujeme, ak sa pýtame „najlepšie“.

Každá odpoveď áno/nie = 1 bit informácie.

Čo je „najlepšie“?

Intuitívne: pýtať sa tak, aby pravdepodobnosť oboch odpovedí bola zhruba 1/2.

Ako merať množstvo informácií?

Pýtať sa dalo rôznymi spôsobmi – lepšími aj horšími.

Množstvo informácie

= počet otázok, ktoré potrebujeme, ak sa pýtame „najlepšie“.

Každá odpoveď áno/nie = 1 bit informácie.

Čo je „najlepšie“?

Intuitívne: pýtať sa tak, aby pravdepodobnosť oboch odpovedí bola zhruba 1/2.

Ako merať množstvo informácií?

Čo ale keď vám poviem nejakú vetu? Koľko informácie obsahuje?

Iný pohľad na to isté

Informáciu dostávame, keď vidíme, že nastal nejaký jav.
Množstvo informácie závisí od jeho pravdepodobnosti.

Príklad

Žrebovanie náhodného dňa.

Poriadnejšia definícia množstva informácie

$I(p)$: Koľko informácie nám dá to, že nastane jav čo mal pravdepodobnosť p?

- I je spojitá a klesajúca, $I(1) = 0$, pre $p \to 0$ je $I(p) \to \infty$.
- Dva nezávislé javy: $I(pq) = I(p) + I(q)$.
- $I(1/2) = 1$ (hod mincou dá jeden bit informácie)

Riešenie

$$I(p) = \log_2(1/p) = -\log_2 p$$

Poriadnejšia definícia množstva informácie

$I(p)$: Koľko informácie nám dá to, že nastane jav čo mal pravdepodobnosť p?

- I je spojitá a klesajúca, $I(1) = 0$, pre $p \to 0$ je $I(p) \to \infty$.
- Dva nezávislé javy: $I(pq) = I(p) + I(q)$.
- $I(1/2) = 1$ (hod mincou dá jeden bit informácie)

Riešenie

$$I(p) = \log_2(1/p) = -\log_2 p$$

Poriadnejšia definícia množstva informácie

$I(p)$: Koľko informácie nám dá to, že nastane jav čo mal pravdepodobnosť p?

- I je spojitá a klesajúca, $I(1) = 0$, pre $p \to 0$ je $I(p) \to \infty$.
- Dva nezávislé javy: $I(pq) = I(p) + I(q)$.
- $I(1/2) = 1$ (hod mincou dá jeden bit informácie)

Riešenie

$$I(p) = \log_2(1/p) = -\log_2 p$$

Poriadnejšia definícia množstva informácie

$I(p)$: Koľko informácie nám dá to, že nastane jav čo mal pravdepodobnosť p?

- I je spojitá a klesajúca, $I(1) = 0$, pre $p \to 0$ je $I(p) \to \infty$.
- Dva nezávislé javy: $I(pq) = I(p) + I(q)$.
- $I(1/2) = 1$ (hod mincou dá jeden bit informácie)

Riešenie

$$I(p) = \log_2(1/p) = -\log_2 p$$

Poriadnejšia definícia množstva informácie

$I(p)$: Koľko informácie nám dá to, že nastane jav čo mal pravdepodobnosť p?

- I je spojitá a klesajúca, $I(1) = 0$, pre $p \to 0$ je $I(p) \to \infty$.
- Dva nezávislé javy: $I(pq) = I(p) + I(q)$.
- $I(1/2) = 1$ (hod mincou dá jeden bit informácie)

Riešenie

$$I(p) = \log_2(1/p) = -\log_2 p$$

Späť k pýtaniu otázok

Opýtam sa áno/nie otázku, kde odpoveď áno má pravdep. p.
Koľko informácie dostanem?

- s pravdep. p dostanem: $-\log_2 p$ bitov
- s pravdep. $1-p$ dostanem: $-\log_2(1-p)$ bitov
- v priemere dostanem: $-p\log_2 p-(1-p)\log_2(1-p)$ bitov

Späť k pýtaniu otázok

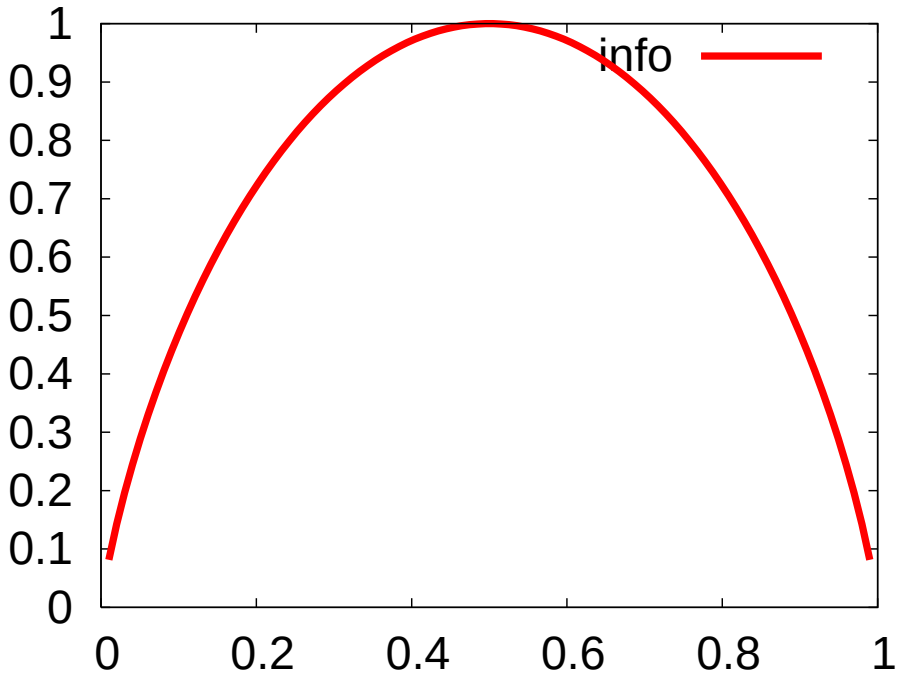
Opýtam sa áno/nie otázku, kde odpoveď áno má pravdep. p.
Koľko informácie dostanem?

- s pravdep. p dostanem: $-\log_2 p$ bitov
- s pravdep. $1-p$ dostanem: $-\log_2(1-p)$ bitov
- v priemere dostanem: $-p\log_2 p-(1-p)\log_2(1-p)$ bitov

Späť k pýtaniu otázok

Opýtam sa áno/nie otázku, kde odpoveď áno má pravdep. p.
Koľko informácie dostanem?

- s pravdep. p dostanem: $-\log_2 p$ bitov
- s pravdep. $1-p$ dostanem: $-\log_2(1-p)$ bitov
- v priemere dostanem: $-p\log_2 p-(1-p)\log_2(1-p)$ bitov



Čo to znamená pre kompresiu?

Kompresia = hra „uhádni súbor na čo najmenej otázok“.
(=„popíš súbor čo najmenej bitmi“)

Správna „stratégia pýtania sa“ závisí od pravdepodobností jednotlivých možností – ak vieme niečo o dátach, ktoré ideme komprimovať, môžeme to využiť.

Existuje jeden „najlepší kompresný algoritmus“?

. . . alebo závisí na tom, ako robíme dekompresiu?

Záver: Kolmogorovská zložitosť

Zložitosť súboru = veľkosť najmenšieho programu, ktorý ho vyrobí.

Existuje jeden „najlepší kompresný algoritmus“?

. . . alebo závisí na tom, ako robíme dekompresiu?

Samorozbaľovacie archívy:

- Zoberieme komprimovaný súbor S.
- Prilepíme k nemu na začiatok kód.
- Máme program P, ktorý po spustení vyrobí pôvodný súbor.
- P má od S len o konštantný počet znakov viac.

Záver: Kolmogorovská zložitosť

Zložitosť súboru = veľkosť najmenšieho programu, ktorý ho vyrobí.

Existuje jeden „najlepší kompresný algoritmus“?

... alebo závisí na tom, ako robíme dekompresiu?

Samorozbaľovacie archívy:

- Zoberieme komprimovaný súbor S.
- Prilepíme k nemu na začiatok kód.
- Máme program P, ktorý po spustení vyrobí pôvodný súbor.
- P má od S len o konštantný počet znakov viac.

Záver: Kolmogorovská zložitosť

Zložitosť súboru = veľkosť najmenšieho programu, ktorý ho vyrobí.

Prax

Všeobecne o praktickej kompresii

Problém

Optimálna kompresia je výpočtovo veľmi náročná.

Obmedzenie z praxe

Časová zložitosť nie moc horšia ako lineárna.

Dôsledok

Kompresný algoritmus si musí „vybrať“ vhodný „druh závislostí“, ktoré hľadá a efektívnejšie ukladá.

Čo sú binárne kódy?

Čo ideme využiť?

To, že sa rôzne znaky vyskytujú v súbore rôzne často.

Binárne kódy

- postupnosť bitov = *kódové slovo*
- každému znaku priradíme *kódové slovo*
- príklad: ASCII kód (všetko má dĺžku 8)
- vieme lepšie: častejším znakom dať kratšie kódy

Aký binárny kód chceme?

Musí sa dať dekódovať

$$a \to 0, \qquad b \to 11, \qquad c \to 011$$

Musí sa dať dekódovať efektívne

$$a \to 1, \qquad b \to 10, \qquad c \to 01$$

Postupnosť 1010101010 je *bbbbb*, ale 10101010101 je *accccc*.

A ešte...

... by aj mal komprimovať.

Aký binárny kód chceme?

Musí sa dať dekódovať

$$a \longrightarrow 0, \qquad b \longrightarrow 11, \qquad c \longrightarrow 011$$

Musí sa dať dekódovať efektívne

$$a \longrightarrow 1, \qquad b \longrightarrow 10, \qquad c \longrightarrow 01$$

Postupnosť 1010101010 je *bbbbb*, ale 10101010101 je *accccc*.

A ešte...

... by aj mal komprimovať.

Aký binárny kód chceme?

Musí sa dať dekódovať

$$a \longrightarrow 0, \qquad b \longrightarrow 11, \qquad c \longrightarrow 011$$

Musí sa dať dekódovať efektívne

$$a \longrightarrow 1, \qquad b \longrightarrow 10, \qquad c \longrightarrow 01$$

Postupnosť 1010101010 je *bbbbb*, ale 10101010101 je *accccc*.

A ešte...

... by aj mal komprimovať.

Ako na to?

Hlavná idea

Budeme postupne "hádať" znaky a zapisovať si odpovede.

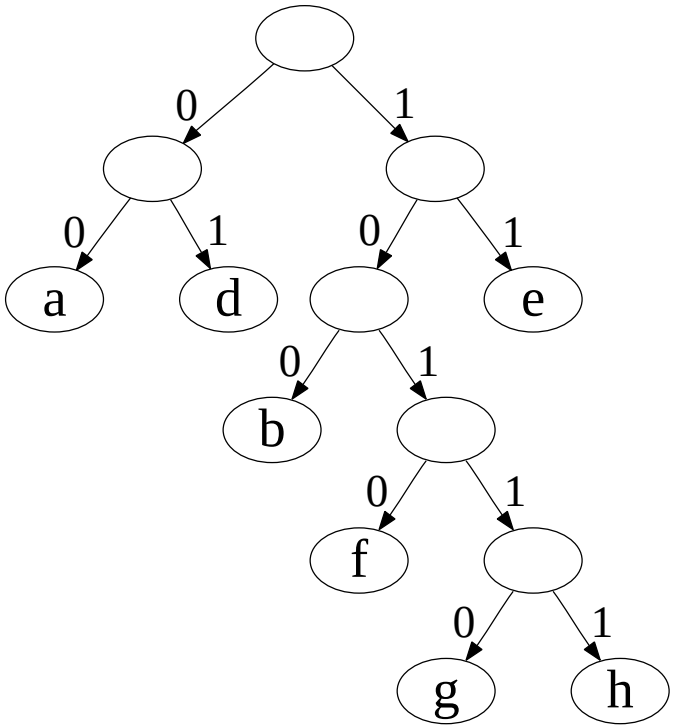
Príklad

a	b	c	d
35%	12%	4%	17%
e	f	g	h
25%	5%	1%	1%

Ako na to?

Hlavná idea

Budeme postupne "hádať" znaky a zapisovať si odpovede.



Príklad

a	b	c	d
35%	12%	4%	17%
e	f	g	h
25%	5%	1%	1%

Strom určuje kódy: $b \rightarrow 100$

Cena nášho kódu: 2.2 bitu/znak.

Ako zostrojiť najlepší strom?

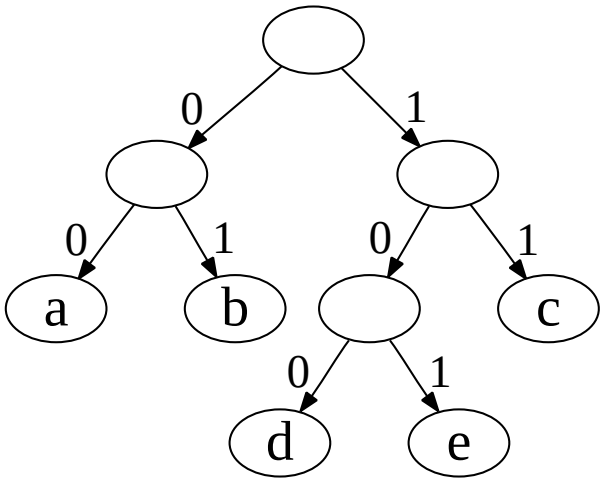
Fanov kód

Vždy čo najrovnomernejšie rozdeliť znaky „na polovicu“.

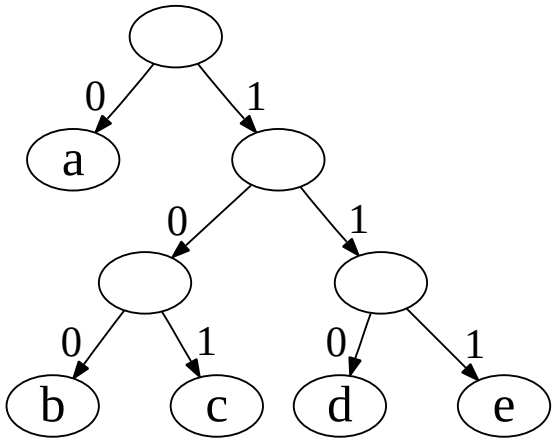
Fanov kód nemusí byť optimálny

Príklad: 40% a, po 15% b, c, d, e.

Fano: 2.45 bitu/znak



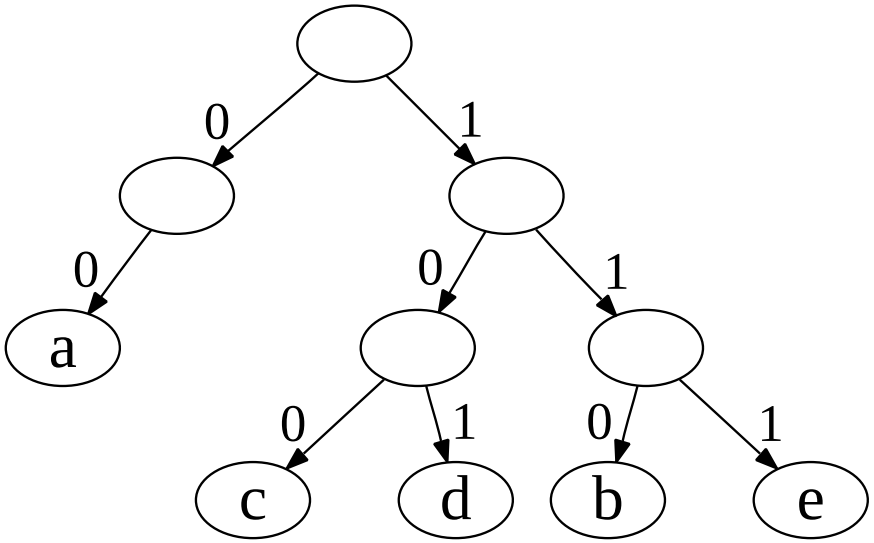
Optimálne: 2.20 bitu/znak



Optimálny Huffmanov kód

Pozorovania o optimálnom strome

- čím častejšie písmeno, tým kratší kód $\Rightarrow$ šmejdy sú dole
- vnútorné vrcholy majú stupeň 2
- na spodnej úrovni sú 2 susedné vrcholy
  $\Rightarrow$ prehádžeme znaky tak, aby 2 najväčšie šmejdy susedili



Praktické výsledky

Slovenský text bez diakritiky

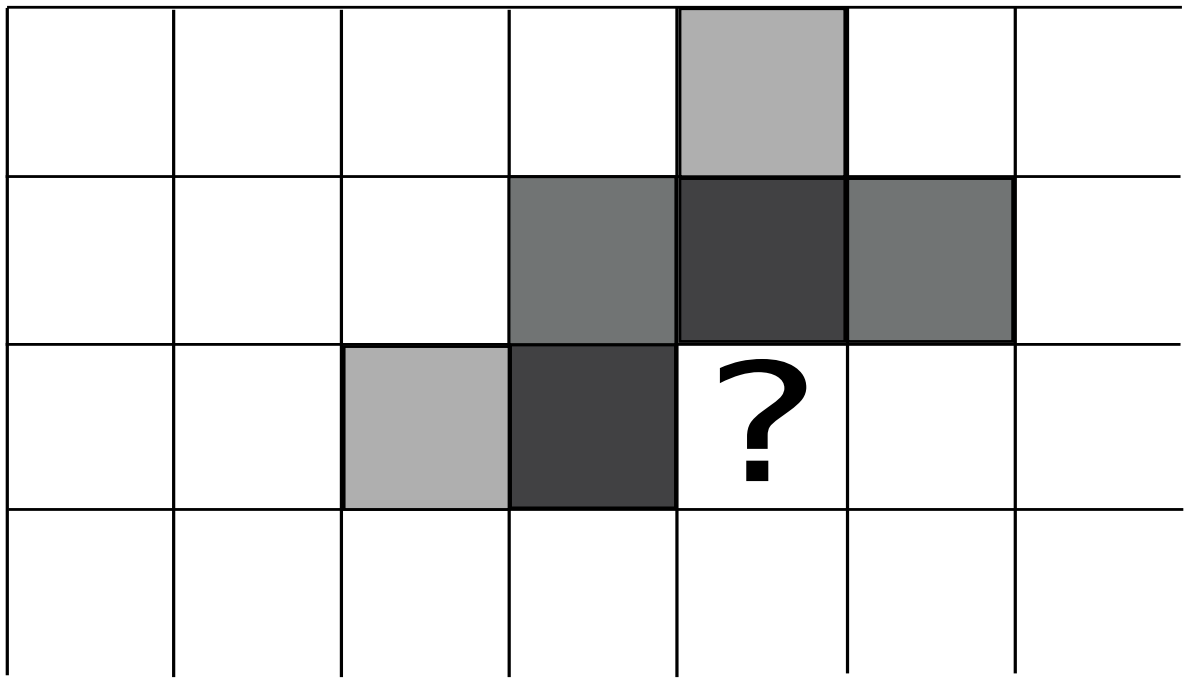
- nekomprimovaný: 4.75 bitu/znak
- Huffmanov kód: 4.26 bitu/znak
- Huffmanov kód podľa predch. trigramu: 2.50 bitu/znak

Odhad je náš kamarát

Pozorovanie

Čo vieme odhadnúť, to vieme aj komprimovať.

Odhad pixelu obrázka:



Jednoduchý algoritmus

Začiatok algoritmu

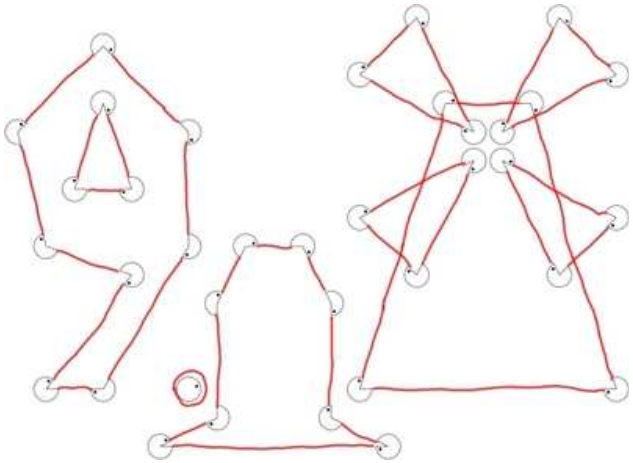
1. Ideme po riadkoch.
2. Každý pixel skúsime odhadnúť pomocou predchádzajúcich.
3. Zapisujeme si veľkosti chyby.

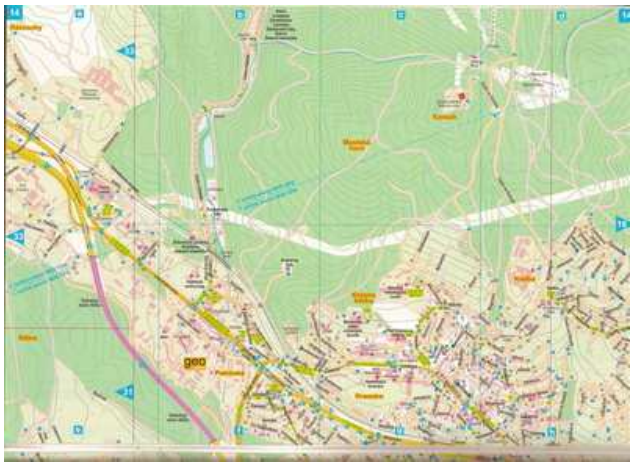
Pozorovania

- Z týchto údajov vieme späť zostrojiť obrázok.
- Ak dobre hádame, budú zapísané hodnoty zväčša okolo 0.

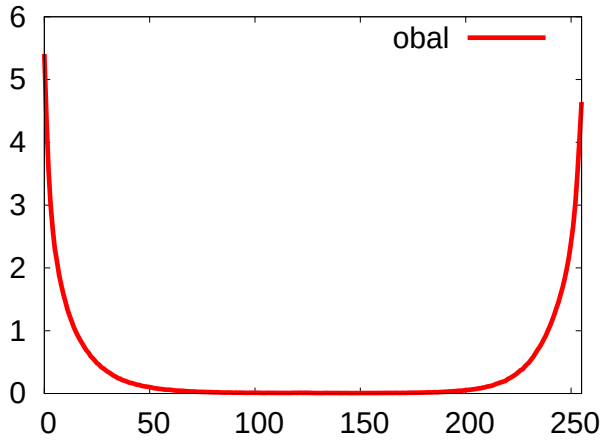
Praktické testy – testovacia sada

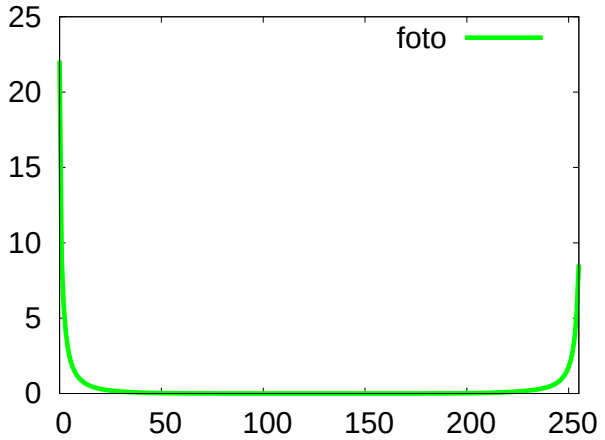
Scan0002.tif

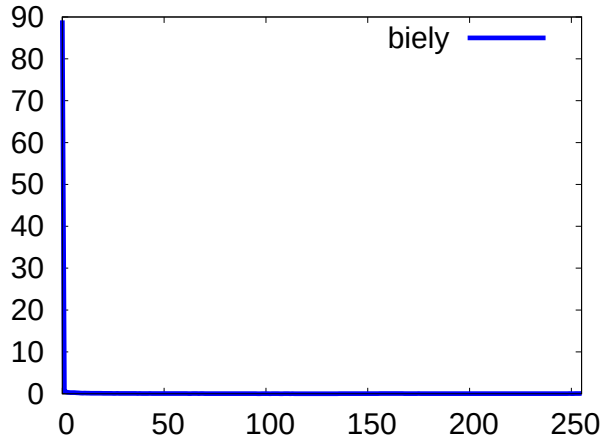


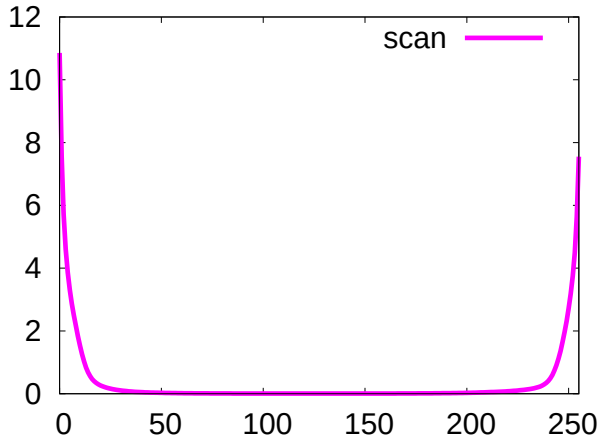


Praktické testy – úspešnosť predpovedania









Jednoduchý algoritmus II

Algoritmus

1. Ideme po riadkoch.
2. Každý pixel skúsime odhadnúť pomocou predchádzajúcich.
3. Zapisujeme si veľkosti chyby.
4. Výsledné údaje skomprimujeme Huffmanovym kódom.

Praktické testy – výsledky kompresie

	obal	foto	biely	scan
BMP	2 746 518	14 745k	892 854	106 855k
PNG	2 120 351	7 754k	71 765	60 903k
ZIP	2 379 730	10 981k	59 633	84 921k
RAR	2 294 352	8 539k	51 457	65 739k
naše	2 120 142	8 898k	202 394	69 933k
TIF	2 761 924	5 751k	70 086	39 779k
JPG	920 558	2 464k	88 238	16 603k

Úvod do stratovej kompresie

Čo si môžeme dovoliť?

Môžeme zmeniť objekt tak, aby to človek (príliš) nespoznal.

Ako sa to robí v praxi?

Magické zaklínadlo: Fourierova transformácia.

Vieme spraviť niečo jednoduché a účinné?

V našom algoritme si môžeme zapísať nie skutočnú chybu, ale „± 1“, ako sa nám hodí.
Inými slovami, môžeme vhodne nastaviť najmenej významný bit (alebo 2) z obrázka a nik si nič nevšimne.

Praktické testy – výsledky kompresie II

	obal	foto	biely	scan
BMP	2 746 518	14 745k	892 854	106 855k
PNG	2 120 351	7 754k	71 765	60 903k
ZIP	2 379 730	10 981k	59 633	84 921k
RAR	2 294 352	8 539k	51 457	65 739k
naše	2 120 142	8 898k	202 394	69 933k
TIF	2 761 924	5 751k	70 086	39 779k
JPG	920 558	2 464k	88 238	16 603k
1bit	1 779 299	7 153k	188 287	56 919k
2bit	1 443 371	5 566k	174 060	43 742k

Nie je to až tak nanič

Pri 2bit ostalo zo vstupu 75% nedotknutých,
na scan+foto komprimovaná veľkosť klesla na 62%.

Zazvonil zvonec, rozprávke je koniec.