

1. INFORMACE, DATA, INFORMATIKA

1.1. DATA

Data (údaje) jsou vjemy, které dokážeme zachytit svými smysly.

1.2. INFORMACE

Informace jsou data, kterým rozumíme, mají pro nás nějaký smysl.

Informace můžeme kvantifikovat (měřit) teprve když je převedeme do podoby čísel. Veškeré stroje pracují s informacemi právě v podobě čísel převedených do dvojkové číselné soustavy. Informace, které zpracovává počítač jsou (prozatím) několika druhů:

- textové informace (texty, programy...)
- obrazové informace (fotografie, grafiky, video, animace...)
- zvukové informace (zvuky, hudba...)

U informací posuzujeme jejich:

- aktuálnost -
- relevanci - adekvátnost potřebě. Informace je relevantní, je-li v ní obsaženo to, co potřebujeme vědět a nic navíc.
- pravdivost - soulad se skutečností
- impaktní faktor ????
-

1.3. INFORMATIKA

Informatika je věda zabývající se uchováním, zpracováním a využíváním údajů a informací.

Informatika se vydělila z matematiky a jako prostředek využívá výpočetní techniku.

Odvětví informatiky:

- Výpočetní technika - zkoumá technické vybavení
- Algoritmizace - navrhování postupů k řešení problémů
- Programování - převádění algoritmů do programovacího jazyka
- Softwarové inženýrství - nauka o vývoji aplikací
- Počítačová grafika - nauka o tvorbě a zpracování 2D i 3D obrazů
- Počítačové modelování a simulace - aplikace matematických modelů na reálné situace
- Formální logika, teorie automatů a formálních jazyků - matematické modely strojů a formálních jazyků pro zápis algoritmů a programů
- Kybernetika a robotika - vývoj strojů schopných samostatné činnosti
- Umělá inteligence - zkoumání procesů lidského myšlení, jejich matematický popis a aplikace při vývoji strojů

Úkoly:

1. Uveďte příklad dat, která nejsou pro vás informací.
2. Jaké druhy informací nejsou dosud běžně elektronicky zpracovatelné?
3. Najděte pomocí webových vyhledávačů vysvětlení, proč kovové předměty v dlaní studí víc než třeba dřevěné, přestože mají stejnou teplotu. U nalezených odkazů posuďte jejich aktuálnost, relevanci a pravdivost. Vyzkoušejte nejméně dva různé [vyhledávače](#).

2. ZDROJE INFORMACÍ

2.1. HISTORIE

Odedávna se informace šířily v ústní podobě, což je způsob velice nespolehlivý. Spolehlivější zdroj informací z pravěku a starověku jsou **nástěnné malby**, **obrazy vytesané do kamene**, nápisy na **stěnách hrobek** či papyrových svitcích.

K systematickému shromažďování informací docházelo na různých místech světa v **knihovnách**. Nejznámější je Alexandrijská knihovna ze 3. století př. n. l., která přetrvala staletí a obsahovala statisíce papyrových svitků.

Ve středověku bylo jediným způsobem uchování a přenosu informací psaní knih. Samozřejmě ručně, takže vlastnictví knih bylo velkým přepychem. Revolucí v pradávě informatice byl v roce 1455 **vynález knihtisku**. Výroba knih se výrazně zrychlila a zlevnila a tím se snáz šířilo vzdělání. Dalším převratným vynálezem byl **psací stroj** (1829). Rozvržení znaků na tehdejší klávesnici se používá dodnes.

Významným objevem podstatně zlepšujícím přenos informací byl **telegraf** (1844). Zprávy se přenášely zakódované do **Morseovy abecedy**. Následoval **telefon** (1848) jež přenášel **hlas**.

Přenos a uchování obrazu by nebylo možné bez objevu **fotografie** (1839). Následoval vynález **filmu**, **fonografu** (zařízení pro záznam a přehrávání zvuku), **gramofonu** a **magnetofonu**. Posledním převratným objevem v oblasti záznamu a uchování obrazu byl **digitální fotoaparát** (1981).

2.2. SOUČASNOST

Mimovolně získáváme nesystematické informace neustále (z knih, televize, novin, z vyprávění druhých, vlastním pozorováním...) Chceme-li ale získat informace systematické, zasazené do souvislostí a utříděné, potřebujeme za tím účelem zřízení informační zdroje:

Encyklopedie

tištěné či elektronické. Obsahují definice a stručná vysvětlení základních pojmů. Běžné jsou odkazy na jiné části textu - **hypertext**. Elektronické encyklopedie jsou **multimediální** - obsahují informace textové, obrazové i zvukové.

Knihy

Jsou ucelené a obsažné publikace, nezastupitelné v případě, že chceme danou tematiku prostudovat skutečně do hloubky a ve všech souvislostech. Nehodí se pro získávání informací souhrnných (různé přehledy nebo seznamy čehokoliv), nebo jen dílčích (kdo co napsal, kolik co stojí...). Knihy jsou shromažďovány v **knihovnách**, v nichž usnadňuje vyhledávání (oproti nedávné minulosti) **vyhledávací systém**.

Internet

Je z hlediska informatiky obrovským zdrojem informací (ale i dezinformací). Pomocí internetových vyhledávačů máme ve vteřině k dispozici informace, které na dané téma napsal kdokoli kdekoli na celém světě. Jsou to ovšem informace bez záruky pravdivosti. Jejich pravdivost si musí uživatel posoudit sám na základě svých znalostí a porovnání s jinými zdroji.

Úkoly:

1. Popište historicky významné zdroje informací, objevy, které vedly k zefektivnění přenosu informací mezi lidmi.
2. Jaké informační zdroje máte v okolí svého bydliště? jaký využívá vyhledávací systém?
3. Popište v současnosti nejvyužívanější zdroje informací, jejich výhody a nevýhody.
4. Popište v současnosti nejvyužívanější způsob předávání informací mezi lidmi. Porovnejte s minulým stoletím

3. ČÍSELNÉ SOUSTAVY

Číselná soustava je způsob reprezentace čísel.

Například:

počet dní v roce lze zapsat mnoha způsoby: **365** nebo **CCCLXV** nebo **101101101** nebo **555** či **16D**.

Desítková číselná soustava

365 - je zápis čísla vyjadřujícího počet dní v roce nám nejbližším způsobem - v **desítkové (dekadické) číselné soustavě**. Tato soustava má k dispozici **10 různých cifer: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9**. (základ soustavy je **10**) Jakékoliv jiné číslo se skládá z těchto cifer a **záleží na pozici cifry** v čísle - určuje její **řád** (to je vlastnost **pozičních číselných soustav**).



Zápis čísel v poziční soustavě lze rozvinout:

Rozvinutý zápis:

$$365 = 3 \cdot 100 + 6 \cdot 10 + 5 \cdot 1 = 3 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$$

řád
základ

Dvojková číselná soustava

101101101 je zápis čísla vyjadřujícího počet dní v roce ve dvojkové (binární) soustavě.

základ soustavy: 2

cifry: 0, 1

rozvinutý zápis:

Rozvinutý zápis:

$$101101101 = 1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Osmičková číselná soustava

555 je zápis čísla vyjadřujícího počet dní v roce v osmičkové (oktanové) číselné soustavě.

základ soustavy: 8

cifry: 0, 1, 2, 3, 4, 5, 6, 7, 8

rozvinutý zápis:

Rozvinutý zápis:

$$555 = 5 \cdot 8^2 + 5 \cdot 8^1 + 5 \cdot 8^0$$

Šestnáctková číselná soustava

16D je zápis čísla vyjadřujícího počet dní v roce v šestnáctkové (hexadecimální) číselné soustavě.

základ soustavy: 16

cifry: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F,

rozvinutý zápis:

Rozvinutý zápis:

$$16D = 1 \cdot 16^2 + 6 \cdot 16^1 + D \cdot 16^0 = 1 \cdot 16^2 + 6 \cdot 16^1 + 13 \cdot 16^0$$

Příklad zápisů čísel v různých pozičních soustavách:

desítková	0	1	2	3	4	5	6	7	8	16	32	64	128
dvojková	0	1	10	11	100	101	110	111	10000	10000	100000	1000000	10000000
osmičková	0	1	2	3	4	5	6	7	10	20	40	100	200
šestnáctková	0	1	2	3	4	5	6	7	8	10	20	40	80

Nepoziční číselné soustavy

Existují číselné soustavy, v nichž pozice cifry v čísle neudává její řád. Takové soustavy se nazývají nepoziční. Příkladem jsou římské číslice.

CCCLXV je zápis čísla vyjadřujícího počet dní v roce v nepoziční číselné soustavě římských čísel.

3.1. PŘEVODY MEZI ČÍSELNÝMI SOUSTAVAMI

Převod čísel z libovolné číselné soustavy do desítkové

Využijeme rozvinutého zápisu v dané soustavě a ten prostě vypočítáme. Například:

$$111010 = 0 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 1 \cdot 2^5 = 0 + 2 + 0 + 8 + 16 + 32 = 58$$

Tedy 111010 ve dvojkové soustavě značí 58 v desítkové soustavě. Zapisujeme: 111010 (2) = 58 (10)

Úkoly:

- Rozviňte zápis čísel v soustavě uvedené v závorce:
 3 659 (Desítková)
 10011 (dvojková)
 1236533 (osmičková)
 215D3A (šestnáctková)
- Jaká je hodnota předchozích čísel v desítkové soustavě? Náповěda: rozvinutý zápis prostě vypočítáme
- Vaše výsledky ověřte na [kalkulačce](#).

Převody z desítkové soustavy do libovolné jiné

Především si uvědomme, jaký význam mají jednotlivé cifry v zápisu čísla v desítkové soustavě: dělíme-li základem (10), je první zbytek po dělení poslední cifrou v ciferném zápisu. Dělíme-li postupně všechny takto vzniklé podíly základem (10), všechny další zbytky po dělení jsou jednotlivé cifry v ciferném zápisu směrem dopředu.

				podíl -				zbytek po dělení -	
				číslo bez poslední cifry				poslední cifra	
3	8	6	:	10	=	3	8	+	6
	3	8	:	10	=	3	+	8	
	3		:	10	=	0	+	3	

Podobně můžeme postupovat při převodu čísel z desítkové soustavy do dvojkové:

Převod čísel z desítkové do dvojkové soustavy

Postupného dělení základem a zapisování zbytků po dělení (pozpátku) využijeme při převodu čísel z desítkové soustavy do dvojkové: **386 (10) = 11000010 (2)**

	podíl	zbytek	
386 : 2 =	193	0	
193 : 2 =	96	1	
96 : 2 =	48	0	
48 : 2 =	24	0	
24 : 2 =	12	0	
12 : 2 =	6	0	
6 : 2 =	3	0	
3 : 2 =	1	1	
1 : 2 =	0	1	

1	1	0	0	0	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---

Převod čísel z desítkové do šestnáctkové soustavy

Stejně tak postupujeme i při převodu z desítkové soustavy do šestnáctkové. Zbytky větší než devět přepíšeme do cifer 16-ové soustavy (10 -> A, 11 -> B, 12 -> C, 13 -> D 14 -> E 15 -> F)

1731 : 16 =	108	3	
108 : 16 =	6	12	→ C
6 : 16 =	0	6	

6	C	3
---	---	---

Úkoly: Převeďte mezi soustavami uvedenými v závorkách:

1. 64 (10) -> (2)
2. 64 (10) -> (7)
3. 3876 (10) -> (16)

3.2. POČÍTÁNÍ VE DVOJKOVÉ SOUSTAVĚ

Základní početní úkony ve dvojkové soustavě můžou být na první pohled nezvyklé, ale po chvíli v nich objevíte stejné zákonitosti, jako při počítání v soustavě desítkové.

Zpaměti je nutné znát ty nejjednodušší počty:

sčítání	odečítání
0 + 0 = 0	0 - 0 = 0
1 + 0 = 1	1 - 0 = 1
1 + 1 = 10	1 - 1 = 0
1 + 10 = 11	10 - 1 = 1
10 + 10 = 100	11 - 10 = 1
10 + 11 = 101	100 - 10 = 10

Sčítání

Sčítání delších binárních čísel provádíme pod sebe, stejně jako u čísel dekadických. Vzpomeňte na druhou třídu základní školy a vyzkoušejte následující příklad:

$$\begin{array}{r}
 1010 \quad (10) \\
 + 10011 \quad (19) \\
 \hline
 11101 \quad (29)
 \end{array}$$


 Přenos 1
 do vyššího řádu

Stejně jako u čísel dekadických při překročení devítky (v případě binárních čísel při překročení jedničky) zvyšujeme o jedničku vyšší řád (přenos jedničky do vyššího řádu). O něco složitější příklad:


$$\begin{array}{r}
 10110 \quad (22) \\
 + 11011 \quad (27) \\
 \hline
 110001 \quad (49)
 \end{array}$$


 Přenos 1
 do vyššího řádu

Odečítání

Analogicky postupujeme i při odečítání binárních čísel:

$$\begin{array}{r}
 1010 \quad (10) \\
 - 1000 \quad (8) \\
 \hline
 0010 \quad (2)
 \end{array}
 \qquad
 \begin{array}{r}
 1000 \quad (8) \\
 - 111 \quad (7) \\
 \hline
 0001 \quad (1)
 \end{array}$$


 Přenos 1
 do vyššího řádu

Násobení

Není nic snazšího, než násobení binárních čísel. Posuďte sami:

$$\begin{array}{r}
 101011 \quad (43) \\
 * \quad 101 \quad (5) \\
 \hline
 101011 \\
 000000 \\
 101011 \\
 \hline
 11010111 \quad (215)
 \end{array}$$

Dělení beze zbytku

Stačí vědět, že: $1:1=1$, $1:0$ nelze, $10:1=10$, $100:10=10\dots$ pak jistě přijdete na to, jak vydělit pod sebou i delší binární čísla. Například:

$$\begin{array}{r}
 \text{(10)} \\
 1\ 0\ 1\ 0 \\
 \underline{0\ 1} \\
 1\ 0
 \end{array}
 :
 \begin{array}{r}
 \text{(2)} \\
 1\ 0
 \end{array}
 =
 \begin{array}{r}
 \text{(5)} \\
 1\ 0\ 1
 \end{array}$$

Úkol: Jak poznáte ve dvojkové soustavě sudé a liché číslo?

Vypočítejte:

$$10110 + 1000110$$

$$101111 - 10110$$

$$1110 + 11111$$

$$1010101 - 101010$$

$$1011 * 10111$$

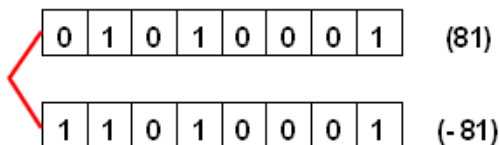
$$11011 : 11$$

3.3. BITOVÁ REPREZENTACE CELÝCH ČÍSEL

Čísla se (stejně jako text, obraz či zvuk) pro zpracování počítačem kódují do dvojkové soustavy a ukládají do 8, 16, 32 nebo 64 bitů (1, 2, 3 nebo 4 Bajtů).

8 bitová reprezentace

Celé číslo je uloženo do 8 bitů, sedm vyjadřuje hodnotu čísla, jeden bit (první) slouží jako znaménko (0 je kladné číslo, 1 záporné) např.:



Je zřejmé, že do 8 bitů nelze uložit nijak vysoká čísla:

maximum 0 1 1 1 1 1 1 1 (127)

minimum 1 1 1 1 1 1 1 1 (-127)

Do osmi bitů lze obecně uložit $2^8 = 256$ různých čísel, ale +0 a -0 jsou různé reprezentace téhož čísla, takže dohromady lze tímto způsobem uložit do osmi bitů (1 Bajtu) pouze 255 různých celých čísel od **-127** do **+127**.

Vícebitové reprezentace

Proto se na reprezentaci celých čísel používají spíše 32 bitové nebo 64 bitové reprezentace.

Do 32 bitů uložíme $2^{32} - 1 = 4\ 294\ 967\ 295$ různých celých čísel.

Do 64 bitů uložíme $2^{64} - 1 = 18\ 446\ 744\ 073\ 709\ 551\ 615$ různých celých čísel.

3.4. BITOVÁ REPREZENTACE REÁLNÝCH ČÍSEL

Množina reálných čísel je sjednocením množiny racionálních čísel a množiny iracionálních čísel. Racionální čísla jsou taková, která lze zapsat konečným desetinným rozvojem. Čísla iracionální nelze zapsat konečným desetinným rozvojem. Čísla jsou v počítači reprezentována konečným počtem bitů, to znamená, že čísla iracionální musíme aproximovat čísly racionálními. Bitová reprezentace reálných čísel je ve skutečnosti reprezentací čísel desetinných, a to pouze s určitou přesností.

Bitová reprezentace desetinných čísel

Desetinná čísla lze reprezentovat dvěma způsoby:

- v pevné řádové čárce
- v plovoucí řádové čárce

V plovoucí řádové čárce

Bitovou reprezentaci desetinných čísel v plovoucí řádové čárce upravuje standard IEEE 754. Předepisuje dvě základní přesnosti:

- **jednoduchá přesnost** (single-precision) - **32 bitů**
- **dvojitá přesnost** (double-precision) - **64 bitů**

Rozebereme jednoduchou přesnost - ukládání desetinných čísel do 32 bitů.

Nejdříve si ukažme, jak lze převést desetinné číslo z intervalu $<0;1$ do dvojkové soustavy (standard povoluje 23 bitů). Potřebujeme takové číslo vyjádřit jako rozvoj o základu 2. Např.:

$$0,36 = 0 * 1/2 + 1 * 1/4 + 0 * 1/8 + 1 * 1/16 + 1 * 1/32 + 1 * 1/64 + 0 * 1/128 + \dots$$

kde jsme vynechali dalších 16 zpřesňujících sčítanců. Přepsáním do mocnin 2 dostaneme:

$$0,36 = 0 * 2^{-1} + 1 * 2^{-2} + 0 * 2^{-3} + 1 * 2^{-4} + 1 * 2^{-5} + 1 * 2^{-6} + 0 * 2^{-7} + \dots$$

tedy

$$0,36_{(10)} = 0.01011100001010001111011_{(2)}$$

je jasné, že čím víc bitů je pro desetinné číslo povoleno, tím je jeho bitová reprezentace přesnější.

$$0,36_{(10)} = \boxed{0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1}$$

Podle standardu převádíme do binárního kódu pouze čísla normalizovaná, to jest vyjádřená pomocí desetinného čísla z intervalu $<1;2$ a jeho násobku s mocninou dvojky. Je tedy třeba jakékoliv desetinné číslo převést na normalizované. Např.: $0,0085 = 1,36 * 2^{-4}$.

Poté uložíme do 32 bitů informaci o:

- znaménku (1. bit)
- exponentu normalizovaného čísla (dalších 8 bitů)
- základu normalizovaného čísla zmenšeného o jedničku (zbývajících 23 bitů)

Shrnutí provedeme na příkladu čísla 0,0085:

1. Určení znaménka (nula znamená plus, jednička znamená mínus)

1. bit: 0

2. Normalizace čísla

$$0.085 / 2^{-1} = 0.17$$

$$0.085 / 2^{-2} = 0.34$$

$$0.085 / 2^{-3} = 0.68$$

$$0.085 / 2^{-4} = 1.36$$

tedy: $0.085 = 1.36 * 2^{-4}$

3. Převod exponentu do 8 bitů: exponent nula se převádí na číslo 127, tedy 01111111. Exponenty nenulové se k 127 přičtou. V našem případě tedy máme $127 + (-4) = 123$, $123_{(10)} = 01111011_{(2)}$
4. Převod mantisy normalizovaného čísla (zmenšené o jedničku) do 23 bitů. (Jak je demonstrováno výše) $1,36 - 1 = 0,36$
 $0.36_{(10)} = 0.01011100001010001111011_{(2)}$
5. Všechno to dáme dohromady:
 $0,0085_{(10)} = 00111101101011100001010001111011_{(2)}$



Převďte binární číslo **11000000110110011001100110011010** vyjádřené v plovoucí desetinné čárce na desetinné číslo v desítkové soustavě.

3.5. KOMPRIMACE

Komprimace (komprese)

Procesu, při kterém se **zmenšuje** velikost souborů a složek, se říká **komprimace** (komprese). Opačnému postupu, tedy **navrácení souboru do původního stavu** se říká **dekomprimace** (extrakce, dekomprese).

Soubor je posloupnost jedniček a nul, která vystupuje jako celek a má své jméno. **Velikost souboru** je **počet jedniček a nul** (ve vhodných jednotkách).

Existuje mnoho komprimačních algoritmů (návodů, jak soubor zmenšit). Jsou to složité výpočetní postupy s různou úrovní zmenšení původního souboru.

Máme-li způsob komprimace nějak hodnotit, musíme si určit kritéria, podle kterých poznáme, je-li komprimace pro nás dobrá, nebo není. Co očekáváme od dobré komprimace?

- výrazné zmenšení původního souboru
- vysokou rychlost komprimačního procesu

Komprimační poměr

Ono zmenšení původního souboru udává **komprimační (kompresní) poměr**.

$$\text{komprimační poměr} = \frac{\text{velikost zkomprimovaného souboru}}{\text{velikost původního souboru}}$$

Komprimační programy

Operační systém Windows XX podporuje komprimaci do formátu .zip:

- **komprimace** složky či souboru: klik PT nad složkou, souborem/ odeslat/ komprimovaná složka (metoda ZIP)
- **dekomprimace** (rozbalení) složky .zip: klik PT nad zip složkou/ extrahovat vše...

Lepší komprimační poměry přinášejí specializované komprimační programy se svými formáty. Nejrozšířenější

programy, které umí komprimovat jsou:

- **WinZIP**
- **WinRAR**
- **7-zip** (který budeme používat my.)

Formáty zkomprimovaných souborů

Stejně jako textové dokumenty mají různé formáty, také zkomprimované soubory mají své formáty. Každý komprimační program vytváří soubory ve svém specifickém formátu, ale většinou si je navzájem dovedou rozbalit. Nejpoužívanější je . **zip** a . **rar**, 7-zip má vlastní formát . **7z**, umí i zip i rar.

Existují dva základní druhy komprese:

- **bezeztrátová**

posloupnost bajtů souboru se zakóduje do jiné, kratší posloupnosti, která má ale totožnou informační hodnotu – po **dekódování získáme původní soubor**. Obvykle se používají na textové soubory, lze je použít i na grafiku a zvuk.

Příklady formátů používajících bezeztrátovou kompresi: **zip, rar, tiff, png, gif, flac, wma lossless**
K bezeztrátové komprimaci se používají známé algoritmy viz níže.

- **ztrátová**

ze souboru jsou odstraněny nedůležité detaily, jejichž ztrátu lidské smysly nevnímají (zrak a sluch).

Po dekódování již nezískáme originální soubor, některé **informace jsou trvale ztraceny**. Množství ztracených informací lze obvykle nastavit stupněm (kvalitou) komprese.

Příklady formátů používajících ztrátovou kompresi: **MP3, OGG, WMA, WMV, MPEG, DivX, JPEG...**

Ke zprátové kompresi se využívají specializované algoritmy šité na míru typu komprimovaných dat. Zvukové i obrazové soubory mají různé komprimační metody.

Kompresní algoritmy bezeztrátové

Příklady jednoduchých bezeztrátových komprimačních algoritmů jsou **RLC a LZ algoritmy**.

RLC (Run – Length code): Kompromace ČB textových dokumentů, nebo jednoduchých bitmapových obrázků s velkými jednobarevnými plochami. Textový dokument se považuje za ČB obrázek, celý se rozdělí na body, které jsou buď bílé, nebo černé. Původní data se pak převedou na posloupnost nesoucí pouze informaci o barvě bodu a počtu jeho výskytů za sebou.

LZ algoritmy (Lempel, Ziv), slovníkové: Textový dokument se porovná se slovníkem frází a jednotlivé řetězce se převedou na odkazy na frází ve slovníku. Nejdůležitější je vytvořit dobrý slovník. Obvykle je slovník (pomocný soubor) na počátku v základním tvaru, a během komprimace se rozrůstá. Tento slovník se po zabalení může odstranit, jelikož k rozbalení není potřeba. Dekomprimační algoritmus si ho opět vytvoří na základě zabalených dat.

Úkol 1: Na testovací složce vyzkoušejte kompresi s různými algoritmy (LZMA a Bzip2), s úrovní komprese normal, fast a ultra do formátů zip, 7z. Stopujte čas komprimace a vypočítejte komprimační poměr. Zjištěné údaje zapište do tabulky a zjistěte, která kombinace přináší nejlepší komprimační poměr a která kombinace je nejrychlejší.

Textové soubory DOC

úroveň komprimace	algoritmus	formát	původní velikost	velikost po komprimaci	čas	komprimační poměr
normální	LZMA	7z				
normální	Bzip2	zip				
fast	Bzip2	7z				
ultra	LZMA	7z				
ultra	Bzip2	zip				

Textové soubory ODT

úroveň komprimace	algoritmus	formát	původní velikost	velikost po komprimaci	čas	komprimační poměr
normální	LZMA	7z				
normální	Bzip2	zip				

Obrazové soubory JPG

úroveň komprimace	algoritmus	formát	původní velikost	velikost po komprimaci	čas	komprimační poměr
normální	LZMA	7z				
normální	Bzip2	zip				

Závěr:

Poznámka: šifrování: 7-zip umožňuje zašifrovat soubor heslem. Stačí soubor zkomprimovat a do dialogového okna zadat heslo.

4. ZÁLOHOVÁNÍ A ARCHIVACE

Zálohování

Poškození operačního systému nebo uživatelských dat je běžná událost. Dochází k nim vlivem poruchy HW, úmyslným poškozením počítačovými viry či neúmyslným zásahem samotného uživatele. Z tohoto důvodu je nutné vytvářet pravidelné zálohy dat a je-li to možné i zálohy operačního systému. Zálohy jsou určeny především pro obnovu poškozených dat, nikoliv pro jejich dlouhodobé ukládání - archivaci.

Zálohování je vytváření kopie dat na jiné paměťové médium.

Zálohování v praxi

pro domácí použití

- při **nevelkém objemu dat** (do kapacity 1 DVD = 4,5 GB) je nejlevnější způsobem jednou za čas vypálit data na DVD. Přesahuje-li velikost zálohovaných dat kapacitu DVD jen mírně, provedeme [komprimaci](#) pomocí některého z [kompresních programů](#).
- při **velkém objemu dat** násobně přesahujícím kapacitu DVD, nebo **při časté** (denní) **změně souborů** se vyplatí pořídit si externí HD a zálohovat denně pomocí synchronizačních programů (zadarmo je např. Unison). Synchronizační programy při prvním použití vytvoří identickou kopii dat a při dalším spuštění kontrolují změny v souborech a ty provedou i v záloze. Po každé synchronizaci máme k dispozici dvě identické kopie dat. Tímto způsobem nelze archivovat - nemáme k dispozici podobu dat z minulosti.

- k zálohování dat lze využít i **nástroj operačního systému**, který umožňuje vše provádět automaticky. Poskytuje jak vytváření záloh dat, tak i jejich obnovu při poškození.
- **zálohování samotného operačního systému** (jeho instalace, nastavení a instalace ovladačů) lze provádět v domácnosti jen ztěžím, operační systém Windows to umožňuje pouze ve verzích Profesional, Home verze tuto možnost neposkytují. Osobně doporučuji zakoupit zálohovací systém (zdarma lze použít i Clonezilla, ale ten nemám vyzkoušený :)) a po první instalaci operačního systému a jeho nastavení vytvořit obraz celého oddílu a v případě havárie (nebo i preventivně každého půl roku) obnovit operační systém z obrazu. Při dodržení tohoto postupu máte jistotu, že obnovíte operační systém čistý, nezavirovaný. Péči o váš počítač lze svěřit i odborníkům - v servisu požadujte oddělení dat od systému rozdělením disku na dvě části a vytvoření zálohy nového operačního systému pro snadnou obnovu v případě potřeby.

zálohování firemních dat:

- profesní SW (databáze, účetní programy...) vždy umožňuje více či méně snadné vytvoření zálohy dat a jejich případnou obnovu.

Archivace

Archivace dat je zajištění jejich dlouhodobého uchování (desítky až stovky let).

Archivace důležitých informací v předdigitální, papírové éře byla mnohem snadnějším procesem než nyní. Informace zapsané, či vytištěné na papíře se vydaly knižně, nebo uschovaly do šanonu a založily na správné místo v knihovně, kanceláři či archivu. Zde mohly bez úhony přečkat třeba staletí. A jestli nebyly smeteny živelnou pohromou, čekají tam dodnes. Jediným podstatným problémem archivace informací tímto způsobem po staletí může být leda přirozený vývoj jazyka. Za 500 let historie jediného národa se jeho jazyk stává pro současníky nesrozumitelným. Pomohou jedině průběžné překlady významných děl.

Dlouhodobé uchování informací, které jsou pouze v digitální podobě čelí více problémům:

1. Informace jsou **vázány na paměťová média**, a tím i na technologii, která je dokáže z média přečíst. Paměťová média se rychle mění (magnetické pásky, diskety, CD, DVD, flash paměti, pevné disky...) a za sto let těžko bude existovat mechanika, která by přečetla obsah současného CD.
2. Existuje množství **různých formátů dat**, často navzájem nekompatibilních a závislých na programu a firmě, která ho vytvořila. Ta za 20 let už nemusí existovat. Navíc i jeden formát se v čase mění, takže není doc jako doc, není pdf jako pdf.
3. Informace v digitální podobě jsou lehce **modifikovatelné**, bez důkladného zkoumání nelze rozeznat, jestli jsou původní, nebo modifikovaná. Na rozdíl od papírové, tištěné podoby.

Firmy a instituce jsou ze zákona povinny archivovat data, například pro Finanční úřad, Správu sociálního zabezpečení a podobně, v desítkách let. Jiné veřejné instituce jako muzea, knihovny by rády archivovaly data v řádu stovek let. V tom případě jsou řešením archivační systémy, které odpovídají mezinárodním archivačním normám. Archivační systémy musí mít vyřešeny výše popsané problémy, tedy musí být nezávislé na konkrétním HW a konkrétních formátech dat (pomocí pravidelných převodů na modernější technologie), musí mít zabezpečenu jejich nemodifikovatelnost.

5. TŘÍDĚNÍ (ŘAZENÍ)

Třídění je vžitě, ale nevýstižné pojmenování pro uspořádání množiny objektů podle zadaných kritérií. (Např. podle abecedy vzestupně.) Výstižnější název je **řazení**, v odborné literatuře se setkáváme s oběma názvy. Cílem řazení je usnadnit (zrychlit) pozdější vyhledávání. Postup, jakým se řazení uskutečňuje předepisují **třídící (řadící) algoritmy**. Řadících algoritmů je několik, každý má své výhody a nevýhody.

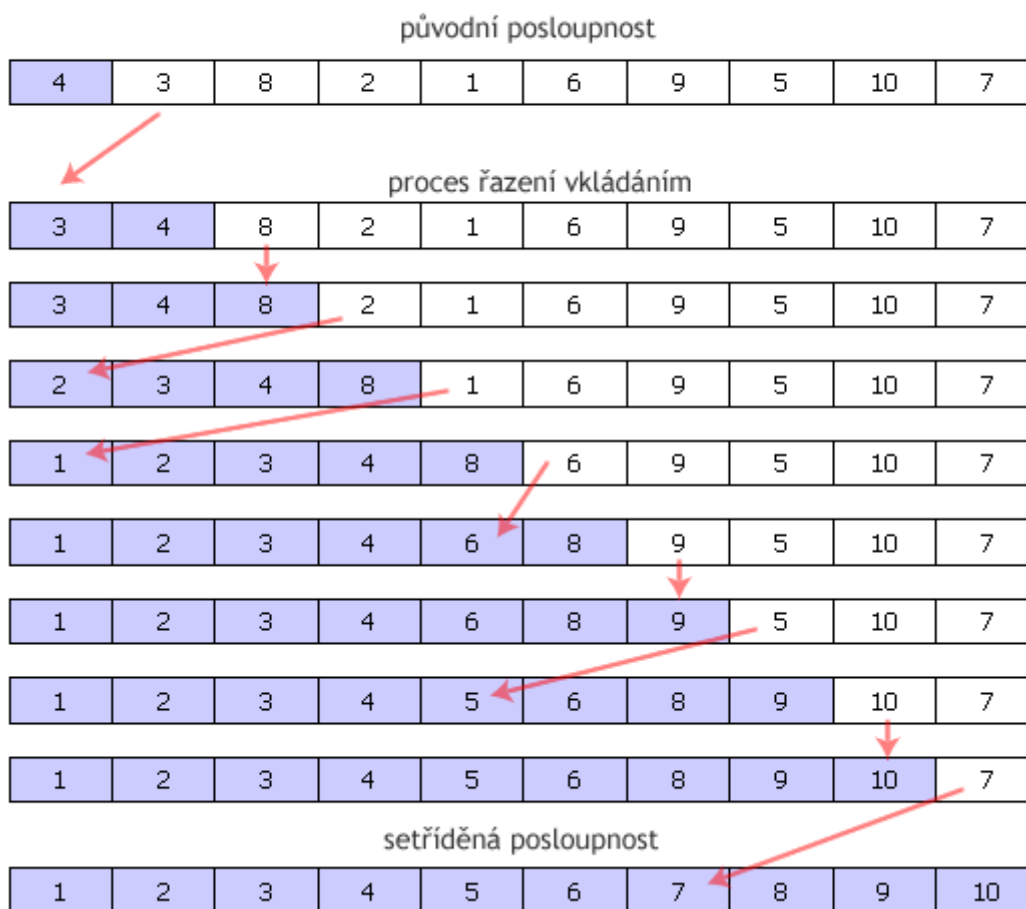
Metody řazení můžeme rozdělit do skupin:

- řazení vkládáním
- řazení výběrem
- řazení výměnou
- řazení rozdělováním

5.1. ŘAZENÍ VKLÁDÁNÍM

Původní posloupnost $a_1 \dots a_n$ si rozdělíme na dvě části, cílovou a zdrojovou posloupnost. Zpočátku je cílovou posloupností a_1 a zdrojovou posloupností $a_2 \dots a_n$. Třídění probíhá tak, že ze zdrojové posloupnosti vybíráme postupně prvky $a_2, a_3, a_4 \dots$ a zařazujeme je do cílové posloupnosti na správné místo, které najdeme porovnáváním s již zařazenými prvky cílové posloupnosti. (Po zařazení musíme úsek mezi novým a starým umístěním přesunutého prvku přeindexovat.) Ze zdrojové posloupnosti prvků ubývá, v cílové přibývá, již ve správném pořadí.

Řazení vkládáním

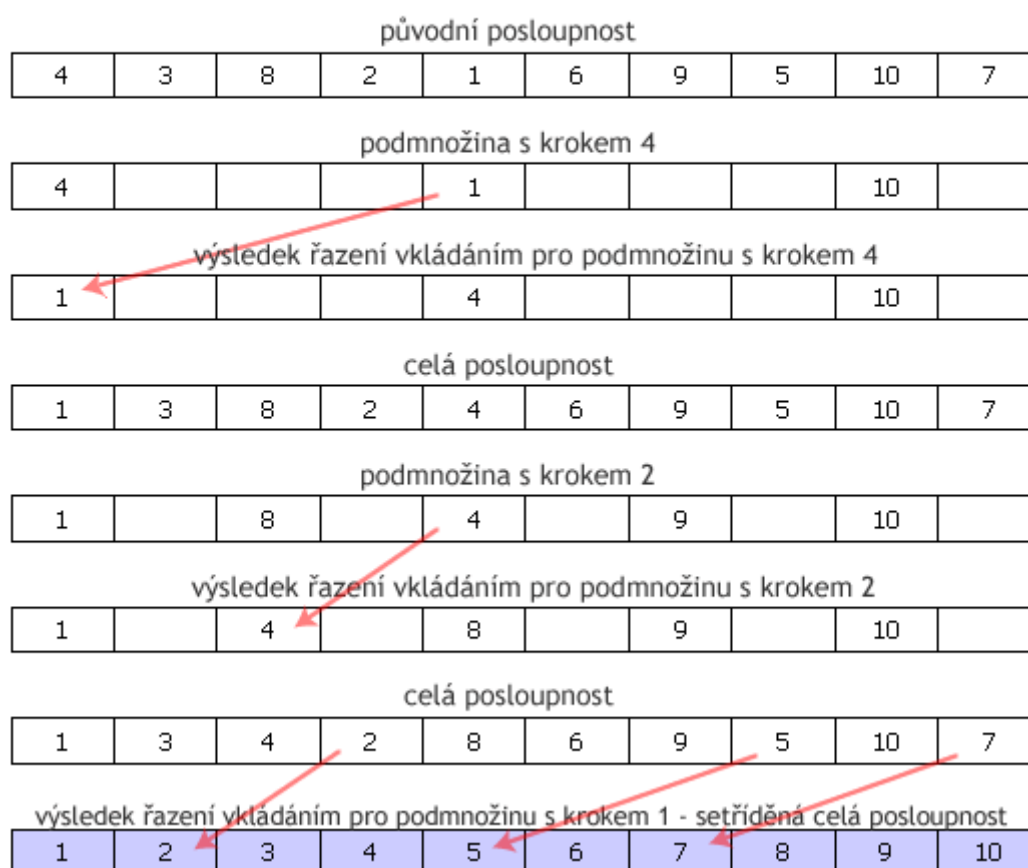


Tato základní třídící metoda se dá vylepšit tzv. **vkládáním se zmenšováním kroku**, které se podle jejího autora říká **Shellův třídící algoritmus - Shell Sort**.

Shell sort

Metoda spočívá v aplikaci přímého vkládání na zvláštní podmnožiny původní posloupnosti. Nejprve se aplikuje např. na podmnožinu s krokem 8 (tvořenou každým 8. prvkem původní posloupnosti.) Poté se krok zmenší třeba na 4, dále na 2 a nakonec 1. Zajímavé je, že nejlepší výsledky přináší posloupnost kroků 1, 4, 13, 40, 121, ... a 1, 3, 7, 15, 31, ... psáno od konce.

Shell sort



5.2. ŘAZENÍ VÝBĚREM

Metoda spočívá ve vyhledání nejmenšího prvku v celé posloupnosti $a_1, .. a_2,$ a jeho výměně s prvním prvkem a_1 . Tento postup opakují pro posloupnost o 1 prvek kratší, tedy $a_2, ... a_n,$ dále $a_3, ... a_n,$ atd.

Řazení výběrem



A jak nalézt nejmenší prvek posloupnosti? Třeba takto:

a_1 porovnávám postupně s $a_2, s a_3, ...$ dokud nenarazím na nějaký menší prvek, řekněme a_k . Poté převezme štafetu a_k , porovnávám ho s $a_{k+1}, a_{k+2}, ...$ dokud nenarazím na nějaký menší ... Takto se dostanu až na konec posloupnosti a mám nalezen její nejmenší prvek.

Přímý výběr lze vylepšovat tím, že při porovnávání během výběru nejmenšího prvku získané informace využijeme efektivněji. K tomu se používají **stromové algoritmy** a neefektivnější z nich – **třídění haldou**. Složitost propracovaných algoritmů přímého výběru – třídění haldou klesá z n na $n \cdot \log n$.

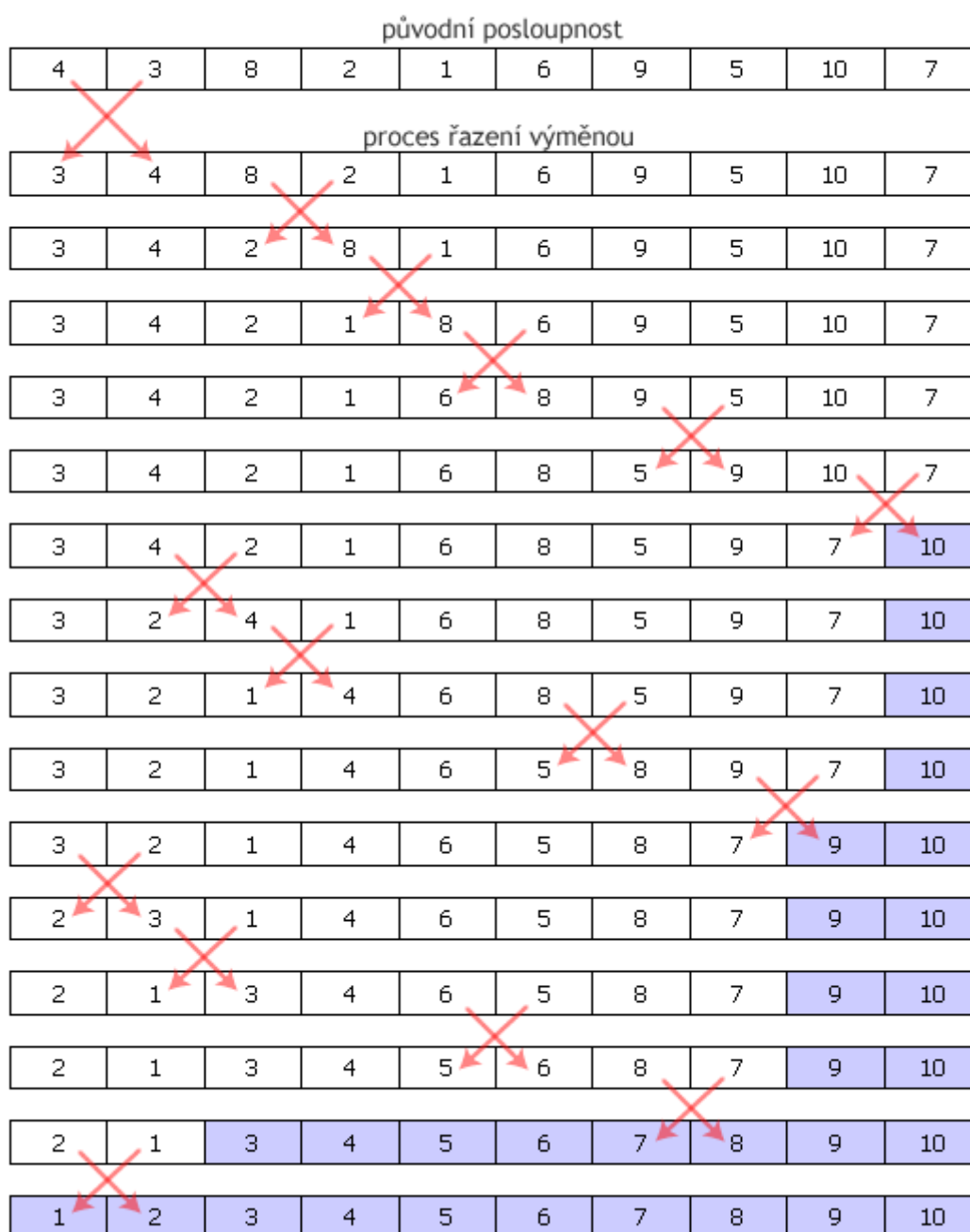
5.3. ŘAZENÍ VÝMĚNOU

Metoda je rozšířená pod názvem bublinové třídění, neboli **Buble Sort**.

Buble sort

Prvky posloupnosti porovnávám po dvojicích, a_1 s a_2 , a_2 s a_3 , ... a_{n-1} s a_n , přičemž je-li $a_i > a_{i+1}$, vyměním je. Postup opakuji tak dlouho, dokud nevznikne setříděná posloupnost.

Řazení výměnou - Buble sort



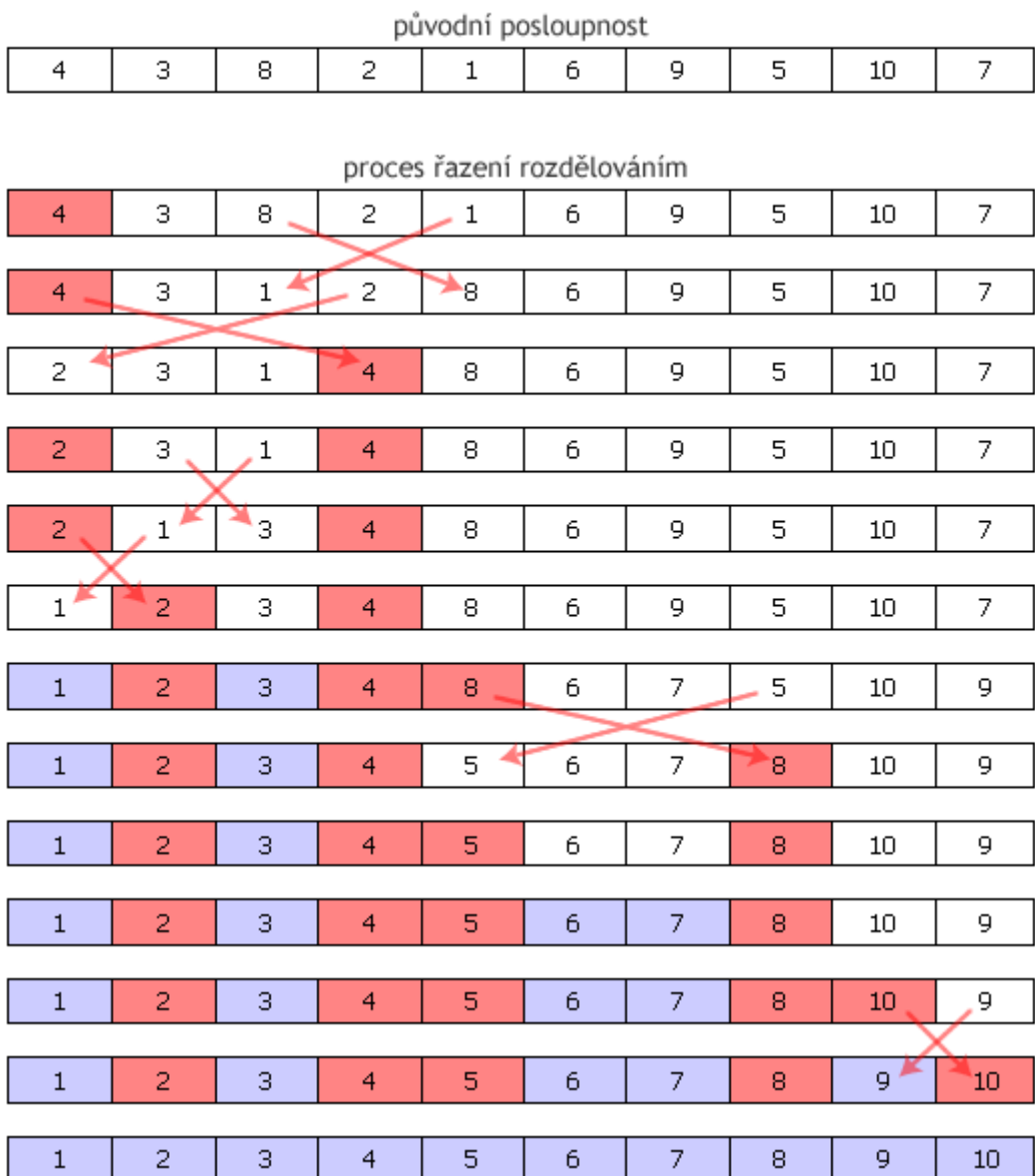
Výrazným vylepšením tohoto jinak málo efektivního algoritmu je **metoda řazení rozdělčováním**, která pro své

vynikající parametry nese jméno rychlé třídění, neboli **Quick sort**.

5.4. ŘAZENÍ ROZDĚLOVÁNÍM

V posloupnosti vybereme náhodně jeden prvek a_i , (nazvěme ho START) (lehce se programuje $a_i = a_1$) a budeme procházet posloupnost zleva do prava dokud nenajdeme prvek větší než START a zároveň budeme procházet posloupnost zprava do leva dokud nenajdeme menší než START. Tyto dva nalezené prvky navzájem vyměníme. Porovnávání pokračuje, dokud se procházení z obou stran nestřetne. Tímto způsobem se nám posloupnost rozdělí na dvě části. V levé jsou všechny prvky menší než START a v pravé jsou všechny prvky větší než START. Prvek START porovnáme s posledními dvěma (nebo jedním) prvkem a s tím správným jej vyměníme. Tímto je prvek START na správném místě. Tento postup opakujeme pro posloupnosti nalevo od START a napravo od START. V nich zvolíme v každé NOVÝ START který nám také rozdělí každou z nich na dvě části, ... a tak dále až do úplného setřídění původní posloupnosti.

Řazení rozdělčováním - Quick sort



Správná volba prvku START má velký vliv na rychlost řazení posloupnosti. Nejrychlejší řazení by bylo zvolit START = mediánu posloupnosti, avšak hledat medián je stejně pracný problém jako setřídít celou posloupnost. V praxi se proto používá několik metod:

- START = první prvek, nebo kterákoli jiná fixní pozice
- START = náhodný prvek – často používaná metoda
- START = medián z x náhodně zvolených čísel

Metoda je ve většině případů rechlá a často používaná.

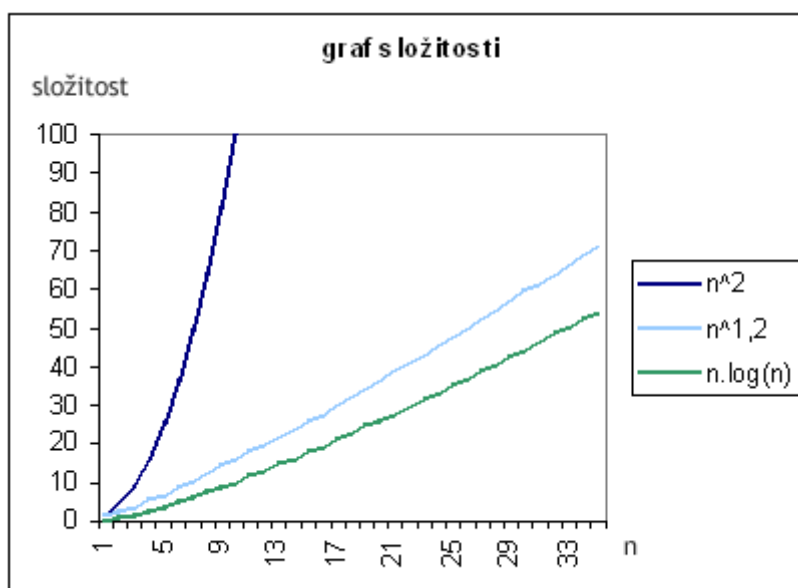
Prvku START se v anglické literatuře říká **pivot**.

5.5. HODNOCENÍ TŘÍDÍCÍCH ALGORITMŮ

Třídících metod je několik, nabízí se otázka, která z nich je nejlepší. Je to samozřejmě ta z nich, která třídí rychle velké množství údajů. Odpověď není snadná. Kriteřiem hodnocení je tzv. **složitosť algoritmu**, která udává **závislost počtu operací** provedených během třídění (porovnávání, výměna prvků ...) na počtu prvků posloupnosti **n**. Stanovit složitosť propracovaných algoritmů jako je Shell sort nebo Quick sort je složité matematický problém. Pro každou metodu existuje uspořádaní původní posloupnosti, pro které je metoda velmi rychlá a naopak uspořádaní, pro které je pomalá. Je otázkou, jak často v praxi nastávají tyto krajní případy, jak moc se v nich třídění zrychlí, nebo zpomalí, atd.

Pořadí třídících algoritmů podle odhadu složitosti (od nejhoršího k nejlepšímu):

algoritmus	složitosť
Buble Sort	n^2
řazení vkládáním	n^2
řazení výběrem	n^2
Shell Sort	$n^{1,2}$
Quick Sort	$n \cdot \log n$



Metoda Quick Sort je v průměru nejrychlejší z nich, přestože existuje situace, ve které se výrazně zpomalí. Jde o již

setříděnou původní posloupnost, ve které náhodný výběr prvku START pokaždé padne na největší prvek oblasti. Přesto je Quick Sort nejpoužívanější třídící metodou.

6. VYHLEDÁVÁNÍ

Pod pojmem **vyhledávání** v této kapitole rozumíme **nalezení prvku v množině** prvků standardního datového typu, např. v řetězci, seznamu, sekvenci ... Podobně jako u řazení existuje několik známých metod vyhledávání - vyhledávací algoritmy. Vyhledávání ve webových stránkách je podrobně popsáno v další kapitole.

Lineární (sekvenční) vyhledávání

je nejjednodušší vyhledávací algoritmus: algoritmus postupně prochází množinu prvek po prvku a porovnává s hledaným prvkem, dokud ho nenajde.

použitelné pro: neseřazené množiny

složitosť: úměrná počtu prvků množiny - n

Binární vyhledávání (metoda půlení intervalu)

Je vyhledávací metoda na seřazené množině: algoritmus najde medián (střední hodnotu) množiny a porovná ho s hledaným prvkem. Je-li hledaný prvek menší než medián, hledání pokračuje v té polovině množiny, ve které jsou menší prvky. V ní algoritmus najde medián a porovná ho s hledaným prvkem... Takto pokračuje rekurzivně dokud hledaný prvek nenalezne.

použitelné pro: seřazené množiny

složitosť: úměrná $\log n$

V informatice pod pojmem **vyhledávání** rozumíme také **hledání řešení nějakého problému**, například hledání řešení rovnice, hledání optimálního tahu ve hrách typu šachy či dáma. I pro takové případy existují známé algoritmy např. Minimax, Backtracking, Alfa-beta ořezávání...

6.1. INTERNETOVÉ VYHLEDÁVAČE

Internetový vyhledávač je **služba**, která poskytuje vyhledání webové stránky, obsahující požadované informace.

Uživatel zadává do rozhraní vyhledávače dotaz prostřednictvím **klíčových slov**, která charakterizují hledanou informaci a vyhledávač na základě **své databáze** vypíše seznam odkazů na stránky, které podle něho hledané informace obsahují.

Cílem vyhledávačů je poskytnout uživateli při odpovědi na dotaz informace s **maximální relevancí a aktuálností** (pravdivost informace nedokáže posoudit :) na prvních místech odkazů. Proto každý vyhledávač různými způsoby **hodnotí důležitost webových stránek**, které mají ve své databázi (např. PageRank).

Jak vyhledávač pracuje

Vyhledávač je systém pracující na desítkách až tisících počítačů. Ve **své databázi** má uloženy všechny stránky světa a u každé z nich posoudí **indexem** (vahou) její **důležitost**.

Obecně většina internetových vyhledávačů pracuje ve třech krocích:

1. **procházení** webových stránek
2. **vytvoření databáze** výskytu slov
3. **indexování**
4. **poskytování** odpovědí na dotazy

Procházení webových stránek

Webové stránky neustále prochází automat (vyhledávací robot). Začne u významných rozcestníků (katalog

Seznamu, Yahoo! Directory ...) a postupně prochází všechny odkazy hlouběji a hlouběji a snaží se prozkoumat všechny stránky internetu. Každou stránku si uloží na pevný disk i s její URL adresou.

Vytvoření databáze výskytu slov

Stránky, které robot uložil na pevný disk systém zpracuje a vytvoří z nich databázi všech nalezených slov a k nim adresy na nichž se vyskytují. To však nestačí, taková datbáze je obrovská a vyhledávání v ní (lineární) by trvalo příliš dlouho. Proto je třeba ohodnotit důležitost slov a stránek na nichž se vyskytují. K tomu slouží indexování:

Indexování

Indexování databáze je klíčový problém každého vyhledávače. Jak dokáže stroj posoudit relevanci informace, když jí sám nerozumí?

K tomu existuje několik metod a každý vyhledávač je využívá po svém (např. PageRank u Google, S-Rank u Seznamu, JyxoRank u Jyxo). Obecně se pro indexování bere v úvahu:

- **Váha slov**

Váhu slova zvyšuje, je-li slovo:

- **v titulku stránky**
- **v nadpisu** (bez css nelze)
- **blízko začátku** stránky
- či se na stránce **opakuje**
- **Atraktivita stránky**
Stránka má **vyšší hodnocení**, směřuje-li na ni množství **odkazů** z jiných stránek, protože zřejmě obsahuje zajímavé informace.
- **Serióznost Webu**
Dlouhodobě prověřené webové servery, které obsahují velké množství kvalitních stránek, jsou při výpočtu váhy zvýhodněny. Seznam takových serverů mohou vytvářet kromě automatů i lidé na základě svých zkušeností.
- **Sponzorované odkazy**
Váha odkazu se zvyšuje zaplacením poplatku. Seriózní vyhledávače se této praxi vyhýbají nebo zřetelně oddělují výsledky zobrazené na základě komerčního zvýhodnění. Tento způsob je jedním z možných zdrojů příjmů vyhledávače.
- **Technická kvalita**
Váha stránky se zvyšuje, pokud je vytvořena podle webových standardů (aktuální verze html, xhtml, css).

Tvůrci webů mohou **zneužívat** těchto informací a uměle zvyšovat váhu svých stránek opakováním slov, vytvářením zbytečných nadpisů, vytvářením falešných stránek, které odkazují na stránku, která má získat vyšší hodnocení...

Vyhledávač se **brání** sledováním podezřelého opakování slov, náhlého hromadění odkazů... Takové stránky jsou pak penalizovány.

Odpovědi na dotazy

Vyhledávač poskytuje uživatelům vstupní formulář, do kterého vkládáme klíčová slova, v rozšířeném vyhledávání můžeme využít:

- logických spojek (AND, OR, NOT)
- zadání pevné fráze
- upřesnit doménu (site)
- vybrat jazyk
- ...

Vyhledávač pak **formulář zpracuje** a poskytne nám odkazy na stránky seřazené od **nejvyššího indexu**. Pro vyšší přehlednost se zobrazuje kromě odkazu ještě titulek stránky, okolí nalezených slov a případně i další informace (stáří informace, kvalita odkazu, ...).

V poslední době je velmi žádanou službou **SEO** - technika, která dokáže stránky upravit tak, aby se co nejlépe umístily ve výsledcích vyhledávání

Nejznámější vyhledávače**Ve světě**

- AltaVista
- Google (vyhledávač)
- Yahoo

V České republice

- Atlas.cz
- Centrum.cz
- Jyxo.cz
- Seznam.cz