

Základy forenzních databází

Tereza Uhlíková

verze y25

Kdo jsem

Tereza Uhlíková

Ústav analytické chemie

skupina teoretické spektroskopie

místnost A277

<https://web.vscht.cz/~uhlikovt/>

tereza.uhlikova@vscht.cz

1. lekce

- Co je to databáze
- Proč, kdo, kdy, jak ...
- Něco málo z historie
- CAP problém

Dobrodružství kriminalistiky - televizní seriál
Dějiny psané Římem - Vojtěch Zamarovský

2. lekce

- Základy informatiky
- Software
 - informace
 - záznam informace
 - číselné soustavy
 - písmenné kódy
- Hardware
 - Alan Turing, John von Neumann a počítač
 - historie vývoje počítače & super počítač
 - procesor a datová uložště

Alan Turing - Enigma
Blade Runner

Dnešní lekce

velká uložště

Enigma

RSA šifra

- Algoritmus
 - vlastnosti (zápis, struktura)
 - Datové typy
 - strukturované programovací jazyky

Arthur C. Clarke - Devět miliard božích jmen

Velká uložště

402,74 million TiB dat se tvoří každý den

<https://explodingtopics.com/blog/data-generated-per-day>

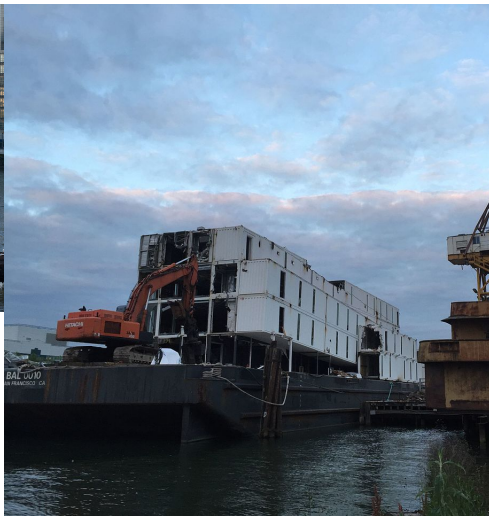
google - 130 datacenter, cca 1 biliony euro na každý

<https://datacenters.google/locations/>

google clouds - 15Gb na účet, cca 3×10^9 účtů



Velká uložště



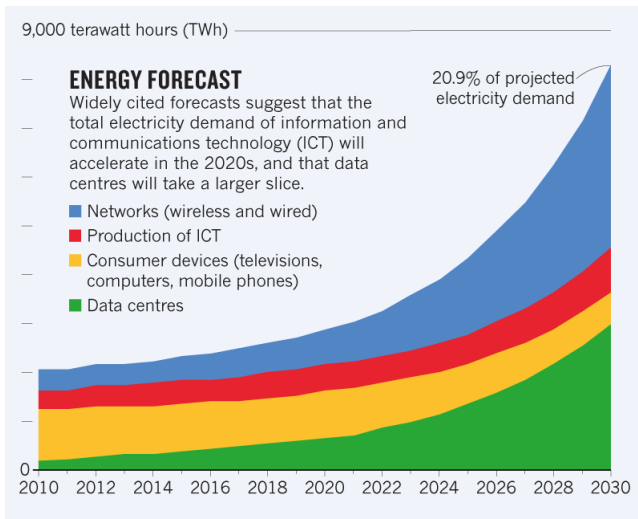
Velká uložště

google roční spotřeba energie 2024 cca 30 TWh

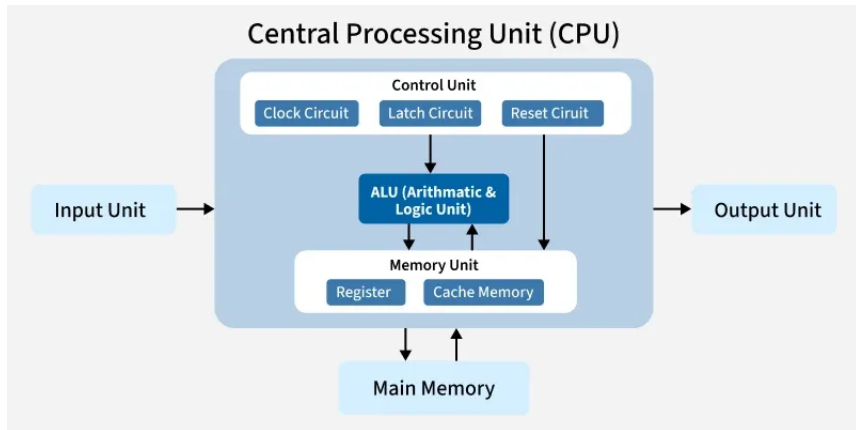
roční výroba energie v Temelínu cca 15 TWh



energie



CPU



<https://www.geeksforgeeks.org/computer-science-fundamentals/central-processing-unit-cpu/>

Enigma



<https://cs.khanacademy.org/computing/computer-science/cryptography/crypt/v/case-study-ww2-encryption-machines>
<https://www.mub.eps.manchester.ac.uk/science-engineering/2018/11/28/cracking-stuff-how-turing-beat-the-enigma/>

Zakodujte si sami

zapište jedno jméno do strojového kodu
rozšifrujte jméno

RSA šifra

RSA (Rivest–Shamir–Adleman) je asymetrický šifrovací algoritmus založený na rozkladu velkých čísel na prvočísla.

asymetrický - každý uživatel má pár klíčů: veřejný (zašifrování) a soukromý (dešifrování), jeden klíč nelze snadno odvodit z druhého.

rozkladu čísel na prvočísla: $15 = 3 \cdot 5$; $2800 = 2^4 \cdot 5^2 \cdot 7$

Kolik je prvočísel?

Jak jsou rozmístěna mezi reálnými čísly?

Prvočíselná věta - přibližně popisuje rozmístění prvočísel mezi přirozenými čísly

<https://www.geogebra.org/m/nhg2JjzV>

<https://mathworld.wolfram.com/PrimeSpiral.html>

→ neexistuje efektivní algoritmus pro klasický počítač

→ složitost sub-exponenciální $O(\exp^N)$

RSA Šifra

Máme dvě prvočísla p, q .

Spočítáme jejich součin $n = p \cdot q$

Spočítáme $\varphi(n) = (p - 1)(q - 1)$

Vybere se veřejný exponent e , který je nesoudělný s $\varphi(n)$

Spočítáme privátní exponent d , což je inverze e modulo $\varphi(n)$

Šifrování a dešifrování probíhá:

$$C = M^e \mod n$$

$$M = C^d \mod n$$

<https://sifrovani.fd.cvut.cz/rsa.html>

<https://cs.khanacademy.org/computing/computer-science/cryptography/modern-crypt/v/intro-to-rsa-encryption>

Algoritmy a programování

Proč algoritmy a programování?

Animation vs. Coding

<https://www.youtube.com/watch?v=EFmxPMdBqmU>



<https://flatironschool.com/blog/reasons-why-you-should-learn-computer-programming/>

Proč algoritmy a programování?

- řeší problémy
- dělí velký problém na menší
- upevňuje logické myšlení
- ujasňuje skutečnost - popisuje skutečnost v základních pojmech
- dává jasné definice, co a kdy se má udělat
- vytváří proces řešení problému
- ...

Pracovní postup - části

Co by mělo být součástí dobrého návodu?

- nadpis - k čemu postup slouží
- popis vstupu
- popis výstupu
- pracovní podmínky + co je či není očekáváno
- jednotlivé kroky postupu

Popis postupu

- jazyková správnost
- instrukce popsány dost podrobně, všechna slova jasně a jednoznačně vysvětlena, jasná návaznost instrukcí („co dělat dál“):
- u rozhodování nutno pokrýt všechny alternativy, konec výslovně sdělit
- pokrytí všech očekávaných eventualit

Algoritmus

formálně zapsaný postup pro řešení daného problému

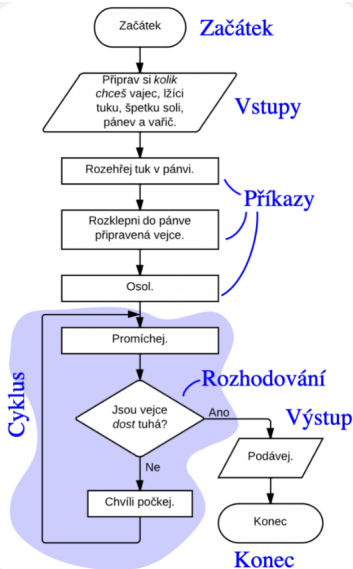
– jde o posloupnost elementárních kroků

- kuchařka: recepty
- výpočet nejmenšího společného dělitele
- řešení Schrödingerovi rovnice

– ten, kdo provádí algoritmus, je **procesor**

procesor určuje, jaké budou elementární kroky, podle toho co zná (jak je naprogramovaný)

Algoritmus



Algoritmus

Napište algoritmus pro tvorbu bakalářské práce.

Algoritmus a Program

příklad:

– recept na bábovku

- procesorem je kuchař, elementární kroky jsou instrukce typu např. odvaž 0,5 kg hladké mouky...(záleží na odbornosti kuchaře)

– algoritmus pro výpočet kořenů kvadratické rovnice

- procesorem je procesor počítače, elementární kroky jsou instrukce daného procesoru

napsat program znamená:

a) vymyslet algoritmus

b) zapsat jej v nějakém programovacím jazyce

Eukleidův algoritmus



Eukleidův algoritmus (též Euklidův) je nejstarší zaznamenaný algoritmus 300 př.nl.

Eukleidův algoritmus



Eukleidův algoritmus (též Euklidův) je algoritmus, kterým lze určit největší společný dělitel dvou přirozených čísel.

Největší společný dělitel (NSD) dvou celých čísel je největší číslo, které beze zbytku dělí obě čísla. Příklady: $\text{NSD}(18, 24) = 6$, $\text{NSD}(30, 85, 90) = 5$.

Použití:

redukci zlomků na jejich nejjednodušší formu

RSA - nesoudělná čísla

Eukleidův algoritmus

- vypíšeme všechny dělitele
- pomocí prvočíselného rozkladu
- Euklides

$$78 \div 66 = 1 \text{ remainder } 12 \quad (78 = 66 \times 1 + 12)$$

$$66 \div 12 = 5 \text{ remainder } 6 \quad (66 = 12 \times 5 + 6)$$

$$12 \div 6 = 2 \text{ remainder } 0 \quad (12 = 6 \times 2 + 0)$$

6 = Greatest Common Factor

Mějme dána dvě přirozená čísla, uložená v proměnných **u** a **w**.

Dokud **w** není nulové, opakuj:

Do **r** ulož zbytek po dělení čísla **u** číslem **w**

Do **u** ulož **w**

Do **w** ulož **r**

Konec algoritmu,
v **u** je NSD.

Eukleidův algoritmus v různých jazycích

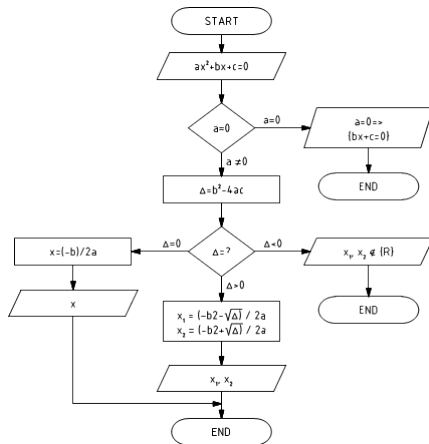
[https://www.geeksforgeeks.org/
euclidean-algorithms-basic-and-extended/](https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended/)

Algoritmus pro výpočet kořenů kv. rovnice

1. je-li $a = 0$, krok 8
2. $D = b^2 - 4ac$
3. je-li $D < 0$ krok 6
4. Rovnice má dva reálné kořeny ...
5. Konec
6. Rovnice má dva komplexní kořeny ...
7. Konec
8. atd.

Zápis algoritmů

vývojové diagramy (grafický zápis) - orientovaný graf



- jednotlivé kroky jsou vyjádřeny grafickými symboly dle typu operace
- posloupnost vykonávání kroků je vyjádřena orientovanou hranou

Základní komponenty algoritmu

- ❶ posloupnost (sekvence) příkazů – kroky v daném pořadí
- ❷ větvení (podmínka) – výběr dalších prováděných kroků závisí na splnění/nesplnění nějaké podmínky
- ❸ cyklus – opakované provádění kroků
 - ❶ s pevným počtem opakování
 - ❷ s podmínkou (na konci/na počátku) - provádění kroků se opakuje, dokud je/není splněna podmínka

Vlastnosti algoritmu

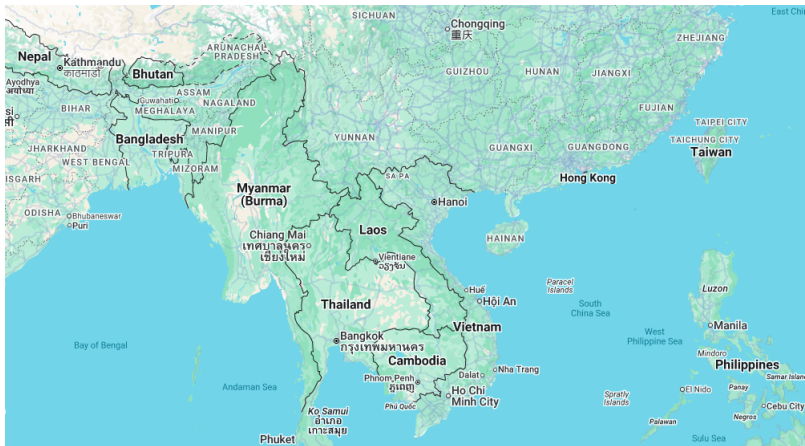
Vlastnosti algoritmů

- rezultativost (výsledkovost) - vydá výsledek
- elementárnost (jednoduchost) – skládá se z konečného počtu jednoduchých kroků
- determinovanost (jednoznačnost) – po každém kroku lze rozhodnout, zda proces skončil
- finitnost (konečnost) – počet kroků algoritmu je konečný

Další vlastnosti algoritmů

- korektnost (správnost) - správný výsledek
- univerzálnost (obecnost, hromadnost) - lze jej použít pro řešení více podobných úloh
- efektivita (úspornost) - rychlejší a pamětově méně náročné

Hanojské věže

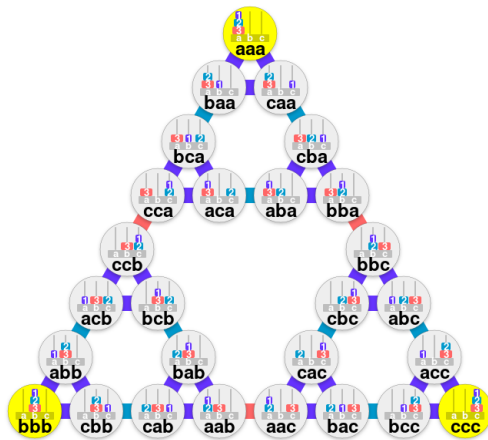


<https://vesmir.cz/cz/casopis/archiv-casopisu/2010/cislo-9/hanojske-veze.html>

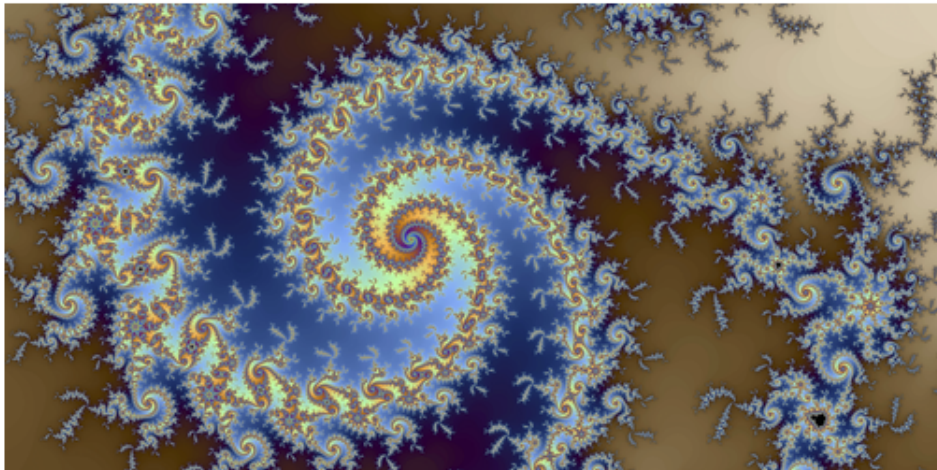
https://en.wikipedia.org/wiki/Tower_of_Hanoi

Hanojské věže

Grafické řešení vede na fraktály.



Fraktály



Algoritmus

```
#include <stdio.h>
// C recursive function to solve tower of hanoi puzzle
void towerOfHanoi(int n, char from_rod, char to_rod, char aux_rod)
{
    if (n == 1)
    {
        printf("\n Move disk 1 from rod %c to rod %c", from_rod, to_rod);
        return;
    }
    towerOfHanoi(n-1, from_rod, aux_rod, to_rod);
    printf("\n Move disk %d from rod %c to rod %c", n, from_rod, to_rod);
    towerOfHanoi(n-1, aux_rod, to_rod, from_rod);
}

int main()
{
    int n = 4;        // Number of disks
    towerOfHanoi(n, 'A', 'C', 'B');    // A, B and C are names of rods
    return 0;
}
```

output

Move disk 1 from rod A to rod B
Move disk 2 from rod A to rod C
Move disk 1 from rod B to rod C
Move disk 3 from rod A to rod B
Move disk 1 from rod C to rod A
Move disk 2 from rod C to rod B
Move disk 1 from rod A to rod B
Move disk 4 from rod A to rod C
Move disk 1 from rod B to rod C
Move disk 2 from rod B to rod A
Move disk 1 from rod C to rod A
Move disk 3 from rod B to rod C
Move disk 1 from rod A to rod B
Move disk 2 from rod A to rod C
Move disk 1 from rod B to rod C

Datové typy

- charakterizuje vlastnost proměnné

Numerické typy:

- celočíselný (INTEGER)
 - SHORTINT - jednobajtový 1B $\rightarrow 2^8 = 256$ různých hodnot
 - SMALLINT - 2B $\rightarrow 2^{16} = -32768$ až 32767
 - INTEGER - 4B $\rightarrow 2^{32}$ - standardní (32-bitový) ...
- reálný (FLOAT)
 - NUMERIC(p,q): p – počet platných číslic, q – počet desetinných míst
 - FLOAT(n): reálné číslo s plovoucí desetinnou čárkou, n – počet des. míst
 - REAL: reálné číslo (rozsah dán implementací příslušného DBS)

Znakové řetězce:

- CHAR(n): *n*-znakový řetězec (např. CHAR(10))
- VARCHAR(n): *n*-znakový řetězec (*n* max. 255)
- CLOB(n): dlouhé *n*-znakové řetězce (* v implementaci MySQL)

Datové typy

Bitové řetězce:

- BIT(n): n udává počet bitů

Temporální data (přesný formát závisí na konkrétní implementaci DBS):

- DATE
- TIME
- TIMESTAMP

Logické typy:

- BOOLEAN (pravda - nepravda)
- LOGICAL

Skripta, stránky a materiály

https://sites.ff.cuni.cz/uisk/wp-content/uploads/sites/62/2016/01/Datov%C3%A9-modely-a-n%C3%A1vrhy-rela%C4%8Dn%C3%ADch-sch%C3%A9mat_Sk%C5%99ivan.pdf

<http://www.databaze.chytrak.cz/modely.htm>

<http://books.fs.vsb.cz/dbacc20/Welcome.htm>

<http://books.fs.vsb.cz/>

http://projekty.fs.vsb.cz/463/edubase/VY_01_042/Z%C3%A1klady%20infor

<http://home.zcu.cz/~pkral/texty.pdf>

<https://www.konvoj.cz/Nakladatelstvi/Knihy/100289-haluza-rybicka-hala-scriptum-43-uvod-do-informatiky/100289-uvod-do-informatiky.pdf>

http://www.bigyzr.cz/shared/clanky/2893/ICT-Pripravy/IS4-Pripravy_informace.pdf

<https://homel.vsb.cz/~hom50/DATABASE/SLBDBASE/DBS3METU.HTM>

<https://mathigon.org/course/fractals/sierpinski>

www.baeldung.com/cs/euclid-time-complexity

<https://www.promotic.eu/cz/pmdoc/Subsystems/Db/Postgres/DataTypes.htm>

<https://www.promotic.eu/cz/pmdoc/Subsystems/Db/MySQL/DataTypes.htm>

<https://mathigon.org/course/fractals/sierpinski>

AI Cat Learns to Run <https://www.youtube.com/watch?v=aBp-3pmKNBY>