

Základy AI v chemii

Lekce 3: Datová uložště, algoritmy

Tereza Uhlíková

verze y26

Kdo jsem

Tereza Uhlíková

Ústav analytické chemie

Skupina Teoretické Spektroskopie - Broad Peak

místnost B4335

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

tereza.uhlikova@vscht.cz

Co bylo minule

- Záhadný soubor: bez metadat (kódování, oddělovač, jednotky) jsou data jen hromada čísel
- Bit = jedna otázka ano/ne; 20 správných otázek rozliší milion látek
- Bajt: „A“ = 65 = 01000001; SMILES kofeinu = 28 B; jedno IR spektrum $\approx$ 29 kB
- Když se data pokazí: Mars Climate Orbiter (jednotky), Ariane 5 (přetečení), Excel a geny

Kolik bitů potřebujete na jedno písmeno anglické abecedy (26 písmen)?

Co bylo minule

- Záhadný soubor: bez metadat (kódování, oddělovač, jednotky) jsou data jen hromada čísel
- Bit = jedna otázka ano/ne; 20 správných otázek rozliší milion látek
- Bajt: „A“ = 65 = 01000001; SMILES kofeinu = 28 B; jedno IR spektrum $\approx$ 29 kB
- Když se data pokazí: Mars Climate Orbiter (jednotky), Ariane 5 (přetečení), Excel a geny

Kolik bitů potřebujete na jedno písmeno anglické abecedy (26 písmen)?

5 bitů, protože $2^4 = 16 < 26 \leq 32 = 2^5$

O čem budeme mluvit dnes

„Od křemíku k tajemství“

- **Hardware:** Turing, von Neumann, paměti, superpočítače, úložiště, energie
- **Algoritmus:** co to je, jak ho zapsat, Eukleides, Hanojské věže
- **Kódy a šifry:** CAS číslo, Caesar, **Enigma**, RSA
- **Programovací jazyky a datové typy:** první krok k programování

DATA → **ALGORITMY** → DATABÁZE → VYHLEDÁVÁNÍ →
STROJOVÉ UČENÍ → AI

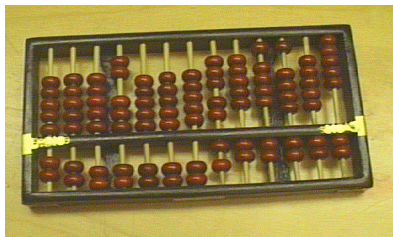
Kód Enigmy (The Imitation Game, 2014)

Kontakt (1997); SETI@home

Arthur C. Clarke – Devět miliard božích jmen

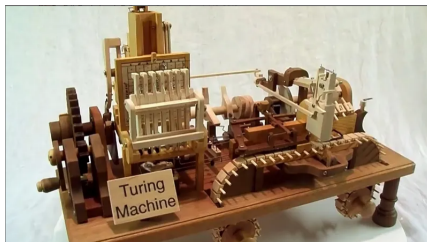
Historie vývoje počítače

- Abakus, logaritmické pravítko
- 1833 Charles Babbage
- 1936 Turingův stroj
- 1945 von Neumannovo schéma
- 1965 Moorův zákon



Alan Turing (1912-1954)

- definuje matematický model univerzálního stroje schopného vykonávat jakýkoliv popsatelný algoritmus
- definuje princip vypočitatelnosti - Turing ukázal, že stačí velmi jednoduchý stroj, aby spočítal cokoli, co lze spočítat.
- koncept symbolického zpracování
- Turingův test
- Enigma – prolomení německé šifry (podrobně v části Kódy a šifry)



Enigma: šifrovací stroj



Enigma: šifrovací stroj

- Patent Arthura Scherbia (1918); od 30. let ji používala německá armáda, letectvo i námořnictvo.
- Vypadá jako psací stroj: místo papíru se rozsvítí lampička se zašifrovaným písmenem.
- Uvnitř: 3 rotory (vybrané z 5), reflektor a propojovací deska s 10 kably.
- Po každém stisku se rotor pootočí, takže **stejné písmeno se pokaždé zašifruje jinak** (polyalfabetická šifra). Frekvenční analýza nefunguje.

Počet možných nastavení (armádní verze): $\approx 1,6 \times 10^{20}$

Kdybyste zkoušeli milion nastavení za sekundu, jak dlouho by to trvalo?

Enigma: šifrovací stroj

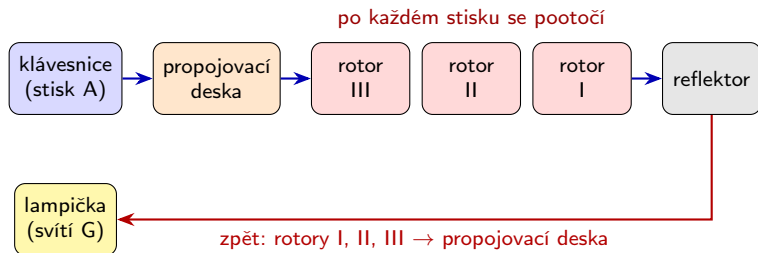
- Patent Arthura Scherbia (1918); od 30. let ji používala německá armáda, letectvo i námořnictvo.
- Vypadá jako psací stroj: místo papíru se rozsvítí lampička se zašifrovaným písmenem.
- Uvnitř: 3 rotory (vybrané z 5), reflektor a propojovací deska s 10 kably.
- Po každém stisku se rotor pootočí, takže **stejné písmeno se pokaždé zašifruje jinak** (polyalfabetická šifra). Frekvenční analýza nefunguje.

Počet možných nastavení (armádní verze): $\approx 1,6 \times 10^{20}$

Kdybyste zkoušeli milion nastavení za sekundu, jak dlouho by to trvalo?

$1,6 \times 10^{14}$ s ≈ 5 milionů let. A klíč se měnil každý den o půlnoci.

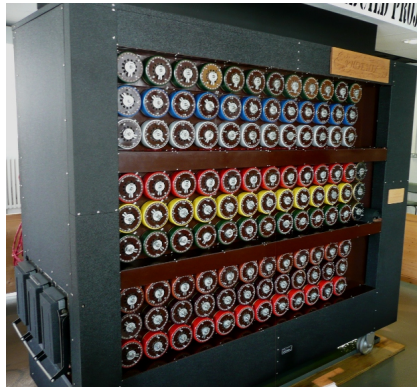
Enigma: cesta jednoho písmene



- Proud projde 9× přes převod písmen: deska, 3 rotory, reflektor, 3 rotory, deska.
- Díky reflektoru je šifra symetrická: stejné nastavení zprávu zašifruje i dešifruje.
- **Slabina:** díky reflektoru se písmeno **nikdy** nezašifruje samo na sebe.

Proč je tato drobnost pro luštitelé tak cenná?

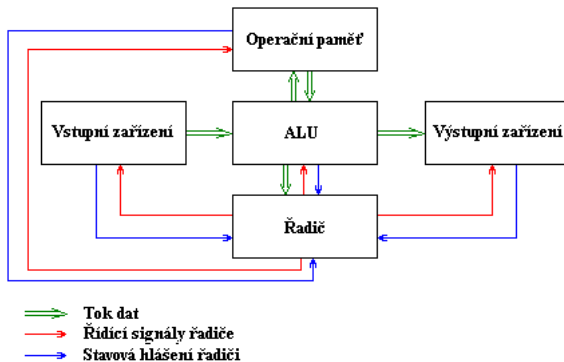
Alan Turing: portréty a otázky



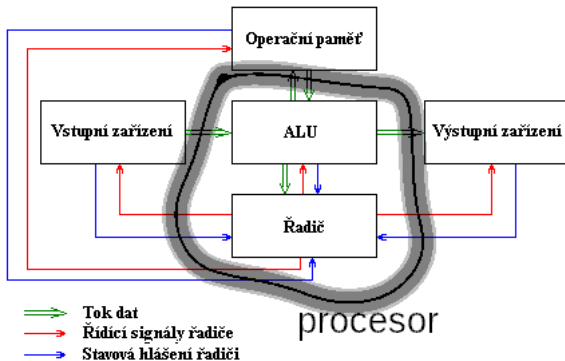
Je váš notebook „silnější“ než Turingův stroj?
Jak byste dnes poznali, že komunikujete s člověkem a ne s AI?

John von Neumann (1903-1957)

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

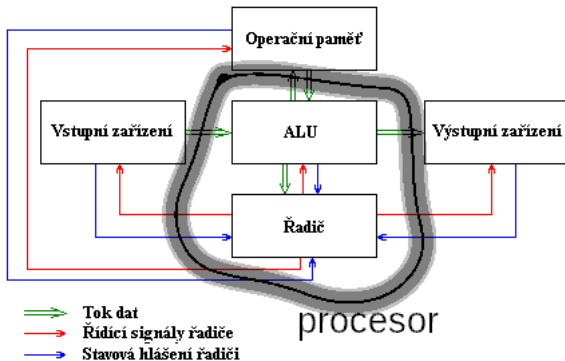


Von Neumannovo schéma



Kde na cestě spektra z disku na obrazovku je nejpomalejší krok?

Von Neumannovo schéma



Kde na cestě spektra z disku na obrazovku je nejpomalejší krok?

Latence: cache ≈ 1 ns; RAM ≈ 100 ns; SSD ≈ 100 μ s; HDD ≈ 10 ms; internet ≈ 100 ms

Von Neumannovo schéma

- Operační paměť : slouží k uchování zpracovávaného programu, zpracovávaných dat a výsledků výpočtu
- ALU - Arithmetic-logic Unit (aritmetickologická jednotka): jednotka provádějící veškeré aritmetické výpočty a logické operace. Obsahuje sčítačky, násobičky (pro aritmetické výpočty) a komparátory (pro porovnávání)
- Řadič: řídicí jednotka, která řídí činnost všech částí počítače. Toto řízení je prováděno pomocí řídicích signálů, které jsou zasílány jednotlivým modulům. Reakce na řídicí signály, stavy jednotlivých modulů jsou naopak zasílány zpět řadiči pomocí stavových hlášení
- Vstupní zařízení: zařízení určená pro vstup programu a dat.
- Výstupní zařízení: zařízení určená pro výstup výsledků, které program zpracoval

Paměťová hierarchie: RAM, cena, architektura

Proč se nevyplatí vše držet v RAM, když je to tak rychlé?

RAM je nestálé (volatilní) médium: K udržení informací potřebuje neustálý přísun elektrického proudu.

Okamžitý výpadek: Jakmile počítač vypneš, restartuješ nebo dojde k výpadku proudu, všechna data v RAM se během vteřiny nenávratně smažou.

Úloha SSD/HDD: Pro trvalé uložení fotek, operačního systému a dokumentů potřebuješ permanentní paměť (NVMe SSD nebo HDD), která data udrží i bez napájení.

Extrémní rozdíl v ceně a kapacitě

Fyzická omezení základních desek

Architektura počítačů je navržena jako **pyramida**. Rychlá a drahá RAM slouží jako krátkodobá pracovní plocha pro to, s čím procesor pracuje právě teď, zatímco levné a stálé SSD slouží jako obří digitální skladiště.

CPU × GPU: proč AI potřebuje grafické karty

- CPU = několik velmi schopných jader; GPU = tisíce jednoduchých jader
- CPU je jako hrstka geniálních profesorů: vyřeší cokoli, ale úlohy berou jednu po druhé
- GPU je jako armáda studentů s kalkulačkou: každý umí jen základní počty, ale tisíce jich počítají najednou
- Trénink neuronové sítě = násobení obrovských matic a to se dá rozdělit mezi tisíce jader

Co z vaší laboratorní práce by šlo „rozdat studentům“, tedy paralelizovat?

CPU × GPU: proč AI potřebuje grafické karty

- CPU = několik velmi schopných jader; GPU = tisíce jednoduchých jader
- CPU je jako hrstka geniálních profesorů: vyřeší cokoli, ale úlohy berou jednu po druhé
- GPU je jako armáda studentů s kalkulačkou: každý umí jen základní počty, ale tisíce jich počítají najednou
- Trénink neuronové sítě = násobení obrovských matic a to se dá rozdělit mezi tisíce jader

Co z vaší laboratorní práce by šlo „rozdat studentům“, tedy paralelizovat?

Porovnání 1 000 spekter s knihovnou: každé spektrum zvlášť → ideální pro GPU.

Amdahlův zákon: když 10 % úlohy paralelizovat nejde, celá úloha se zrychlí nejvýš 10×, ať máte jader kolik chcete.

CPU × GPU: paralelizace v chemii

Vlastnost	CPU (Central Processing Unit)	GPU (Graphics Processing Unit)
Architektura	Málotaktovaná, ale supervýkonná jádra (typicky 4 až 64).	Tisíce malých, jednoduchých jader (tisíce až desetitisíce).
Styl práce	Sekvenční (úlohy zpracovává postupně za sebou).	Masivně paralelní (tisíce úloh zpracovává najednou).
Silná stránka	Nízká latence, komplexní logika, větvení kódu.	Obrovská propustnost dat při opakovaných operacích.
Příklad v chemii	Kvantová chemie (CCSD(T)), výpočet trajektorie jednoho atomu.	Molekulární screening, klasická MD (GROMACS na tisících atomů).

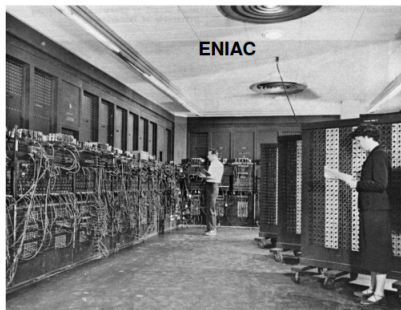
První generace 1944

elektronkové počítače, strojový kód, assembler autokód, dřené štítky, desítky skříní, operační rychlost op./s $10^2 - 10^4$, vnitřní paměť 1-2 KB

Elektronka



Kondenzátory



Druhá generace 1951

tranzistory, vyšší programovací jazyky FORTRAN, ALGOL, COBOL, do deseti skříní, operační rychlost op./s $10^4 - 10^5$, vnitřní paměť 16-32 KB

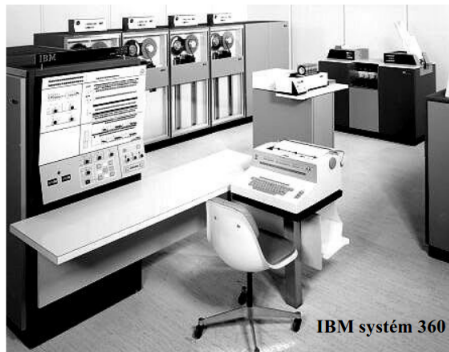
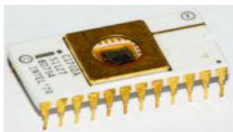
Tranzistory



Třetí generace 1965

integrovaný obvod, operační systémy a modernější programovací jazyky PASCAL, LISP, do pěti skříní, operační rychlost op./s 10^5 - 10^6 , vnitřní paměť 0,5-2 MB dále až 16MB

Integrovaný obvod



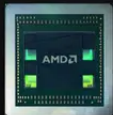
Čtvrtá generace

mikroprocesor, operační systémy a jazyky přizpůsobené uživatelům, jedna skříň, operační rychlost op./s 10^5 - 10^6 , vnitřní paměť 1-18 MB



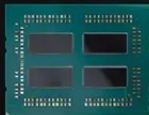
Nyní 3D technologie

AMD LEADERSHIP PACKAGING

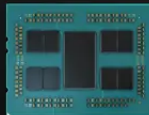


2.5D HBM

2015

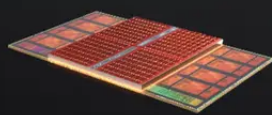
MULTICHIP
MODULE

2017



CHIPLETS

2019

3D CHIPLETS
(Chiplet + Advanced 3D Stacking)

2021

Led Industry in HBM, 2.5D & Chiplet Architecture | Aggressive Roadmap for Chiplet & 3D Integration

CPU × GPU × TPU × QPU: paralelizace v chemii

Vlastnost	CPU (Central Processing Unit)	GPU (Graphics Processing Unit)	TPU (Tensor Processing Unit)	QPU (Quantum Processing Unit)
Architektura	Málo, ale supervýkonná jádra (typicky 4 až 64)	Tisíce malých, jednoduchých jader (tisíce až desetitisíce)	Maticové jednotky specializované jen na násobení matic (Google)	Qubity (desítky až stovky), chlazené na tisíce kelvinů
Styl práce	Sekvenční (úlohy zpracovává postupně za sebou)	Masivně paralelní (tisíce úloh zpracovává najednou)	Paralelní násobení matic v nízké přesnosti	Superpozice a interference; výsledek je pravděpodobnostní, měření se opakuje
Silná stránka	Nízká latence, složitá logika, větvení kódu	Obrovská propustnost dat při opakovaných operacích	Nejvyšší výkon na watt při tréninku neuronových sítí	Do budoucna: simulace kvantových systémů, rozklad čísel (Shor)
Příklad v chemii	Kvantová chemie (CCSD(T)), výpočet trajektorie jednoho atomu	Molekulární screening, klasická MD (GROMACS na tisících atomů)	Trénink ML modelů (predikce struktury proteinů, neuronové potenciály)	VQE: základní stav malých molekul (H_2 , LiH); zatím demonstrace

Rok 2025

www.top500.org

Rank	Site	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	DOE/NNSA/LLNL United States	El Capitan - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS HPE	11,039,616	1,742.00	2,746.38	29,581
2	DOE/SC/Oak Ridge National Laboratory United States	Frontier - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS HPE	9,066,176	1,353.00	2,055.72	24,607
3	DOE/SC/Argonne National Laboratory United States	Aurora - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11 Intel	9,264,128	1,012.00	1,980.01	38,698
4	EuroHPC/FZJ Germany	JUPITER Booster - BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux EVIDEN	4,801,344	793.40	930.00	13,088
5	Microsoft Azure United States	Eagle - Microsoft NDv5, Xeon Platinum 8480C 48C 2.6GHz, NVIDIA	2,073,600	561.20	846.84	

El Capitan



IT4 Ostrava

<https://www.it4i.cz/>

Rank	site	cores	Rmax(Tf/s)	Rpeak(Tf/s)	power(kW)
1	El Capitan	11,039,616	1,742,000.00	2,746,376.09	29,580.98
195	Karol1naGPU	71,424	6,752.00	9,080.18	310.99
421	Karol1naCPU	92,160	2,837.40	3,833.86	502.70
...	PC - Mira	24	~1	~1	0.250
	Mobile s25	8			

Hardware - stručné opakování

Hardware

procesor - Intel a AMD + chladič a větrák

základní deska - s instalovaným Basic Input Output Systémem (BIOS)

sběrnice, řadiče

zdroj

operační paměť

pevné disky - HDD a Solid state

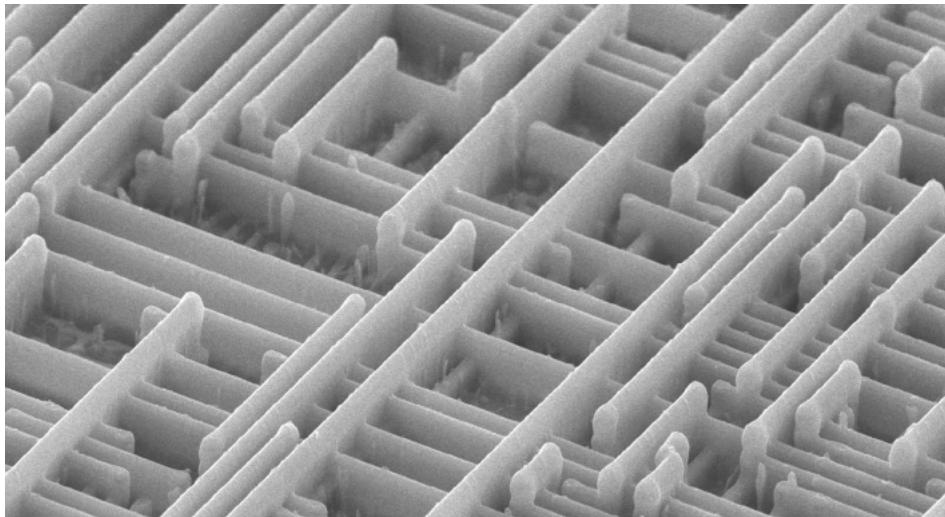
záznamová média

grafická karta

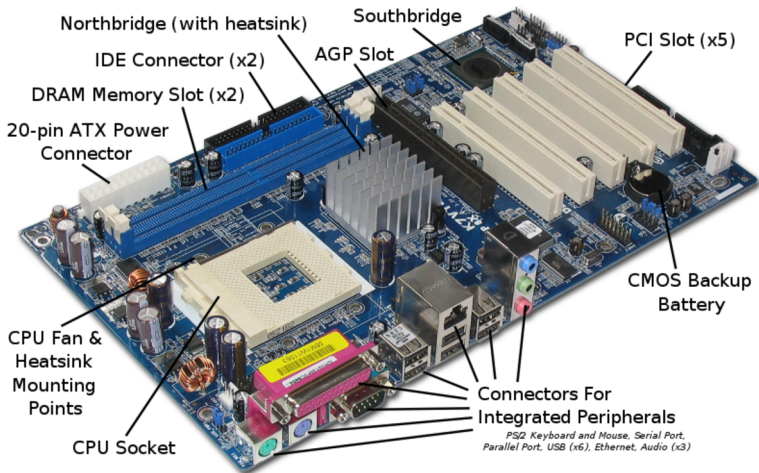
síťová karta

Processor

<https://www.youtube.com/watch?v=bor0qLifjz4>



motherboard



Úložná media: od děrného štítku po SSD

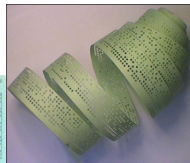
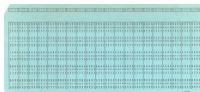
- neelektronické - papyrus, pergamen, papír, kámen, tabule, fotografická deska
- magnetické - disketa, pevný disk, videokazeta,...
- optické - CD, DVD, Blu-ray, ...
- elektronické - USB flash, SSD, paměťové karty, ...

1805 Jacquardův děrný štítek;

1956 IBM RAMAC (5 MB);

1984 disketa 1,44 MB;

dnes microSD s kapacitou TB



1957 první tvrdý disk - IBM



HD



HD x SSD

hard disk se uvnitř otáčí

pevný disk



HDD × SSD: forenzní pohled (1/2)

Vlastnost / aspekt	Klasický pevný disk (HDD)	Polovodičový disk (SSD)
Fyzická architektura	Mechanické rotující plotny a magnetická čtecí hlava.	Flash paměť (NAND čipy), žádné pohyblivé součásti.
Princip smazání souboru	Smaže se pouze záznam v souborovém systému (ukazatel). Data fyzicky zůstávají na sektorech, dokud nejsou přepsána novými daty .	Smazání spustí interní procesy disku. Data jsou aktivně přeskupována a připravována na kompletní výmaz čipů.
Interní optimalizace	Žádná automatická destrukce dat na pozadí ze strany disku.	TRIM a Wear Leveling (vyrovnávání opotřebení). Disk sám na pozadí přesouvá a trvale maže bloky dat.

HDD × SSD: forenzní pohled (2/2)

Vlastnost / aspekt	Klasický pevný disk (HDD)	Polovodičový disk (SSD)
Šance na obnovu smazaných dat	Vysoká. Data lze často kompletně obnovit (carving) i dlouho po smazání, pokud se na disk intenzivně nezapisovalo.	Nízká až nulová. Kvůli příkazu TRIM může ke kompletnímu fyzickému vymazání bloku dojít během několika sekund až minut od smazání uživatelem.
Klíčový forenzní důsledek	Čas není kritický faktor. Disk stačí zajistit, data na plotnách bez napájení i při běžícím OS drží relativně stabilně.	Čas je kritický (Race against TRIM). Zapnutý disk aktivně ničí důkazy na pozadí.
Postup zajištění důkazů	Standardní pořízení bitové kopie přes hardwarový write blocker (blokátor zápisu).	Okamžité odpojení od napájení. Při vytváření image je write blocker kriticky nutný , ale nezabrání internímu TRIMování, pokud je disk napájen.

Kam uložit 10 TB měření? Požadavky na forenzní úložiště

kam uložíme 10 TB a jak dlouho potrvá přenos.

Přenos: 10 TB přes 1 Gbit/s $\approx$ 22 h; přes 10 Gbit/s $\approx$ 2,2 h; přes 100 Mbit/s $\approx$ 9 dní; osobní převoz disku může být rychlejší

1. **Neměnnost** (Integrita): Data se po zapsání nesmí změnit ani o jediný bit. Používají se WORM úložiště (Write Once, Read Many), hardwarové snapshoty chráněné proti ransomware a striktní kontrolní soubory (SHA-256 / SHA-3).
2. **Dohledatelnost** (Audit): Musí existovat digitální Chain of Custody (řetězec dokumentace). Každé zobrazení, analýza nebo přístup k souboru se loguje. Musí být jasné, kdo s daty pracoval v roce 2026 i v roce 2036.
3. **Dlouhodobé zálohování** (Pravidlo 3-2-1): Data musí přežít dekády.
 - 3 kopie dat (1 primární + 2 zálohy).
 - 2 různá média (např. rychlé SSD pole pro práci + pásková knihovna LTO pro archiv).
 - 1 kopie mimo laboratoř (geograficky oddělené zabezpečené úložiště pro případ požáru).

Velká uložistiště

https://www.youtube.com/watch?v=281z-B0F_Uc

Odhadem vznikne každý den ≈ 400 milionů TB (0,4 ZB) dat

Google: desítky datacenter po celém světě, každé stojí miliardy dolarů

Google účet: 15 GB zdarma, cca 2×10^9 účtů



Velká uložistě



Velká uložistě



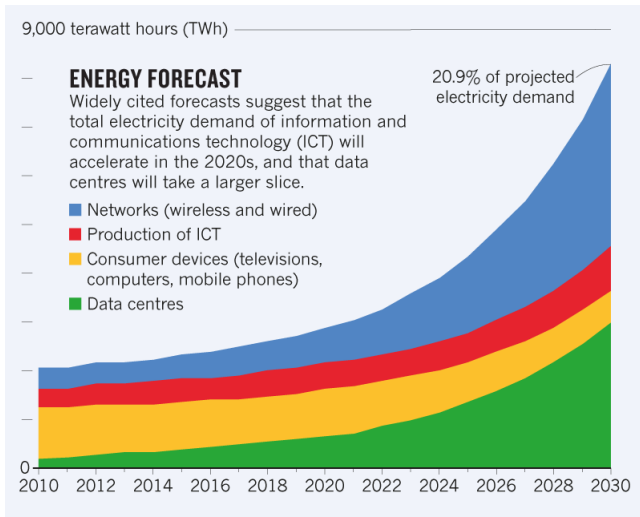
Velká uložistě

Datacentra Google: 2020 cca 15 TWh → 2024 cca 31 TWh → 2025 přes 42 TWh za rok

Temelín vyrobí cca 16 TWh za rok → Google dnes potřebuje přibližně 2,5 Temelína



energie



Energie a cena: nejmenší dostatečný nástroj

Úroveň hardwaru	Spotřeba (energie)	Finanční náročnost	Ideální chemická úloha
1. Notebook	~60–150 W (jako silnější žárovka)	Nízká (máte ho v batohu)	Příprava a analýza: kreslení molekul, psaní skriptů, vizualizace (PyMOL, VMD). Výpočty NE, hrozí přehřívání.
2. Klasické PC (s RTX 4090/5090)	~400–850 W (jako mikrovlnka)	Střední (~100 tisíc Kč)	Lokální král: virtuální screening (milion molekul), standardní MD simulace proteinů. Skvělá efektivita na watt.
3. Laboratorní cluster	~2–4 kW (jako zapnutá sauna)	Vysoká (stovky tisíc + chlazení)	Týmová práce: nonstop běžící simulace (GROMACS, AMBER) pro celou diplomovou/doktorskou skupinu.
4. Superpočítač (HPC)	Stovky kW až MW (jako menší město)	Extrémní (miliony Kč / granty)	Těžká váha: kvantová chemie (ORCA, Gaussian), obří systémy (viry, membrány), AIMD.

Odhad z hlavy (Fermiho úloha)

Každý den vznikne ≈ 400 milionů TB dat. Kolik 20TB disků by bylo potřeba, kdybychom chtěli všechno uložit?

Odhad z hlavy (Fermiho úloha)

Každý den vznikne ≈ 400 milionů TB dat. Kolik 20TB disků by bylo potřeba, kdybychom chtěli všechno uložit?

$4 \times 10^8 \text{ TB} / 20 \text{ TB} = 2 \times 10^7$, tedy ≈ 20 milionů disků **denně**.

Co z toho plyne?

Odhad z hlavy (Fermiho úloha)

Každý den vznikne ≈ 400 milionů TB dat. Kolik 20TB disků by bylo potřeba, kdybychom chtěli všechno uložit?

$4 \times 10^8 \text{ TB} / 20 \text{ TB} = 2 \times 10^7$, tedy ≈ 20 milionů disků **denně**.

Co z toho plyne?

- Rozhodnout, **co** uložit (a s jakými metadaty), je odborná, ne technická otázka.
- Ve forenzní praxi nesmíme ztratit ani bit důkazu, proto 3-2-1 a kontrolní součty.
- Vývoj nových technologií na ukládání (silico,DNA)

Proč spotřeba datacenter od roku 2020 vzrostla skoro trojnásobně?

Odhad z hlavy (Fermiho úloha)

Každý den vznikne ≈ 400 milionů TB dat. Kolik 20TB disků by bylo potřeba, kdybychom chtěli všechno uložit?

$4 \times 10^8 \text{ TB} / 20 \text{ TB} = 2 \times 10^7$, tedy ≈ 20 milionů disků **denně**.

Co z toho plyne?

- Rozhodnout, **co** uložit (a s jakými metadaty), je odborná, ne technická otázka.
- Ve forenzní praxi nesmíme ztratit ani bit důkazu, proto 3-2-1 a kontrolní součty.
- Vývoj nových technologií na ukládání (silico,DNA)

Proč spotřeba datacenter od roku 2020 vzrostla skoro trojnásobně?

Trénink a provoz AI modelů na GPU.

Nebojme se programovat

Animation vs. Coding

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



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

Nebojme se programovat

- 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

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

SOP v laboratoři = algoritmus pro člověka

Standardní operační postup (SOP) v analytické laboratoři má všechno, co má algoritmus:

- vstup: vzorek, činidlo, kalibrační standardy
- kroky v pevném pořadí: navaž, rozpust', zřed', změř
- rozhodování: „je-li signál nad kalibračním rozsahem, zřed' vzorek 10× a měř znovu“
- výstup: koncentrace $\pm$ nejistota, protokol

Co se stane, když v SOP chybí větev „je-li signál nad rozsahem. . .“?

Každý analytik se zachová jinak, výsledky nejsou opakovatelné a u soudu neobstojí.

Počítač je ten nejpečlivější a nejhoupější analytik: udělá přesně to, co je napsané.

Algoritmus

formálně zapsaný postup pro řešení daného problému

– jde o posloupnost elementárních kroků

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

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

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

Algoritmus

Napište algoritmus pro výrobu bábovky.

Bábovka: test jednoznačnosti

Vyměňte si recept se sousedem. Najděte v něm všechna místa, kde by se robot zastavil.

- „trochu mléka“: kolik je trochu?
- „peč do zlatova“: jak robot pozná zlatovou?
- „vyšlehej“: jak dlouho? čím?
- „nemáš-li máslo, ...“: je pokrytá každá alternativa?

Co lidský kuchař doplní ze zkušenosti, musí programátor zapsat explicitně.

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

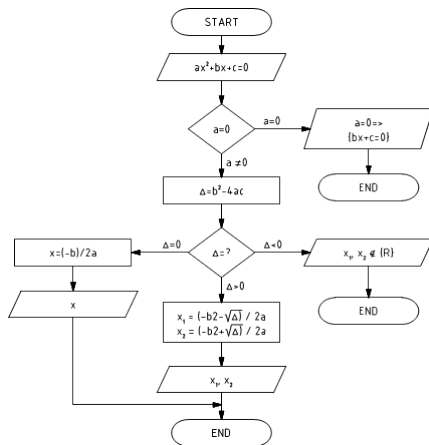
b) zapsat jej v nějakém programovacím jazyce

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 paměťově méně náročné

Je recept algoritmus?

Které vlastnosti algoritmu recept na bábovku splňuje a které ne?

- **rezultativnost**: ano, vznikne bábovka (snad)
- **konečnost**: ano, recept má konečný počet kroků
- **determinovanost**: spíš ne, „do zlatova“ není jednoznačné
- **elementárnost**: záleží na „procesoru“ (zkušený cukrář $\times$ robot)
- **hromadnost**: ano, funguje pro libovolný počet porcí (vynásob suroviny)

A co algoritmus „Dokud nenajdeš největší prvočíslo, zkoušej další čísla“?

Není konečný: prvočísel je nekonečně mnoho (Eukleides, asi 300 př. n. l.).

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

tvorí součást kryptografických protokolů - šifra RSA

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.

Eukleides krok za krokem

$\text{NSD}(1071, 462) = ?$

① $1071 = 2 \cdot 462 + 147$

② $462 = 3 \cdot 147 + 21$

③ $147 = 7 \cdot 21 + 0 \rightarrow$ zbytek je 0, takže **NSD = 21**

Totéž v Pythonu:

```
def nsd(u, w):
    while w != 0:
        u, w = w, u % w
    return u
print(nsd(1071, 462))    # 21
```

Proč je Eukleides lepší než vypsát všechny dělitele? Zkuste to pro dvě čísla o 600 cifrách.

Eukleides krok za krokem

$\text{NSD}(1071, 462) = ?$

① $1071 = 2 \cdot 462 + 147$

② $462 = 3 \cdot 147 + 21$

③ $147 = 7 \cdot 21 + 0 \rightarrow$ zbytek je 0, takže **NSD = 21**

Totéž v Pythonu:

```
def nsd(u, w):  
    while w != 0:  
        u, w = w, u % w  
    return u  
print(nsd(1071, 462))    # 21
```

Proč je Eukleides lepší než vypsát všechny dělitele? Zkuste to pro dvě čísla o 600 cifrách.

Eukleides potřebuje jen několik tisíc kroků, vypsát všechny dělitele by nestihl ani vesmír. Právě toho využívá šifra RSA.

Eukleidův algoritmus v různých jazycích

<https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended/>

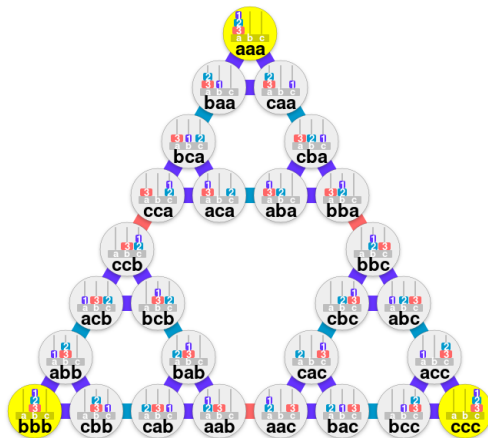
Hanojské věže

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

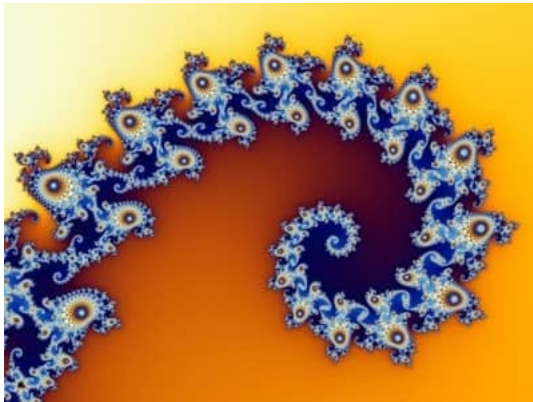
https://en.wikipedia.org/wiki/Tower_of_Hanoi

Hanojské věže

Grafické řešení vede na fraktály.



Příklad fraktálu



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

Hanojské věže: kolik to trvá?

Kolik tahů potřebujeme pro 1, 2, 3, 4 disky? Vidíte pravidlo?

1, 3, 7, 15, ..., obecně $2^n - 1$ tahů

Legenda: mniši v chrámu přenášejí 64 zlatých disků, jeden tah za den. Až skončí, nastane konec světa. Kdy to bude?

$2^{64} - 1 \approx 1,8 \times 10^{19}$ s (rychlejší) $\approx$ 585 miliard let (vesmír je starý asi 14 miliard let)

Algoritmus může být správný a krátký, a přesto prakticky nepoužitelný.

Složitost algoritmů probereme příště.

Srovnejte s povídkou A. C. Clarka Devět miliard božích jmen.

Výroková logika

Logické výrazy a výroková logika

$(x < 0 \parallel y > 0 \&\& y < 1) \&\& !(z < 0)$

"x je záporné nebo y je mezi 0 a 1, a současně z není záporné"

$p \&\& q$ je konjunkce výroků, čteme „výrok p a (zároveň) výrok q“

$p \parallel q$ je disjunkce výroků, čteme „výrok p nebo výrok q“

$p \Rightarrow q$ je implikace, čteme „jestliže výrok p, pak výrok q“

$p \Leftrightarrow q$ je ekvivalence, čteme „výrok p právě tehdy když výrok q“

Moje bakalářka jako datový projekt

Příprava na závěrečnou prezentaci (5–10 min na konferenci):

- 1 Jaká data vznikají (typ, formát)?
- 2 Kolik (velikost/rok)?
- 3 Jaká metadata?
- 4 Kam uložím a jak zálohuji?
- 5 Jak ověřím integritu (hash) a kdo smí data číst (šifrování, podpis)?
- 6 Jak by mohl vypadat algoritmus mé bakalářky?
- 7 Má všechny důležité vlastnosti?
- 8 Jaké datové typy budou mít hlavní veličiny?
- 9 Co by mohlo ML predikovat?

Skripta, stránky a materiály

http://projekty.fs.vsb.cz/463/edubase/VY_01_042/Z%C3%A1klady%20informatiky.pdf
<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.bigyzt.cz/shared/clanky/2893/ICT-Pripravy/IS4-Pripravy_informace.pdf
<http://www.ivt.mzf.cz/informatika-v-kostce/zakladni-pojmy/kodovani-a-sifrovani/>
<https://www.itnetwork.cz/hardware-pc/principy-pocitacu/reprezentace-cisel-v-pocitaci>
<https://www.napocitaci.cz/33/symetricke-a-asymetricke-sifrovani-uniqueidgOkE4NvrWuNY54vrLeM677jX7sp3Lu-ZpLpGVMy1prA/>
https://en.wikipedia.org/wiki/Charles_Babbage
<https://en.wikipedia.org/wiki/FLOPS>
<https://explodingtopics.com/blog/data-generated-per-day>
<https://www.google.com/about/datacenters/>
<https://www.cryptomuseum.com/crypto/enigma/>
<https://cs.wikipedia.org/wiki/RSA>
<https://www.baeldung.com/cs/euclid-time-complexity>
<https://www.promotic.eu/cz/pmdoc/Subsystems/Db/MySQL/DataTypes.htm>