Základy forenzních databází

Tereza Uhlíková

verze 2.0

Kdo jsem

Tereza Uhlíková

Ústav analytické chemie

skupina teoretické spektroskopie

místnost A235

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

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 uložiště
- Architektura databází

O čem budeme mluvit dnes

- základy programování
  - algoritmus
  - strukturované programovací jazyky
- datové typy

Nebojme se programovat

Nebojme se programovat

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

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

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

Algoritmus

formálně zapsaný postup pro řešení daného problému
– jde o posloupnost elementárních kroků

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

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

Napište algoritmus pro výrobu bábovky.

Algoritmus a Program

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:

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)

Eukleidův algoritmus

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

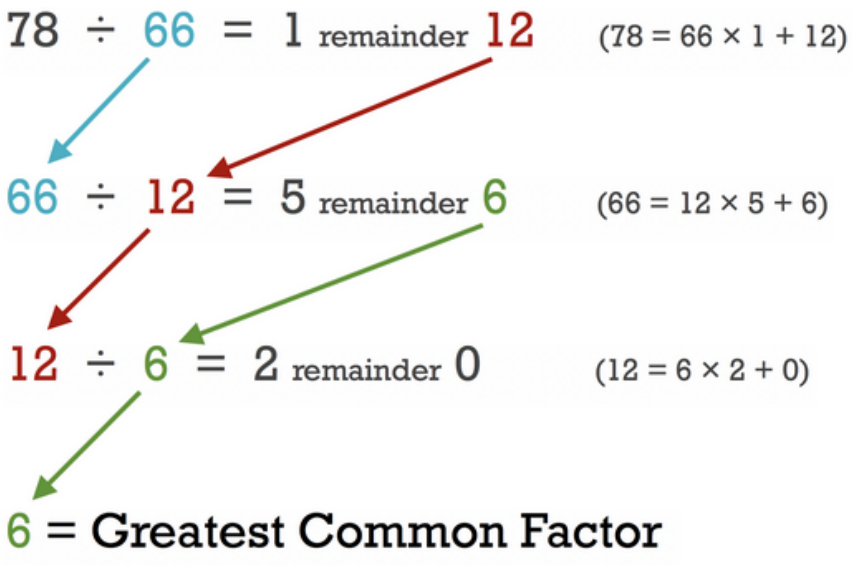
Největší společný dělitel (NSD) dvou celých čísel je největší číslo, které beze zbytku dělí obě čísla. Příklady: NSD(18, 24) = 6, NSD(30, 85, 90) = 5.
Použití:
redukci zlomků na jejich nejjednodušší formu
tvoří součást kryptografických protokolů - šifra RSA

Eukleidův algoritmus

- vypíšeme všechny dělitele
- pomocí prvočíselného rozkladu
- Euklides

Eukleidův algoritmus

- vypíšeme všechny dělitele
- pomocí prvočíselného rozkladu
- Euklides



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/

Algoritmus pro výpočet kořenů kv. rovnice

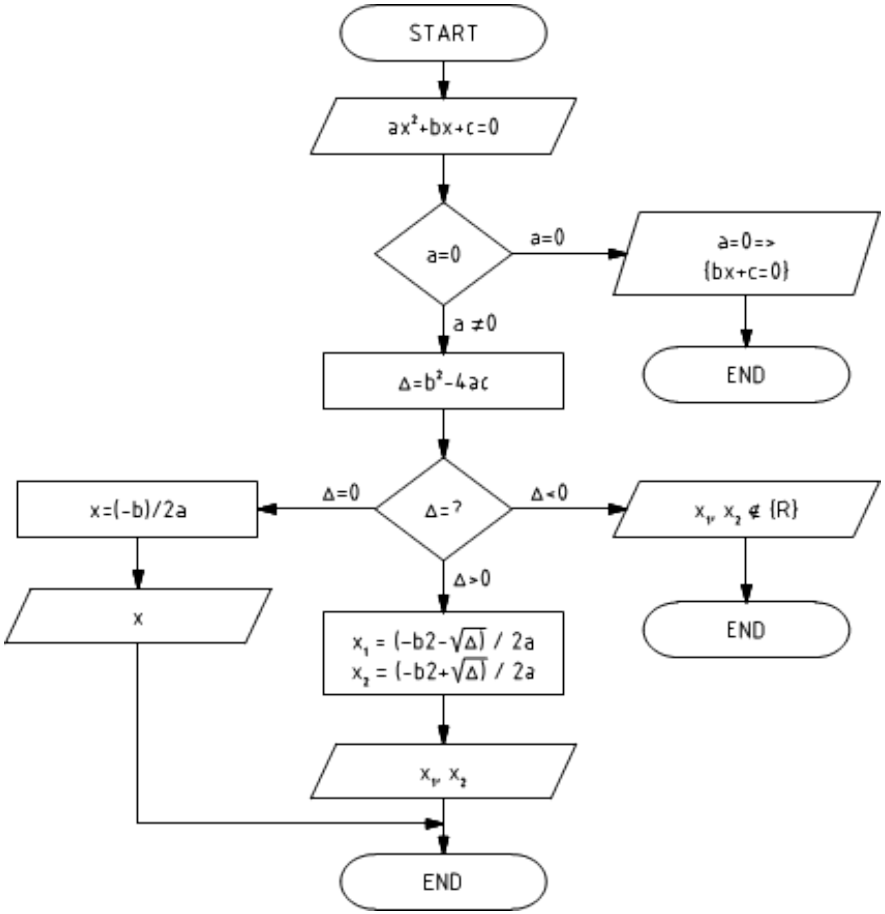
Algoritmus pro výpočet kořenů kv. rovnice

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

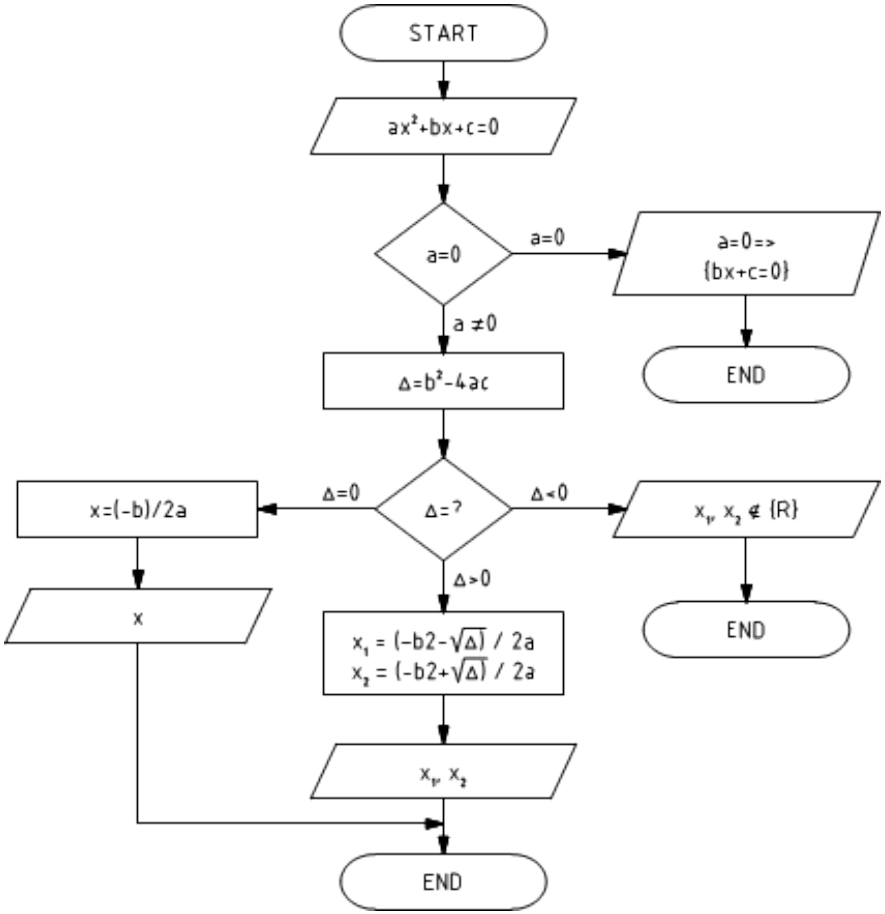
Zápis algoritmů

vývojové diagramy (grafický zápis) - orientovaný graf



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

Základní komponenty algoritmu

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

Vlastnosti algoritmu

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

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

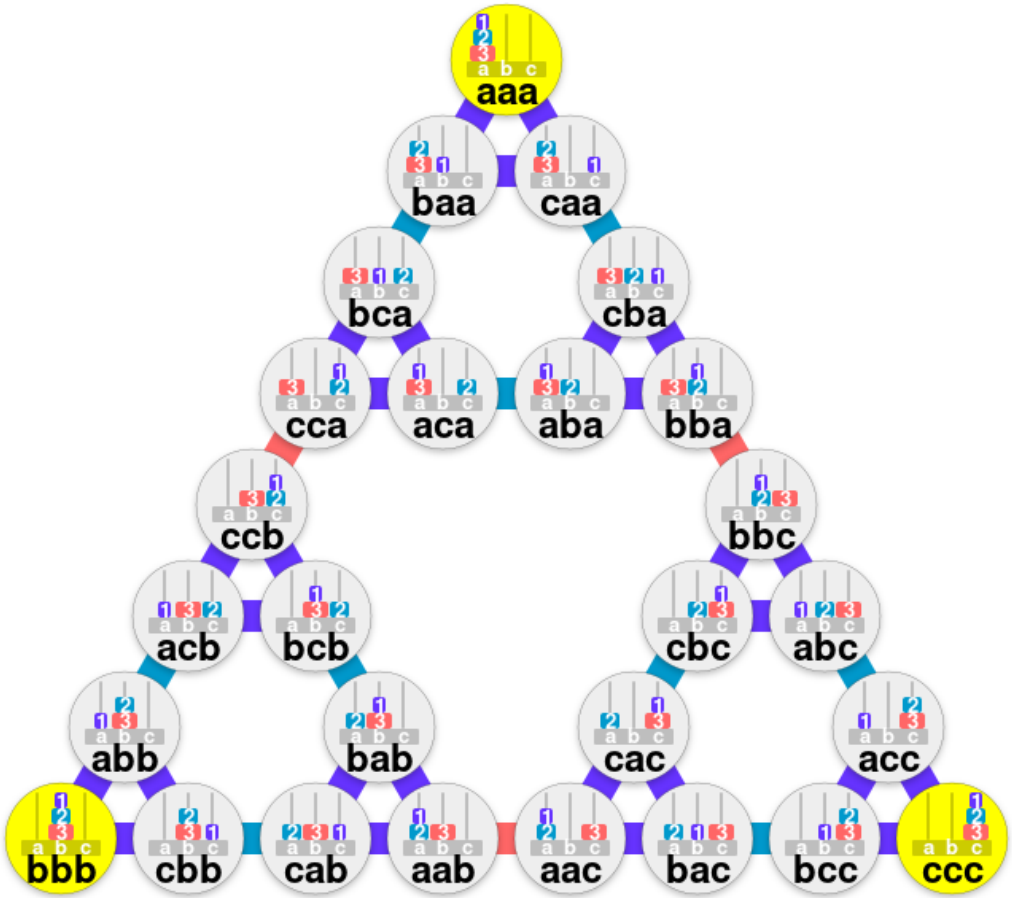
Hanojské věže

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.



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

Datové typy

- charakterizuje vlastnost proměnné

Numerické typy:

- celočíselný (INTEGER)

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)

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 řetzce:

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 řetzce:

- 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 řetzce:

Datové typy

Bitové řetzce:

- BIT(n): n udává počet bitů

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

Datové typy

Bitové řetzce:

- BIT(n): n udává počet bitů

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

- DATE
- TIME
- TIMESTAMP

Logické typy:

Datové typy

Bitové řetzce:

- 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

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
https://www.mgplzen.cz/download/ivt/ivt_programovani.pdf