Matematické algoritmy

prof. RNDr. Miroslav Vlček, DrSc.

vlcek@fd.cvut.cz

čtvrtek v místnosti 209

$8^{00} - 9^{30}$

a

$9^{45} - 11^{15}$

11. října 2007

Literatura k dalšímu studiu

1 Notes for the course advanced algorithms by Johan Hastad, Royal Institute of Technology, SE-100 44 Stockholm, Sweden

2 Algorithms and Complexity by Herbert S. Wilf, University of Pensylvania, Philadelphia, PA 1904-6395

3 Lecture Notes on Algorithm Analysis and Computational Complexity by Ian Parberry, Department of Computer Science, University of North Texas, 2001

4 Design and Analysis of Computer Algorithms by David M.Mount, University of Maryland, College Park, MD, 20741

Co víte o algoritmech?

1. Typy algoritmů

2. Co potřebujete znát ?

3. Kam až můžeme dojít ?

Algoritmus - vlastnosti

Algoritmem rozumíme postup, podle kterého se z dat vstupních $x(n)$ vygenerují data výstupní $y(n)$.

Algoritmus je předpis na řešení "nějakého" problému. Každý algoritmus musí mít následující vlastnosti:

1. **Konečnost** znamená, že výpočet se ukončí v "rozumně" konečném čase.

2. **Hromadnost** znamená, že algoritmus není sestrojen pouze na jediný problém, ale na celou řadu problémů.

3. **Jednoznačnost** znamená, že přechod do následujícího stavu algoritmu je jednoznačně určen výsledkem stavu předchozího.

Algoritmus - komenář k vlastnostem

1. **Konečnost** - přepověď počasí na zítra dosažená výpočtem o den později nemá význam

2. **Hromadnost** - program pro výpočet odmocniny pracuje nad množinou čísel, není konstuován pro každé číslo zvlášť

3. **Jednoznačnost** - každý algoritmus je složen z kroků, které na sebe vzájemně navazují. Každý krok je charakterizován jako přechod z jednoho stavu do jiného. Každý stav algoritmu je určen zpracovávanými daty a na tom, jak data v jednotlivých stavech vypadají. Je tedy pevně určeno, který krok bude následovat.

Příklady algoritmů - Euklidův algoritmus

Metoda nalezení největšího společného dělitele (NSP≡ GCD Greatest Common Divisor) byla známá již ve starověkém Řecku před 2000 lety. Spočívá v jednoduchém pozorování, že největší společný dělitel dvou čísel $p > q$ je shodný s největším společným dělitelem čísel $p - q, q$.

Tento poznatek již stačí k sestavení algorimu.

given p, q
body
    do if $p < q$
        $r = p;\ p = q;\ q = r;$
    end
        $p = p - q;$
    repeat until $p = 0$
result
 gcd = q

Kód programu pro vyhledání největšího společného dělitele dvou čísel v prostředí MATLAB

```
function nej=nej(u,v)
    while u > 0,
        if u < v
            var=u;
            u=v;
            v=var;
        end
    u=u-v;
    end
    nej=v;
```

Kód programu pro vyhledání největšího společného dělitele dvou čísel v jazyce C

```
#include <stdio.h>
int gcd(int u,int v)
{
    int t;
    while(u > 0)
    {  if(u < v)
           { t = u; u = v; v = t; }
    u=u-v;
    }
    return v;
}
```

Euklidův algoritmus největšího společného dělitele

- **dělitelnost čísel**
  Říkáme, že dvě přirozená čísla p a q jsou kongruentní podle modulu n, jestliže p i q při dělení číslem n dávají stejný zbytek nebo, že jejich rozdíl $p-q$ je násobkem čísla n. Píšeme

  $$p = q \,(\mathrm{mod}\ n).$$

  Například

  $$38 = 14 \,(\mathrm{mod}\ 12),$$

  protože $38-14 = 24$ a to je násobkem 12. Můžeme se také ptát, jaký zbytek poskytuje příslušné modulární dělení, a píšeme například

  $$38 \equiv 2 \bmod 12, \text{ a } 14 \equiv 2 \bmod 12.$$

Počítání, které se týká dělení se zbytkem se nazývá “Modulární aritmetika”. Ta byla poprvé studována německým matematikem Karlem Friedrichem Gaussem (1777-1855) v roce 1801.

- **prvočísla**

  Máme určit, zda dané číslo N (dostatečně velké, např. o 100 cifrách) je prvočíslem? Předpokládejme, že délka tohoto čísla je n bitů, takže $N \approx 2^n$.

  Číslo N může být označeno za provočíslo, jestliže prozkoumáme podíly $\frac{N}{q}$, kde $q \simeq \sqrt{N}$. Protože platí $\sqrt{N} = \sqrt{2^n}$ bude počet dělení úměrný času $O(2^{n/2})$, což není příliš výhodné. Typicky

$$
\begin{aligned}
N &= 29 \\
\sqrt{N} &> 5, 4, 3, 2 \\
29 &\equiv 4 \bmod 5, \\
29 &\equiv 1 \bmod 4, \\
29 &\equiv 2 \bmod 3, \\
29 &\equiv 1 \bmod 2,
\end{aligned}
$$

Chytrý nápad ...

Malá Fermatova věta: Jestliže p je prvočíslo, které není dělitelem a, pak

$$a^{p-1} \equiv 1 \bmod p.$$

Malá Fermatova věta se používá k testování velkých prvočísel.

Pierre de Fermat 1601-1665

- Algebraická teorie **faktorizace čísel** tvoří základ **kryptologie**

  6127049569075942128162558427465565726086846076031984148279798226031515626369211302403342844679615718899328182959467198786178804369420568835255125831938299094831810499556245105899126965734290671274428791298474527796793829737067915624465318975470874259865885243634437323405729823083965762652694544353738979426311879135265107590764939899242733126274345754163996323330034278445641825406211650726467719980 59

  Can you factor the above 402 digit number? It is composed of two large prime numbers. The first correct factorisation sent to **webmaster@mathematik.com** wins 100 US Dollars.

Příklady algoritmů - numerický výpočet odmocniny

$$y(n) = \frac{1}{2}\left[y(n-1) + \frac{x(n-1)}{y(n-1)}\right] . \qquad (1)$$

Odmocnina z čísla 10 je s přesností na 10 desetinných míst rovna

$$\sqrt{10} = 3.16227766017.$$

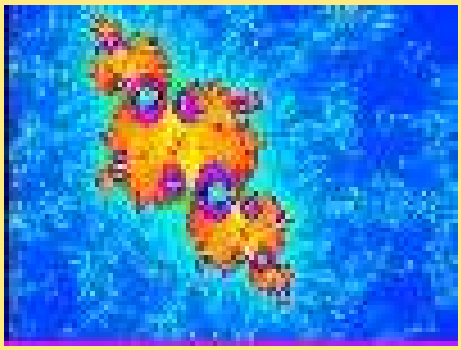
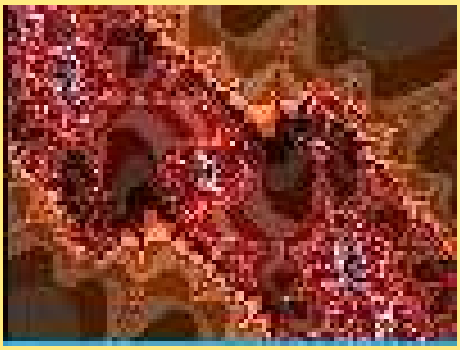
Pro

$$x(n-1) \equiv x(0) = 10$$

dostáváme postupně

$y(1) = 3$	$y^2(1) = 9$
$y(2) = 3.165$	$y^2(2) = 10.017225$
$y(3) = 3.162278$	$y^2(3) = 10.0000021493$
$y(4) = 3.1622776601$	$y^2(4) = 9.9999999996$
$\vdots$	

- **numerické řešení algebraických rovnic, diferenciálních rovnic a speciálních funkcí**

- **metody konečných prvků**
  řešení složitých parciálních diferenciálních rovnic s praktickými aplikacemi

- **jinak těžko řešitelné úlohy**
  Navier Stokesovy rovnice, nelineární parciální diferenciální rovnice

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\nabla)\mathbf{u} = \mathbf{f} - \nabla p + \nu \triangle \mathbf{u}$$

kde $\mathbf{u}$ a $\mathbf{f}$ jsou vektorové funkce rychlosti a síly, p je tlak a ν je úměrná viskozitě kapaliny.

$$\frac{\partial u_x}{\partial t} + u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_y}{\partial y} + u_z \frac{\partial u_z}{\partial z} =$$

$$= f_x(x,y,z,t) - \frac{\partial p}{\partial x} + \nu \left[\frac{\partial^2 u_x}{\partial x^2} + \frac{\partial^2 u_x}{\partial y^2} + \frac{\partial^2 u_x}{\partial z^2} \right]$$

Co potřebujete znát ?

- základy algebry a matematické analýzy
- základy numerické matematiky
- diferenční rovnice a jejich řešení
- základy strukturovaného programování
- aktivní znalost alespoň jednoho programovacího jazyka (Visual Basic, Pascal, C, C++) nebo prostředí MATLAB či MAPLE

Kam až můžeme dojít ?

- Objevit krásu některých algoritmů.
- Pochopit třeba numerické základy kryptologie.
- Nebát se inženýrských úloh, které vyžadují algoritmizaci.

Kam až můžeme dojít ? - cont.

- Pochopit rychlé algoritmy s aplikcemi v reálném světě

Rychlá Furierova transformace - analýza EEG signálu