

Algoritmus

Algoritmus je schematický postup pro řešení určitého druhu problémů, který je prováděn pomocí konečného množství přesně definovaných kroků. Ačkoliv se dnes tento pojem používá především v informatice a přírodních vědách obecně, tak je jeho působnost daleko širší (kuchyňské recepty, návody a postupy...). Samotné slovo *algoritmus* pochází ze jména perského matematika 9. století *Abu Jafar Muhammada ibn Mūsā al-Chwārizmīho*, který ve svých dílech položil základy algebry (arabské číslice, řešení lineárních a kvadratických rovnic).

Vlastnosti algoritmů

1. **Konečnost** – algoritmus má konečné množství kroků.
2. **Určitost** – všechny kroky algoritmu jsou přesně definovány.
3. **Korektnost** – algoritmus skončí pro libovolná (korektní) data správným výsledkem v konečném množství kroků.
4. **Obecnost** – algoritmus řeší všechny úlohy daného typu.

Dělení algoritmů

Rekurzivní a iterativní algoritmy

Iterativní algoritmus je takový, který spočívá v opakování určité své části (bloku). Rekurzivní algoritmus naproti tomu opakuje kód prostřednictvím volání sebe sama (obvykle na podproblémech menší velikosti). Každý rekurzivní algoritmus lze převést do iterativní podoby. Samotný převod často řeší automaticky kompilátor nebo virtuální stroj daného programovacího jazyka.

Výhoda rekurzivních algoritmů je v jejich snadno čitelném a kompaktním zápisu. Nevýhodou je potřeba dodatečných systémových prostředků pro udržení jednotlivých rekurzivních volání.

- **Iterativní** – *bubble sort*, *insertion sort*
- **Rekurzivní** – *merge sort*, *quicksort*

Deterministické a nedeterministické algoritmy

Deterministický je takový algoritmus, který má v každém svém kroku právě jednu možnost, jak pokračovat. Nedeterministický jich má více. Příkladem může být *deterministický a nedeterministický automat*.

Sériové, paralelní a distribuované algoritmy

Sériový algoritmus vykonává všechny kroky v sérii (jeden po druhém), paralelní algoritmus tyto kroky vykonává zároveň (ve více vláknech) a distribuovaný algoritmus kroky vykonává zároveň na více strojích.

Asymptotická složitost algoritmu

Asymptotická složitost algoritmu charakterizuje počet provedených operací v závislosti na velikosti dat. Například pokud procházíme pole, pak složitost bude lineární (na každý prvek připadá konstantní množství operací), pokud jej ovšem řadíme například *bubble sortem*, pak složitost bude kvadratická (na n prvků bude připadat n^2 operací).

Třída složitosti

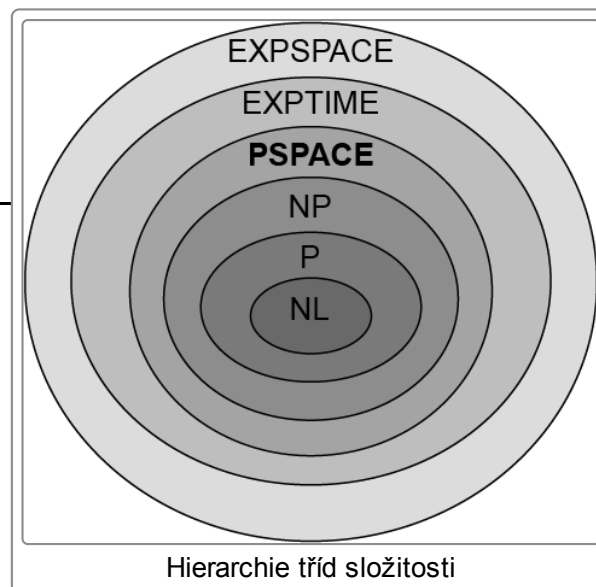
Třída složitosti stanovuje obtížnost rozhodnutelnosti daného problému na *Turingově stroji*.

- **Třída P** – obsahuje problémy rozhodnutelné v polynomiálním čase.
 - Má daný *graf* kostru o velikosti maximálně k ?
 - Existuje v daném acyklickém grafu mezi uzly a a b cesta, jejíž délka je nejvýše k ?
- **Třída NP** – obsahuje problémy, které jsou rozhodnutelné pomocí nedeterministického Turingova stroje v polynomiálním čase – tzn. jsme schopni ověřit jejich řešení v polynomiálním čase.

- Lze daný graf obarvit maximálně k barvami?
- Existuje v daném grafu hamiltonovská kružnice?
- Existuje v daném grafu klika o alespoň k vrcholech?

Literatura

- WRÓBLEWSKI, Piotr. Algoritmy : Datové struktury a programovací techniky. Brno : Computer press, 2004. 351 s. ISBN 80-251-0343-9.
- KVASIL, Bohumil, et al. Algoritmus. In Malá československá encyklopedie. Praha : Academia, 1984. s. 108.
- VELEBIL, Jiří. Diskrétní matematika : Text k přednášce. Praha : [s.n.], 2007. 197 s.



– Pavel Mička
Zakladatel a administrátor stránek Algoritmy.net