

Otázka: Algoritmus, algoritmická složitost

Předmět: Programování, Informatika

Přidal(a): Ondřej Rethy

Algoritmus

- Z obecného hlediska se jedná o „přesný návod nebo postup pro vyřešení konkrétní úlohy“
- Název je od perského matematika Al-Chorezmí (9. století)
- Teoretický princip řešení programu (nejen programu, ale i například skládání rubikovy kostky)
- Například kuchařka je též algoritmus pro kuchaře na přípravu jídla
- Základní vlastnosti algoritmu
 - Elementárnost
 - Rozložit problém na co nejmenší prvky a ty poté řešit
 - Například úklid pokoje
 - Nejdříve sebrat oblečení, poté utřít prach, poté vyluxuovat, přemístit věci atd.
 - Konečnost
 - Algoritmus musí končit
 - Například úklid pokoje
 - Algoritmus musí někdy dojít k závěru, že je pokoj uklizen
 - Obecnost
 - Algoritmus by měl být obecný, tzn. Aby šel použit na více podobných řešení.
 - Například úklid pokoje

- Musí jít tedy na všechny typy a rozměry pokojů
- Determinovanost
 - Stejně úkony udělají za stejný čas
 - Například úklid pokoje
 - Necháme 2 roboty uklízet identický pokoj s identickým nepořádkem => Výsledek bude naprosto identický a za stejný čas
- Determinismus
 - Každá část znamená přesně danou věc
 - V programování se s tím nemusíme tolik často setkávat
 - Například úklid pokoje
 - Sebrat oblečení musí přesně znamenat sebrat oblečení tím a tím způsobem.
- Výstup
 - Každý algoritmus by měl být výsledek
 - Například úklid pokoje
 - Chceme aby tento algoritmus měl jako výsledek uklizený pokoj
- Metody návrhu
 - Shora dolů ($X \rightarrow x$)
 - postup řešení rozkládáme na jednodušší operace, až dospějeme k elementárním krokům.
 - Zdola nahoru ($x \rightarrow X$)
 - z elementárních kroků vytváříme prostředky, které nakonec umožní zvládnout požadovaný problém.
 - Kombinace obou
 - obvyklý postup shora dolů doplníme „částečným krokem“ zdola nahoru tím, že se například použijí knihovny funkcí nebo vyšší programovací jazyk

Algoritmická složitost

- Udává rychlost algoritmu vzhledem k množině vstupních dat
- Dvě hlavní zaměření
 - Paměť
 - Čas
- Faktorialová, exponenciální, kubická, kvadratická, lineární, logaritmická

Rekurze

- Je volání funkce z funkce, která ještě nedokončila svůj průběh
- Z důvodu konečnosti musí být volání podmíněné (memory overflow)
- Funkci můžeme volat:
 - Přímo
 - Nepřímo
- Podprogram může být volán jednou nebo vícekrát
 - Lineární rekurze
 - Stromová struktura (zakořeněný strom, binární)

Náhodnost

- Náhodné algoritmy se snaží nalézt řešení problému náhodným rozhodováním o svém postupu
- Při vytváření musíme dokončit všechny, směry, kterými se může program vydat
- Program se rozhoduje několika způsoby:
 - V každém uzlu rozhodne postup náhodně
 - Na začátku vybere jeden z deterministických algoritmů
- Generování pseudonáhodných čísel - Kongruenční generátor