

Algoritmizace

- Algoritmus je přesný postup řešení problému
- Popisuje postup, ne konkrétní implementaci

Vlastnosti algoritmů

- Elementárnost – algoritmus se skládá z konečného počtu jednoduchých kroků
- Konečnost – algoritmus musí skončit v konečném počtu kroků pro každý vstup
 - V některých literaturách se jako algoritmus označují i tzv. reaktivní procesy, které průběžně reagují na vstupy
- Obecnost – algoritmus neřeší konkrétní problém (3 x 7), ale obecnou třídu obdobných problémů (jak spočítat součin dvou celých čísel)
- Determinovanost – pokud za stejných podmínek poskytuje stejné výstupy
 - Někdy je ale i potřeba náhodnost (simulace hodu kostkou, generování klíčů)
- Determinismus – každý krok programu je jednoznačně definován
- Výstup – algoritmus obsahuje alespoň jeden výstup kterým se tvoří odpověď

Metody návrhu

- Shora dolů – postup řešení rozkládáme na jednodušší operace, až dospějeme k elementárním krokům.
- Zdola nahoru – 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

Způsoby zápisu algoritmu

Přirozený jazyk

- Algoritmus je popsán slovy
- Jednoduché, ale nejednoznačné
- Nepochopitelné pro cizince

Vyšší programovací jazyk

- Snadné vytvoření programu, ale nutná znalost konkrétního jazyka, nepřehledné

Vývojový diagram

- Graficky znázorňuje jednotlivé kroky algoritmu
- Je složen z různých obrazců propojených šipkami
- Jednoduché a pochopitelné
- Symboly:
 - Šipka – určuje směr zpracování
 - Preferovaný směr dolů a doprava
 - Obdélník s popisem – definuje dílčí krok zpracování
 - Kosočtverec – větvení postupu na základě splnění podmínky
 - Zaoblený obdélník – počátek či konec zpracování
 - Kruh – spojka několika šipek
 - Rovnoběžník – získání vstupů/zobrazení výstupů

Druhy algoritmů

- Rekurzivní algoritmy
 - Algoritmy, které volají samy sebe
 - Fibonacci
- Pravděpodobnostní algoritmy
 - Provádějí některá rozhodnutí pseudonáhodně
- Genetické algoritmy
 - Pracují na základě napodobování biologických evolučních procesů
 - Mutace a křížení
- Heuristický algoritmus
 - Neklade si nalézt přesné řešení, ale pouze vhodné přiblížení
 - Používá se, kde zdroje nepostačují k využití exaktních algoritmů

Složitost algoritmu

- Potřebujeme metriky algoritmů, kterými můžeme porovnávat efektivitu algoritmů
- **Časová složitost** (neboli asymptotická) algoritmu uvádí časovou náročnost algoritmu v závislosti na počtu vstupních dat
 - Nezáleží jen na počtu, ale i na velikosti
- **Paměťová složitost** algoritmu uvádí velikost paměti využívané při výpočtu