

02 - Algoritmus, algoritmická složitost

Algoritmus

- teoretický princip řešení problému
- označuje postupy, které splňují:
 - **konečnost** (konečný počet prvků)
 - **obecnost** (neřeší konkrétní problém, ale problémy obecně)
 - **determinovanost** (ze stejného vstupu generuje vždy stejný výstup)
 - **determinismus** (každý krok je jednoznačně definován)
 - **výstup** (obsahuje výstup, které tvoří odpověď na daný problém)

Algoritmická složitost

- určuje, zda je algoritmus schopen skončit v použitelném čase, který umožňuje algoritmus prakticky využít
- díky tomu se algoritmus srovnává s jinými
 - i. podle doby trvání algoritmu (časová složitost)
 - ii. podle velikosti používané paměti (paměťová složitost)
- pro označení doby se používá symbol $O()$
- **typické algoritmické složitosti:**
 - $\log n$, n^2 , n^3 , n , $n!$, n^n

Rekurze

- stav, kdy je objekt součástí sebe samého
- funkce je znovu volána dříve, než se dokončilo její předchozí volání
- funkce volá sama sebe
- je paměťově náročná
- rekurze musí být ukončena, jinak dojde k tzv. **zacyklení** - program nepokračuje dál a zasekne se na rekurzi
- **Přímá rekurze**
 - funkce volá přímo sama sebe
- **Nepřímá rekurze**
 - funkce A volá funkci B, ta poté volá funkci A - vzniká kruh
- **Lineární rekurze**
 - funkce volá sama sebe jednou
- **Stromová rekurze**
 - funkce volá sama sebe vícekrát

Náhodnost

- **Generátor náhodných čísel** - vybírá náhodné číslo podle zadaných parametrů / rozsahu
 - Pseudonáhodný generátor - zdánlivě náhodná čísla
 - Pravý náhodný generátor - podle fyzikálních jevů
- náhodné algoritmy hledají řešení pomocí náhodného rozhodování o svém postupu
- je třeba předem myslet na všechny možnosti