

12. Co víte o složitosti algoritmu?

- Každému algoritmu lze jednoznačně přiřadit rostoucí funkci zvanou asymptotická složitost, která charakterizuje počet operací algoritmu v závislosti na rostoucím rozsahu vstupních dat
- Asymptotická složitost algoritmu A je řád růstu funkce $f(N)$, která charakterizuje počet elementárních operací algoritmu A při zpracování dat o rozsahu N
- Řád růstu funkce f je taková „co nejjednodušší“ funkce g, pro kterou platí že f je asymptoticky ohraničena funkcí g z obou stran (až na konstantu)
- „Technický“ přístup: řád funkce určíme tak, že zanedbáme:
 - Aditivní členy rostoucí pomaleji nebo stejně rychle
 - Multiplikativní konstantu

Příklad:

$$f(n) = 7 \cdot 2^n + 2 \cdot 2^{n-1} + 4 \cdot 3^{n/2} \quad \text{řád } f(n) \text{ je } 2^n$$

$$ff(n) = n + \log_2(n) \quad \text{řád } ff(n) \text{ je } n$$

- Obvyklý zápis složitosti algoritmu je pomocí „velké O notace“ jako $O(f(N))$ (např. $O(N)$).

Amortizovaná složitost

- Určuje časovou složitost jako průměr v sekvenci nejhorších případů
- Na rozdíl od průměrné nevyužívá pravděpodobnost a je zaručena
- Paradoxně může mít lepší průběh než asymptotická (ASS)
 - Algoritmy s vysokou ASS často mění strukturu samotných dat
 - To se samozřejmě promítá do časové složitosti
 - A potom se „špatný případ“ nestane dlouhou dobu – nastane jeho amortizace