

Cvičení z Algoritmizace a Programování 1

4. cvičení

Algorithmizace

DÚ Asymptotická složitost - řešení

- Nevynechávat kvantifikátory
- Odpověď začít uvedením, zda tvrzení platí nebo ne
- ... pak teprve samotný důkaz

Hornerovo schéma

- Obecně: střídání dvou operací
- Aplikace:
 - Převod mezi číselnými soustavami
 - Umocňování
 - Vyhodnocování polynomu
 - ...

$$\begin{aligned}110010 &\approx 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = \\ &= (((((1 \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 0) = 50\end{aligned}$$

- převod číselné hodnoty **do dvojkové soustavy**
- algoritmus: Hornerovo schéma využité v opačném směru
posloupnost zbytků při celočíselném dělení dvěma
tvoří **odzadu** dvojkový zápis čísla
→ připojování dvojkových cifer do stringu **zleva**

$$50 : 2 = 25, \text{ zb. } 0$$

$$25 : 2 = 12, \text{ zb. } 1$$

$$12 : 2 = 6, \text{ zb. } 0$$

$$6 : 2 = 3, \text{ zb. } 0$$

$$3 : 2 = 1, \text{ zb. } 1$$

$$1 : 2 = 0, \text{ zb. } 1$$

DÚ Soustavy

- Viz ReCodex
- Tentokrát je to „programovací úkol“
- -> funguje to stejně jako DÚ z Programování
 - Odevzdáváte program v Pythonu
 - ReCodex Vám ho hned vyhodnotí, můžete opravovat dokud nedostanete 100 %

Programování

Domácí úkol - řešení

- Rozklad na prvočinitele
- Postupně dělit dokud to jde, jinak zvýšit dělitele
- Proč to funguje – proč vrací jen prvočísla?

Úlohy na práci s **list** a funkce

- Viz ReCodEx – Nepovinné Programování
- Debugging
 - Breakpoints
 - Debug režim (F5)

Domácí úkol

- Průsečík úseček