

# Základy programovania

Marek Tesař

Predpokladám, že všetci z Vás, milí čitatelia, ste sa už stretli so slovami Matematická olympiáda (ďalej len MO) a asi viete, že za týmto spojením sa ešte zvyknú písať akési ďalšie písmenká. Konkrétnejšie písmená  $A, B$  a  $C$  znamenajú, že ide o číslu matematiku, kým písmenko  $P$  má úplne iný význam. Ide o kategóriu *programovanie*, ktorá má zopár odlišností. V podstate vždy tam od vás chcú aby ste napísali čo najefektívnejší algoritmus. No ale čo to ten algoritmus vlastne je a kedy je algoritmus efektívny?

**Definícia.** *Algoritmus je sekvenčný zoznam inštrukcií (elementárnych operácií), ktoré prevedú vstupné dáta na výstupné.*

Kde *elementárnymi operáciami* budeme rozumieť napríklad aritmetické operácie, načítanie čísla zo vstupu, porovnanie dvoch čísiel a ostatné operácie, ktoré vieme vykonať v konštantnom čase. No a algoritmus je potom iba postup podľa ktorého počítač funguje (vykonáva elementárne operácie). Počítač totiž robí iba to, čo má vo svojom programe a pri tom tomu vôbec nerozumie. Je to niečo podobné ako keby ste chceli naučiť malé štvoročné dieťa binárne sčítovanie. Na to mu len vysvetlíte, že sa má na dve postupnosti číslic pozerať z prava do ľava a vždy si má pozrieť dve číslice a akýsi prechod a podľa toho zapísať nejaký nový prechod a výslednú číslicu (toto mu dáte napríklad v nejakej tabuľke). Ak sa bude toto dieťa správať podľa tohoto postupu, tak bude v podstate sčítovať dve čísla a pri tom nebude mať ani tuha, že čo to vlastne robí. No a počítač je na tom dosť podobne. A keď už vymyslíme algoritmus, tak to ešte zďaleka neznamená, že sme hotový, pretože ešte potrebujeme určiť jeho časovú a priestorovú zložitosť. Skôr ako si povieme, že o čo ide, tak si napíšme ešte jednu definíciu.

**Definícia.** *O funkcii  $f : \mathbb{N} \rightarrow \mathbb{N}$  povieme, že  $f(n) = O(n^k)$  pre nejaké  $k \in \mathbb{N}$  ak existuje  $c > 0$  také, že  $f(n) \leq c \cdot n^k$  pre všetky  $n \in \mathbb{N}$ .*

**Príklad.** Tak napríklad nasledujúce funkcie sú  $O(n)$ :  $n, n + 2, 4n, 234n + 432, \dots$ . Medzi funkcie, ktoré sú  $O(n^3)$  patrí napríklad:  $n^3, 3n^3 + 2, 0,01n^3 + 10n^2 - 25n, \dots$

**Poznámka.** Často sa však používajú aj označenia  $f(n) = O(n \cdot \log(n))$  a to vtedy, keď existuje  $c > 0$ , že  $f(n) \leq c \cdot n \cdot \log(n)$  pre  $\forall n \in \mathbb{N}$ .

**Definícia.** *Označme si teraz  $T(n)$  počet elementárnych operácií, ktoré vykoná náš algoritmus pri spracúvaní vstupu veľkosti  $n$ . O algoritme povieme, že jeho časová zložitosť je  $O(n^k)$  ak  $T(n) = O(n^k)$ .*

Analogicky ak si označíme  $N(n)$  veľkosť pamäte, ktorú použil náš algoritmus, tak o algoritme povieme, že jeho pamäťová zložitosť je  $O(n^k)$  ak  $N(n) = O(n^k)$ . Analogicky by sme mohli hovoriť o tom, že pamäťová, resp. časová zložitosť sú  $O(n \cdot \log(n))$  a iné.

**Poznámka.** O algoritme potom povieme, že je efektívny ak neexistuje iný algoritmus riešiaci ten istý problém, ktorý by však pracoval rádovo rýchlejšie.

Tak z teoretického hľadiska je toto asi všetko čo budeme k prednáške potrebovať. Zvyšok času už budeme iba vymýšľať algoritmy k daným problémom, rozoberať ich zložitosť (či už časovú alebo priestorovú) a následne ich vylepšovať. Takže po tejto prednáške by ste mali mať v sebe základ na to, aby ste mohli budúci rok riešiť MO-P. Medzi úlohy ktorými sa budeme zaoberať určite patrí:

**Úloha.** Na vstupe máme neutriedenú postupnosť rôznych čísel  $a_1, a_2, \dots, a_n$ , kde  $n \in \mathbb{N}$ . Navrhňte čo najefektívnejší algoritmus, ktorí by túto postupnosť utriedil.

**Úloha.** Na vstupe máme prirodzené číslo (maticu)  $A$  a prirodzené čísla  $k$  a  $m$ . Vypočítajte čo najefektívnejšie hodnotu  $A^k \bmod m$ .

**Úloha.** Nech  $f_i$  je  $i$ -te Fibonnachiho číslo, t.j.  $f_0 = f_1 = 1$  a  $f_{n+2} = f_{n+1} + f_n$ , pre  $n \geq 0$ . Vypočítajte efektívne hodnotu  $f_k$  pre dané  $k \in \mathbb{N}$ .

**Úloha.** Majme tabuľku o rozmeroch  $m \times n$ , kde  $m, n \in \mathbb{N}$ . Na každom políčku je napísaná hodnota 0 alebo 1. Navrhňte algoritmus, ktorý nájde obdĺžnik s najväčším možným obsahom, ktorého strany budú rovnobežné so stranami našej tabuľky a ktorý bude obsahovať iba samé jednotky.

Tak toto bola malá ukážka príkladíkov, ktorým sa budeme venovať na prednáške. Určite sa tam však vyskytnú aj nejaké iné, takže ak ste tieto už poznali, tak sa nemusíte báť, že by ste sa tam nudili...