

## Cvičení 1

**Na rozeřtí:** Pro zadané funkce  $f(n), g(n)$  zjistěte, zda  $f = \mathcal{O}(g)$  nebo  $f = \Omega(g)$  (a kdy nastane obojí, tj.  $f = \Theta(g)$ ).

- (a)  $f(n) = 100n + \log n$ ,  $g(n) = n + (\log n)^2$       (d)  $f(n) = (\log n)^{\log n}$ ,  $g(n) = n / \log n$   
(b)  $f(n) = n \log n$ ,  $g(n) = 10n \log 10n$       (e)  $f(n) = n!$ ,  $g(n) = 2^n$   
(c)  $f(n) = n^{1.01}$ ,  $g(n) = n \log^2 n$       (f)  $f(n) = \sum_{j=1}^n j^k$ ,  $g(n) = n^{k+1}$

### Úlohy na RAMu

**Úloha 1:** Jak byste na RAMu naprogramovali podmínky a cykly?

**Úloha 2:** A jak napsat volání funkce?

**Úloha 3:** Jak prohodit obsah dvou paměťových buněk bez použití jakékoli jiné buňky?

**Úloha 4:** Naprogramujte binární vyhledávání na RAMu. Rozmyslete, kolik instrukcí se v nejhorším případě provede při hledání v posloupnosti délky  $n$ .

**Úloha 5:** Co když v programu na RAMu budeme potřebovat několik různých polí, jejichž velikost předem neznáme?

**Úloha 6:** Uvažme RAM s neomezenou velikostí čísel.

- (a) Jak zakódovat libovolné množství celých čísel  $c_1, \dots, c_n$  do jednoho  $C$  (aby se jednotlivá čísla dala jednoznačně dekodovat)?  
(b) Jak pozměnit libovolný program, aby použil jen konstantně mnoho paměťových buněk? Na časové složitosti nám teď nezáleží.

### K rozmyšlení

**Úloha 7:** Napadá vás, jak model RAM rozšířit, aby program mohl interagovat s uživatelem?

**Úloha 8:** Je prostorová složitost omezená tou časovou?

## Rekurzivní hádanky

Poznáte, co dělají následující funkce?

```
f(x,y):
    if x==0 => return y
    else => return f((x&y) << 1, x^y)

g(x,y):
    if y==0 => return 0
    else if even(y) => return 2*g(x, y/2)
    else => return 2*g(x, y/2) + x

h(x,y):
    if x<y => return (0,x)
    else:
        (a,b) <- 2*h(x/2, y)
        if odd(x) => b <- b+1
        if b>=y => a <- a+1, b <- b-y
        return (a,b)

d(x,y):
    if x==y => return x
    if even(x) and even(y): return 2*d(x/2, y/2)
    if even(x): return d(x/2, y)
    if even(y): return d(x, y/2)
    if x>y: return d(x-y, y)
    else: return d(x, y-x)
```