

PŘÍKLAD 1. *Bublínkové třídění na RAMu.* Napište implementaci Bubble-Sortu v modelu RAM. Na vstupu je v buňce [0] délka posloupnosti celých čísel a samotná posloupnost se nachází v buňkách [1] - [[0]].

PŘÍKLAD 2. *Překlad pro RAM.* V minulém cvičení jsme potřebovali napsat cykly `for` a `while` v modelu RAM. Rozmyslete si, jak přepisovat do RAMu další konstrukce z vyšších jazyků: složitější podmínky, volání podprogramů a funkcí, popř. i s rekurzí.

PŘÍKLAD 3. *Více polí.* Jak na RAMu napsat program, který potřebuje při svém běhu více (různě velkých) pomocných polí?

PŘÍKLAD 4. *Zneužíváme RAM.* Mějme model RAM s neomezenou velikostí čísel. Vymyslete, jak zakódovat libovolně mnoho celých čísel  $c_1, \dots, c_n$  do jednoho čísla  $C$  tak, aby původní čísla šla jednoznačně dekodovat.

PŘÍKLAD 5. *Konstantní paměť.* Navrhněte postup, jak v případě neomezené kapacity paměťové buňky pozměnit libovolný program na RAMu tak, aby používal jen konstantně mnoho buňek. Program můžete libovolně zpomalit.

PŘÍKLAD 6. *Rychlé násobení matic.* Předpokládejme jednotkovou cenu instrukce s neomezenými čísly. Jak v čase  $O(n)$  zakódovat dva vektory  $n$  celých čísel, abychom mohli v konstantním čase spočítat skalární součin dvou vektorů? Jak z toho odvodit algoritmus pro násobení matic v čase  $O(n^2)$ .

PŘÍKLAD 7. *Konstantní mocnina.* Jak na RAMu v konstantním čase otestovat, jestli je číslo mocninou dvojky?

PŘÍKLAD 8. *Swap.* Jak na RAMu prohodit obsah dvou buněk, aniž bychom použili jakoukoliv jinou buňku?

PŘÍKLAD 9. *Konstantní paměť 2.* Kolik nejméně buněk je potřeba v př. 4?

PŘÍKLAD 10. *Rekurzivní hádanky.* Co dělají následující funkce?

$f(x, y)$ :

if  $x==0$  => return  $y$

$g(x, y)$ :  
else => return  $f((x \& y) \ll 1, x^y)$

if  $y==0$  => return  $0$

else if  $\text{even}(y)$  => return  $2 * g(x, y/2)$

else => return  $2 * g(x, y/2) + x$