

**Příklad 1.**

Ukažte, jak pomocí RAMu zapíšete podmínky, cykly.

**Příklad 2.**

Naprogramujte na RAMu třídění počítáním. V buňce 0 najdete délku seznamu  $n$ , buňky  $1, \dots, n$  pak obsahují jednotlivé prvky seznamu. Ostatní buňky obsahují nedefinovanou hodnotu. Všechna čísla jsou nezáporná, ale nejsou nijak omezená.

Jakou časovou složitost (ať už počet instrukcí přímý i následnou asymptotickou) má tento algoritmus? Co prostorová složitost?

Jak se situace změní, pokud povolíme i záporná čísla v poli?

**Příklad 3.**

Jak byste na RAMu naprogramovali volání funkce?

**Příklad 4.**

Musí platit, že prostorová složitost RAMu je shora omezená časovou?

**Příklad 5.**

Jak se změní odpověď na předchozí otázku, pokud změním definici prostorové složitosti jako počet různých buněk:

- do kterých jsme někdy něco zapsali
- ze kterých jsme někdy něco přečetli
- ke kterým jsme někdy přistoupili

**Příklad 6.**

Mějme program v RAMu, kde dopředu na základě vstupu velikosti  $n$  umíme v čase  $\mathcal{O}(f(n))$  a v konstantním prostoru určit, že všechna používaná čísla budou v rozsahu  $0, \dots, m$ . Na základě toho ukažte, jak tento program převedete na jiný program, který kromě vstupu využije  $\mathcal{O}(1)$  buněk.

Zkuste nejprve vymyslet, jak zakódovat buňky, a potom pro jednotlivé instrukce ukázat, jak pracovat s tímto kódováním.

**Příklad 7.**

Uvažme model časové složitosti, kde máme neomezená čísla a jednotkovou cenu všech instrukcí.

Jak se asymptoticky změní časová složitost programu, jestliže původně byla  $\mathcal{O}(g(n))$ , pokud použijeme předchozí převod?

**Příklad 8.**

Rozmyslete si, jak na RAMu napsat program, který potřebuje více různých polí, jejichž velikost dopředu neznáme.