2. CVIČENÍ Z ADS 1

Výpočetní model RAM

1 Nerekurzivní Levenštejn

Na přednášce byl předveden algoritmus počítající Levenštejnovu vzdálenost za pomocí rekurze a cacheování. Vymyslete algoritmus počítající Levenštejnovu vzdálenost bez použití rekurze.

2 Třídění výběrem na RAMu

Napište implementaci SelectSortu v modelu RAM. Na vstupu je v buňce `[0]` délka posloupnosti celých čísel a samotná posloupnost se nachází v buňkách `[1]`–`[[0]]`.

3 Překlad pro RAM

Na přednášce jsme si ukázali, jak cykly `for` a `while` v modelu RAM. Rozmyslete si, jak přepisovat do RAMu další konstrukce z vyšších jazyků: složitější podmínky (`and`, `or`, `not`), volání[1] jednoduchých podprogramů a funkcí, popř. i s rekurzí.

4 Více polí.

Jak na RAMu napsat program, který potřebuje při svém běhu více (různě velkých) pomocných polí? Tj. chceme program, který ve svém průběhu bude chtít vytvořit N polí, přičemž velikost každého pole zjistí teprve ve chvíli, kdy ho bude chtít vytvořit.

5 Zneužíváme RAM

Mějme model RAM s neomezenou velikostí čísel. Vymyslete, jak pro zadané n (pro začátek přemýšlejte o $n = 2$) zakódovat n libovolně velkých celých čísel $c_1, \ldots, c_n$ do jednoho čísla C tak, aby původní čísla šla jednoznačně dekódovat bez žádné další informace.

Co když k dekódování nebude k dispozici ani n? Tedy dostaneme nějak dlouhý seznam čísel, ten nějak zakódujeme do C, a pak chceme pouze z C dekódovat ten samý seznam čísel.

6 Konstantní paměť

Navrhněte postup, jak v případě neomezené kapacity paměťové buňky pozměnit libovolný program na RAMu tak, aby používal jen konstantně mnoho buněk. Program můžete libovolně zpomalit.

7 Rychlé násobení matic

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

[1] Nezapomeňte, že návěstí je konstanta. Tedy nelze použít například `goto [0]`.

[2] Můžete si nejprve rozmyslet variantu, že jsou čísla omezená konstantou 2^{64}, poté variantu s libovolně velkými čísly.

8 Bonusy:

8.1 Swap

Jak na RAMu prohodit obsah dvou buněk, aniž bychom použili jakoukoliv jinou buňku?

8.2 Konstantní mocnina

Jak na RAMu v konstantním čase otestovat, jestli je číslo mocninou dvojky?

8.3 Rekurzivní hádanky

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
```

8.4 Konstantní paměť 2

Kolik nejméně buněk je potřeba v př. 6?