

Z minula Je rychlejší mocnit matici a následně ji pronásobit maticí nebo postupně pronásobovat matici vektorem. (Matice je $n \times n$ a k -tá mocnina.)

Definice RAM Zdefinujeme si RAM. Co když počítáme prostorovou složitost jako počet použitých buněk.

Velikost buňky Na jaký problém narazíme, pokud velikost buňky je

- (a) libovolně velká
- (b) konstatně velká
- (c) polynomiální vzhledem k velikosti vstupu
- (d) libovolně velká, ale velikost v bitech se projevuje při počítání složitostí

Kódování Jak byste kódovali vektor pokud může obsahovat libovolně velké hodnoty? Umíte v daném kódování takové vektory rychle sčítat?

Vyměň Vymyslete jak na RAMu vyměnit dvě buňky aniž byste použili pomocnou proměnnou.

Program Zkuste si na RAMu naprogramovat bubblesort nebo jiný algoritmus.

Na zamyšlení Co dělají následující pseudokódy funkcí f,g,h?

```
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)
```

Program Rozmyslete si jak na RAMu naprogramovat tyto konstrukce známé z klasických programovacích jazyků:

- (a) podmínky (if else, switch),
- (b) cykly (while, do while),
- (c) volání podprogramů s lokálními proměnnými a rekurzí.

(15 bodů)

Co počítám? Vstupem následujícího algoritmu je přirozené číslo N zapsané v buňce [0]. Určete, co je jeho výstupem a spočítejte přesně počet provedených instrukcí vůči N . **(8 bodů)**

```
I := 1
Z := 1
VNEJSI: if I > [0] then halt
J := 1
I := I + 1
VNITRNI: if J > I then goto VNEJSI
[Z] := 1
Z := Z + 1
J := J + 1
goto VNITRNI
```