1. CVIČENÍ Z ADS 1, ČTVRTEK 15:40, LS '24

úvodní příklady a programování na RAMu

1. *Součet dvojice.* Na vstupu je setříděné pole délky N a číslo K. Vymyslete algoritmus, který v poli najde co nejefektivněji dvojici čísel, jež mají součet přesně K, případně vrátí, že tam taková dvojice není. Pokud je takových dvojic více, stačí vrátit jednu libovolnou.

2. *Nejbohatší úsek* Máme zadanou posloupnost celých čísel $x_1, \ldots, x_n$ a chceme v ní nalézt úsek (tj. souvislou podposloupnost), jehož součet je největší možný.

3. *Překlad pro RAM.* Na přednášce jsme napsali cykly `for` a `while` v modelu RAM. Rozmyslete si, jak přepisovat do RAMu další konstrukce z vyšších jazyků: složitější podmínky (AND, OR), volání 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?

5. *Zneužíváme RAM.* Mějme model RAM s neomezenou velikostí čísel. Vymyslete, jak zakódovat libovolně mnoho celých čísel $c_1, \ldots, c_n$ do jednoho čísla C tak, aby původní čísla šla jednoznačně dekódovat. (Dekódování nemusí být příliš efektivní.)

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, abychom mohli v konstantním čase spočítat skalární součin dvou vektorů? Jak z toho odvodit algoritmus pro násobení matic v čase $O(n^2)$.

Bonusové úlohy:

8. *Součet úseku* Máme zadanou posloupnost **kladných** celých čísel $x_1, \ldots, x_n$ a chceme v ní nalézt úsek se součtem **přesně** K pro zadané K. [Bonus^2: co když máme na vstupu i záporná čísla?]

9. *Konstantní mocnina.* Jak na RAMu (nebo v C) v konstantním čase otestovat, jestli je číslo mocninou dvojky?

10. *Swap.* Jak na RAMu (nebo v C) prohodit obsah dvou buněk, aniž bychom použili jakoukoliv jinou buňku?

11. *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
```