

**RAM:** „Random Access Machine“, pomyslný stroj s pamětí celočíselných buněk vykonávající program dle posloupnosti instrukcí.

*Program* ::= (návěstí: instrukce | instrukce)\*

*Instrukce* ::= **halt** | **goto** návěstí | **if** výraz **then** instrukce | *místo* := výraz

*Místo* ::= registr **A-Z**, [celočíselná adresa], [*místo*]

*Hodnota* ::= *místo* | celočíselný literál

*Výraz* ::= *hodnota* | *hodnota* operátor *hodnota*

*Operátor* ::= + | - | \* | / | % | & | | ^ | << | >> | = | <> | < | > | <= | >=

1. Napište bublinkové třídění jako program pro RAM, popište formát vstupu a výstupu a vyjádřete přesně (počtem instrukcí) časovou a paměťovou složitost programu v nejlepším i nejhorším případě.
2. Vyberte si nějaký strojový/assembly jazyk reálného počítače a porovnejte ho s jazykem instrukcí RAM.
3. Uvažte RAM s jednotkovou cenou instrukcí a bez omezení na velikost buněk.
  - a) Najděte způsob, jak libovolných  $n$  čísel omezené velikosti zakódovat do jedné buňky.
  - b) Jaké aritmetické operace jdou na takto zakódovaných „vektorech“ provádět v konstantním čase?
  - c) Vymyslete, jak transformovat libovolný program na RAMu s omezenou velikostí buňky do této neomezené varianty tak, aby měl konstantní paměťovou složitost.
  - d) Najděte reprezentaci dvou množin vektorů  $A, B$  takovou, že skalární součin  $a \cdot b$  (pro libovolné  $a \in A, b \in B$ ) lze vypočítat v čase  $\mathcal{O}(1)$ . Použijte tuto reprezentaci k násobení matic v čase  $\mathcal{O}(n^2)$ .
4. Minule jste dokázali, že existují dvojice funkcí neporovnatelné inkluzí tříd  $\mathcal{O}$ . Nyní dokažte, že to platí i pokud se omezíme jen na neklesající funkce.
5. Navrhněte grafovou reprezentaci jízdního řádu a popište algoritmus, který najde pro výchozí zastávku, čas odjezdu a maximální čas příjezdu množinu zastávek, do kterých se lze dopravit (tzv. izochronu).