Cvičení 8

Příklad 1: Uvažujme následující problém:

VSTUP: Dvě n-bitová přirozená čísla x a y.

VÝSTUP: Hodnota součtu $x + y$ reprezentovaná binárně.

a) Navrhněte paralelní algoritmus pro stroj PRAM typu EREW řešící tento problém s časovou složitostí $\mathcal{O}(\log n)$ při použití $\mathcal{O}(n)$ procesorů.

b) Navrhněte paralelní algoritmus pro stroj PRAM typu CRCW COMMON řešící tento problém v čase $\mathcal{O}(1)$ s polynomiálním počtem procesorů.

   Kolik procesorů bude vámi navržený algoritmus potřebovat?

Příklad 2: Navrhněte paralelní algoritmus pro stroj PRAM typu CREW řešící následující problém s časovou složitostí $\mathcal{O}(\log n)$ při použití $\mathcal{O}(n^2)$ procesorů:

VSTUP: Dvě n-bitová přirozená čísla x a y.

VÝSTUP: Hodnota součinu $x \cdot y$ reprezentovaná binárně.

Příklad 3: Předpokládejme, že máme dán nějaký fixní nedeterministický konečný automat $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$. Uvažujme následující problém:

VSTUP: Slovo $w \in \Sigma^*$.

OTÁZKA: Je slovo w přijímáno automatem $\mathcal{A}$ (tj. platí $w \in \mathcal{L}(\mathcal{A})$) ?

Popište, jak pro daný automat $\mathcal{A}$ vytvořit paralelní algoritmus pro stroj PRAM typu EREW řešící tento problém s časovou složitostí $\mathcal{O}(\log n)$ při použití $\mathcal{O}(n)$ procesorů (kde n je délka slova w).

Příklad 4: Uvažujme jazyk L nad abecedou $\Sigma = \{\mathtt{[}, \mathtt{]}\}$ generovaný následující bezkontextovou gramatikou:

$$A \longrightarrow \varepsilon \mid AA \mid \mathtt{[}\,A\,\mathtt{]}$$

Navrhněte efektivní paralelní algoritmus pro stroj PRAM řešící následující problém:

VSTUP: Slovo $w \in \Sigma^*$.

VÝSTUP: Zjistit, zda slovo w patří do jazyka L a pokud ano, tak ke každé levé závorce najít pozici jí odpovídající pravé závorky, a podobně ke každé pravé závorce najít pozici jí odpovídající levé závorky.

Jaká je časová složitost vámi navrženého algoritmu a kolik procesorů potřebuje?

Příklad 5: Navrhněte paralelní algoritmus pro stroj PRAM typu CREW řešící následující problém s časovou složitostí $\mathcal{O}(\log n)$ při použití $\mathcal{O}(n)$ procesorů:

VSTUP: Trojice polí A, L a R. Všechna tři pole jsou velikosti n a indexována od 0. Jejich prvky jsou přirozená čísla.
Pro hodnoty prvků polí L a R navíc platí $0 \leq L[i] \leq R[i] < n$ pro každé i, kde $0 \leq i < n$.

VÝSTUP: Pro každé i, kde $0 \leq i < n$, hodnota

$$\min\{\, A[j] \mid L[i] \leq j \leq R[i] \,\}$$

Příklad 6: Neorientovaný graf $G = (V, E)$, který je souvislý a acyklický, se nazývá strom. Označme takový druh grafu jako ***nezakořeněný strom***, protože žádný z vrcholů v něm není označen jako kořen.

Pokud zvolíme v nezakořeněném stromě G některý vrchol $r \in V$ jako kořen, můžeme hrany grafu G orientovat tak, aby pro každou hranu platilo, že bude směřovat od vrcholu, který má větší vzdálenost k vrcholu r, do vrcholu, který má menší vzdálenost k vrcholu r. Můžeme se na to dívat tak, že hrany vedou od potomků k rodičům. Označme takový strom, kde jsou hrany zorientovány tímto způsobem, jako ***zakořeněný strom*** s kořenem r.

Navrhněte paralelní algoritmus pro stroj PRAM typu CREW řešící následující problém s časovou složitostí $\mathcal{O}(\log n)$ při použití $\mathcal{O}(n)$ procesorů:

VSTUP: Nezakořeněný strom $G = (V, E)$ a jeden z jeho vrcholů r.

VÝSTUP: Pro každou hranu $(u, v) \in E$ určit její směr v zakořeněném stromě, který vznikne z grafu G, pokud vrchol r bude zvolen jako jeho kořen.

Poznámka: Předpokládejte, že graf G je reprezentován jako pole vrcholů V, kde ke každému vrcholu máme dán seznam hran, které z něj vedou. Každá hrana je v této reprezentaci uložená dvakrát — jednou ve směru (u, v) a jednou ve směru (v, u). Předpokládejte, že prvky seznamů reprezentující tutéž hranu v jednom a druhém směru mají na sebe navzájem odkaz.