Prof. Andreas Goerdt
Andreas Berger

TU Chemnitz
Wintersemester 2002/03

Parallele Algorithmen

3. Übung

1. Aufgabe

Im List-Ranking-Algorithmus aus der Vorlesung wurde ausgenutzt, dass man auf einer EREW-PRAM zu einer Liste mit n Elementen eine 3-Färbung optimal (Arbeit $\mathcal{O}(n)$) in Zeit $\mathcal{O}(\log n)$ finden kann. Geben Sie einen Algorithmus an!

(Hinweis: Die Anwendung einer Runde von *SimpleColor* ergibt eine korrekte Färbung mit $\mathcal{O}(\log n)$ Farben. Die Listen-Elemente können nach ihrer Farbe sortiert werden. In jeder Farbgruppe befinden sich nun ausschließlich Elemente, die nicht benachbart sind. Schließlich kann man in $\mathcal{O}(\log n)$ Runden jeweils den Elementen einer Gruppe eine von drei Farben zuordnen.)

2. Aufgabe

Gegeben sei in einem Array S der Größe n eine Menge von Listen (Nachfolgerarray) mit insgesamt n Elementen. $S[i]$ enthält also den auf i folgenden Knoten bzw. 0, falls i das Ende der Liste markiert. Jedes Element gehört nur genau zu einer Liste.

Entwickeln Sie einen optimalen parallelen Algorithmus, der für jedes Element ermittelt, in welcher Liste es sich befindet. Dabei soll eine Liste nach ihrem ersten Element benannt sein. Geben Sie die Laufzeit ihres Verfahrens an!

3. Aufgabe

In der Vorlesung wurde ein Verfahren angegeben, welches für einen Baum eine sogenannte Euler-Tour in konstanter Zeit bestimmt. Voraussetzung war allerdings eine erweiterte Adjazenzlistendarstellung, wonach für jedes Element einer solchen Liste unmittelbar hervorging, auf welcher Adjazenzliste es sich befindet. Betrachten Sie nun Bäume ohne diese Zusatzinformation.

Wie schnell können Sie hier eine Euler-Tour bestimmen? (Hinweis: $\mathcal{O}(\log n)$ auf EREW-PRAM)

Kann dieser Zeitbedarf verringert werden, wenn man sich auf Bäume begrenzten Grades einschränkt? ($\mathcal{O}(1)$ für konstanten Grad)

4. Aufgabe

Bäume können in verschiedenen Reihenfolgen durchmustert werden. Bei der sogenannten preorder-Traversierung wird zuerst der Wurzelknoten betrachtet und anschließend werden die Teilbäume von links nach rechts jeweils in preorder-Reihenfolge behandelt. Geben Sie einen optimalen parallelen Algorithmus an, der zu einem gegeben Baum mit n Knoten jedem dieser Knoten dessen Preordernummer, also die Position in einer preorder-Reihenfolge der Knoten, zuordnet. Der Algorithmus sollte in Zeit $\mathcal{O}(n)$ auf einer EREW-PRAM ausführbar sein.

5. Aufgabe

Wir betrachten eine etwas andere Darstellung eines Baumes. Ein Baum mit n Knoten wird durch zwei Arrays der Größe n dargestellt: $fc(v)$ enthält die Nummer des ersten (linkesten) Kindes und $s(v)$ enthält die Nummer des nächsten (weiter rechts) stehenden Geschwisterknotens. Die Einträge enthalten 0, falls der enstsprechende Knoten nicht vorhanden ist.

(a) Geben Sie an, wie die Vaterknoten $p(v)$ zu jedem v optimal in Zeit $\mathcal{O}(\log n)$ berechnet werden können!

(b) Entwickeln Sie einen Algorithmus, der die Nummer der Blätter von links nach rechts in aufeinanderfolgende Speicherpositionen schreibt! Auch dies sollte mit linearer Arbeit in Zeit $\mathcal{O}(\log n)$ möglich sein.

6. Aufgabe

Ähnlich wie in Aufgabe 4 sollen diesmal die Inorder-Nummern für alle Knoten eines binären Baumes ermittelt werden. Geben sie einen entsprechenden Algorithmus mit Zeit- und Arbeitsschranken an!