

## Übungen zu „Parallele Programmierung“, WS 2008/09

Nr. 5, Abgabe der Aufgaben: 28. November 2008 in der Übung

### 5.1 Untere Schranken für paralleles Sortieren

2 Punkte

Begründen Sie die Korrektheit der im folgenden angegebenen unteren Schranken für das Sortieren von  $n$  Elementen auf verschiedenen Netzwerken mit jeweils  $n$  Knoten. Vor und nach dem Sortiervorgang sollen die zu sortierenden Elemente gleichmäßig verteilt sein, d.h. ein Element pro Prozessor.

- (a)  $\Omega(n)$  auf einem eindimensionalen Gitter
- (b)  $\Omega(\sqrt{n})$  auf einem zweidimensionalen Gitter

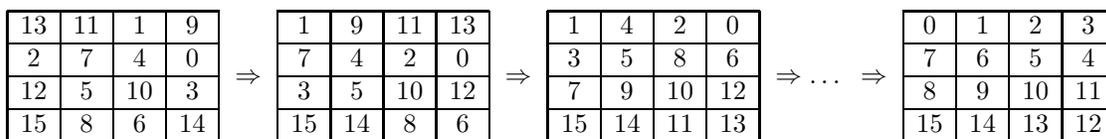
### 5.2 Sortieren auf einem zweidimensionalen Gitter

5 Punkte

Auf einem  $n \times n$  Gitter wird zur Sortierung von  $n^2$  Zahlen in Schlangenlinien (siehe Skizze) folgender Algorithmus vorgeschlagen:

Führe in  $\log n$  Phasen durch:

- Sortiere parallel alle Zeilen mit geradem Index  $\geq 0$  in aufsteigender Reihenfolge (d.h. kleinstes Element nach links), die Zeilen mit ungeradem Index in absteigender Reihenfolge.
- Sortiere parallel alle Spalten, so daß die jeweils kleinsten Elemente oben stehen.



- (a) Zeigen Sie die Korrektheit des Algorithmus unter der Voraussetzung, dass nur Nullen und Einsen sortiert werden sollen.

*Hinweis:* Eine *unreine* Zeile sei eine Zeile, die sowohl Nullen als auch Einsen enthält. Zeigen Sie durch Betrachtung übereinanderliegender Zeilen, dass sich ab dem zweiten Durchlauf der Zählschleife die Anzahl der unreinen Zeilen in jeder Sortierphase halbiert.

- (b) Welcher Algorithmus wird sinnvollerweise zum Sortieren der Zeilen und Spalten verwendet? Berechnen Sie (asymptotisch) den Aufwand des gesamten Sortieralgorithmus und vergleichen Sie ihn mit der unteren Schranke für das Sortieren auf einem zweidimensionalen Gitter.

Bitte wenden!

### 5.3 Verteilter Workpool

5 Punkte

In der Vorlesung wurde ein Programm vorgestellt, das mit Hilfe eines verteilten Workpools das „Kürzeste Wege“-Problem für Graphen berechnet. Es wird von dem Programm eine 1:1-Relation zwischen Prozessen und Graphknoten erzeugt.

- (a) Modifizieren Sie das Programm, so dass jeder Prozess für mehrere Knoten zuständig ist. Außerdem sollte jede virtuelle Maschine auf einem separaten Rechner (Host) erzeugt werden und genau einen Prozess erhalten.
- (b) Messen Sie die Laufzeiten mit 1, 2, 4 und 8 Rechnern. Begründen Sie das Laufzeitverhalten Ihres Programms.

*Hinweis:* Benutzen Sie die auf der VL-Seite bereitgestellte Datei `adjMatrix.txt` und den Linuxbefehl `time` für die Zeitmessung. Achten Sie darauf, dass die Zeitmessung nicht durch Debug-Ausgaben verfälscht wird.