PHILIPPS-UNIVERSITÄT MARBURG
Fachbereich Mathematik und Informatik
Prof. Dr. R. Loogen
Dipl.Inf. J.Berthold

D-35032 Marburg
Hans Meerwein Straße

16. Juni 2005

Übungen zu „Parallele Programmierung“, SS 2005

Nr. 10, Abgabe der Aufgaben: ab 27. Juni 2005 in den Tutorien

---

Hinweis:
Dieses Übungsblatt hat eine verlängerte Bearbeitungszeit und wird bereits bei Abgabe der Lösungen (ab **27. Juni 2005**) besprochen.
Am 4. und 5. Juli findet kein Tutorium statt.

Aufgaben

10.1 Untere Schranken für paralleles Sortieren — 3 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

(c) $\Omega(\log n)$ auf einem Shuffle-Exchange-Netzwerk

10.2 Bitonisches Sortieren — 6 Punkte

Eine Variante des in der Vorlesung besprochenen *Odd-Even-Merge Sort* ist das *bitonische Sortieren*. Eine Folge von ganzen Zahlen $(a_i)_{i\in\{0\ldots n-1\}}$ heißt *bitonisch*, falls sie durch eine zyklische Verschiebung in eine aufsteigenden Teilfolge gefolgt von einer absteigenden Teilfolge transformiert werden kann, d.h.:

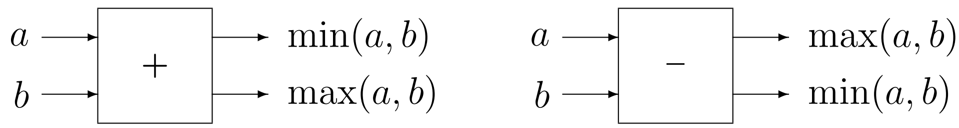
1. $\exists i_0 \in \{0\ldots n-1\}: \quad a_0 \leq a_1 \leq \ldots \leq a_{i_0} \geq a_{i_0+1} \geq \ldots \geq a_{n-1}$

oder 2. $\exists k \in \{0\ldots n-1\}:$ die Folge $(a_{(i+k) \mod n})_{i\in\{0\ldots n-1\}}$ (zyklische Verschiebung um k) erfüllt Bedingung 1

Eine zweielementige Folge ist stets bitonisch.

(a) Beweisen Sie den folgenden **Satz**:
Seien $n = 2^k$ mit $k > 0$ und $(a_i)_{i\in\{0\ldots n-1\}}$ bitonische Folge.
Dann sind die Folgen $b_i = \min(a_i, a_{\frac{n}{2}+i})$ und $c_i = \max(a_i, a_{\frac{n}{2}+i})$ *ebenfalls bitonisch* (wobei $(i \in \{0\ldots \frac{n}{2}-1\})$). Weiter gilt: $b_j \leq c_k \forall j, k \in \{0\ldots \frac{n}{2}-1\}$
Sie können zur Vereinfachung von einer Liste ohne Verschiebung um k Stellen ausgehen.

(b) Skizzieren Sie ein Sortiernetzwerk, das mit aufsteigend und absteigend sortierenden *Komparatorbausteinen*



aus einer beliebigen Folge zunächst eine bitonische Folge herstellt und diese danach (nach Aussage des Satzes) rekursiv sortiert.

10.3 Sortieren auf einem zweidimensionalen Gitter

6 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 ≥ 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.

13	11	1	9
2	7	4	0
12	5	10	3
15	8	6	14

$\Rightarrow$

1	9	11	13
7	4	2	0
3	5	10	12
15	14	8	6

$\Rightarrow$

1	4	2	0
3	5	8	6
7	9	10	12
15	14	11	13

$\Rightarrow \ldots \Rightarrow$

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12

(a) Benutzen Sie das 0-1-Prinzip, um die Korrektheit des Algorithmus zu zeigen.

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.