

Übungsblatt 7

Abgabe bis Dienstag, den **24. Juni** um 12:00 Uhr

Aufgabe 1 (10 Punkte)

Nehmen Sie Folgendes an, entsprechend den Erklärungen aus der Vorlesung: Der langsame und der schnelle Speicher sind aufgeteilt in Blöcke der Größe $B = 4$. Der schnelle Speicher hat Größe $M = 8$, kann also zwei Blöcke fassen. Im langsamen Speicher steht ein Feld A mit 12 Elementen. Die Elemente stehen (wie bei einem Feld üblich) hintereinander im Speicher. Das Element $A[0]$ steht am Beginn eines Blockes und damit stehen auch die Elemente $A[4]$ und $A[8]$ jeweils am Beginn eines Blockes. Wenn auf ein Element aus A zugegriffen wird (ob es gelesen oder geschrieben wird, ist für diese Aufgabe egal), wird gleich der ganze Block, der das Element enthält, in den schnellen Speicher gelesen. Es wird dann der Block aus dem schnellen Speicher geworfen, auf den am längsten nicht mehr zugegriffen wurde („*least recently used*“).

Nehmen wir jetzt beispielsweise an, es wird nacheinander auf die folgenden zehn Elemente des Feldes zugegriffen (es spricht nichts dagegen, mehrfach auf dasselbe Element zuzugreifen): $A[2]$, $A[0]$, $A[8]$, $A[4]$, $A[9]$, $A[7]$, $A[3]$, $A[6]$, $A[3]$, $A[0]$. Nehmen wir außerdem an, dass der schnelle Speicher zu Beginn leer ist. Vergewissern Sie sich (am besten anhand einer Abbildung), dass dann genau *vier* Blockoperationen benötigt werden, das heißt, es wird genau vier mal ein Block in den schnellen Speicher geladen, der dort zum Zeitpunkt des Zugriffs aus das Element noch nicht bzw. nicht mehr stand. Machen Sie mit der Aufgabe nicht weiter, bevor Sie nicht verstanden haben, warum es genau vier Blockoperationen sind.

Bestimmen Sie dann für jedes $n \in \{1, \dots, 10\}$ eine Folge von zehn Elementen des Feldes, so dass wenn auf die Elemente in dieser Folge zugegriffen wird, die Anzahl der benötigten Blockoperationen genau n ist. Die Folge aus dem Absatz vorher würde diese Bedingung also für $n = 4$ erfüllen.

Aufgabe 2 (10 Punkte)

Zeigen Sie, dass der in Vorlesung 1 erklärte *MergeSort* Algorithmus $\Theta(n/B \cdot \log n)$ Blockoperationen benötigt. Zeigen Sie dann, dass die in Vorlesung 7 vorgestellte Variante des *MergeSort* Algorithmus $\Theta(n/B \cdot \log(n/B))$ Blockoperationen benötigt.

[bitte innerhalb desselben Blocks wenden]

In beiden Fällen ist n die Eingabegröße und B die Blockgröße. Sie können dabei der Einfachheit halber annehmen, dass n und B Zweierpotenzen sind und dass das Eingabefeld an einer Blockgrenze beginnt. Sie können außerdem annehmen, dass $n \gg M$ (wobei M die Größe des schnellen Speichers ist), dass $M \geq 2B$, dass ein Listenelement komplett in eine Speicherzelle passt, und dass für Hilfsvariablen (z.B. *middle*, *right*) keine Blockzugriffe benötigt werden (stellen Sie sich z.B. vor, dass diese Variablen in Registern stehen).

Hinweis: In der Vorlesung wurde die Anzahl Blockoperation für jeden der beiden Algorithmen für eine Beispielergabe bestimmt. Machen Sie mit der Aufgabe nicht weiter, bevor Sie nicht genau verstanden haben, warum die Anzahl Blockoperationen so ist wie in der Vorlesung angegeben. Wenn Sie Verständnisprobleme haben, spielen Sie jeden der beiden Algorithmen an einem eigenem Beispiel durch und zählen Sie dabei die Blockoperationen. Bei Problemen fragen Sie bitte im Forum.

Committen Sie Ihre Lösung (als PDF) und Ihre *erfahrungen.txt* (mit den üblichen Informationen) in unser SVN, in einen neuen Unterordner *blatt-07*.

Wie gefällt Ihnen bzw. hat Ihnen die lange Pause zwischen V7 und V8 gefallen?