

Übungen zu Algorithmen

Wintersemester 2003/2004

Blatt 8

Die Klausur findet am Dienstag, den 3.2.2004 im Raum 01/E01+E02 in der Zeit von 08:00 - 10:00 Uhr statt!

Aufgabe 8.1 (15 Punkte)

Gegeben sei die Zahlenfolge 7, 4, 8, 9, 3, 2, 10, 12, 14, 6, 0, 6, die aufsteigend sortiert werden soll. Stellen Sie die Arbeitsweise von `QuickSort.sort()` in `/home/ainf/Vorlesung/QuickSort.java` durch Zwischenergebnisse geeignet dar (vgl. Aufgabe 7.3).

Aufgabe 8.2 (15 Punkte)

- Wieviele Vergleiche von 2 Elementen und wieviele rekursive Aufrufe führt `QuickSort.sort()` beim Sortieren der - bereits sortierten - Folge 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 durch?
- Geben Sie eine Folge der 10 ganzen Zahlen 1 bis 10 an, die den Worst Case bezogen auf die Zahl der Vergleiche zweier Elemente für `QuickSort.sort()` in `/home/ainf/Vorlesung/QuickSort.java` darstellt. Belegen Sie Ihre Behauptung durch die Angabe von Zwischenergebnissen beim Sortieren dieser Folge.

Aufgabe 8.3 (30 Punkte)

a) Implementieren Sie eine Klasse `ElementSelection`, die eine Methode `select(int[] s, int k)` enthält, welche das k -t kleinste Element im Array `s` liefert. Verwenden Sie die Methode `QuickSort.sort`, wenn Sie den Median einer Menge bestimmen müssen, die weniger als 50 Elemente enthält. Bedenken Sie, dass die Anzahl der Elemente in `s` evtl. nicht restlos durch 5 teilbar ist. Überprüfen Sie Ihre Implementation mit Hilfe der Klasse `/home/ainf/Uebung/Blatt8/Aufg8.3.java`.

Wenn Ihre Lösung korrekt ist, wird beim Ausführen der Klasse `Aufg8.3` nur die Begrüßung ausgegeben. Um die Klasse `Aufg8.3` übersetzen und ausführen zu können, benötigen Sie die Klasse `/home/ainf/Uebung/Blatt8/FeldFueller.class`.

b) Kopieren Sie die Klasse `/home/ainf/Vorlesung/QuickSort.java` in Ihr Arbeitsverzeichnis, benennen Sie sie in `QuickSortMedian` um und verändern Sie die Methode `sort`, so dass der Median als Pivotelement gewählt wird. Überzeugen Sie sich vom Laufzeitvorteil dieser Strategie, indem Sie die Klassen `/home/ainf/Uebung/Blatt8/SpeedTest.class` und `/home/ainf/Uebung/Blatt8/FeldFueller.class` in Ihr Arbeitsverzeichnis kopieren und die Klasse `SpeedTest` ausführen. Dokumentation zur Klasse `SpeedTest` finden Sie in `/home/ainf/Uebung/Blatt8/SpeedTest.html`.

Aufgabe 8.4 (20 Punkte)

Stellen Sie für die gegebene unsortierte Folge mit zwölf Elementen: 7, 4, 8, 9, 3, 2, 10, 12, 14, 6, 0, 6, die Arbeitsweise von `HeapSort.sort()` in `/home/ainf/Vorlesung/HeapSort.java` durch die Angabe von Zwischenergebnissen geeignet dar. Stellen Sie jeweils den Inhalt des Arrays auch als binären Baum dar. Beachten Sie dabei die beiden Phasen des Algorithmus: Aufbau des Heaps und Sortieren der Folge.

Aufgabe 8.5 (20 Punkte)

Geben Sie zwei Folgen der acht ganzen Zahlen 1, 3, 4, 7, 8, 10, 12, 15 an, für die `HeapSort.sort()` in `/home/ainf/Vorlesung/HeapSort.java` die meisten bzw. die wenigsten Vertauschungen von zwei Elementen im Array durchführt. Demonstrieren Sie exemplarisch für eine Best- und eine Worst-Case-Folge den Sortiervorgang und geben Sie die Anzahl der benötigten Vertauschungen an.

Aufgabe 8.6 (0 Punkte)

In UNIX gibt es die Möglichkeit, Dateien und Kataloge zu „verstecken“, so daß sie durch ein einfaches `ls` nicht angezeigt werden. Solche Namen beginnen mit einem Punkt. Um diese Dateien und Kataloge trotzdem zu sehen, gibt es die Option `-a` für das `ls`-Kommando (s. Tabelle). Im Heimatverzeichnis gibt es mehrere solcher Dateien. Viele davon enden auf `rc` und sind Ressource-Dateien, die Einstellungen an Programmen vornehmen, z.B. `.exrc` für den `vi`.

Als Multiuser-Betriebssystem braucht UNIX Schutzrechte, die Daten vor dem Zugriff von Unbefugten schützen. Die Rechte an Dateien und Katalogen sieht man mit dem Kommando `ls -l`. Es gibt drei Benutzerarten mit eigenen Rechten, die der Besitzer der Datei oder des Katalogs mit dem Kommando `chmod` (s. Tabelle) einstellen kann. Die drei Benutzerarten sind:

- der Besitzer, englisch *user*, Kürzel `u`
- die Gruppe, englisch *group*, Kürzel `g`
- die Welt, englisch *others*, Kürzel `o`

- Pro Benutzerart gibt es drei Rechte:
- lesen, englisch *read*, Kürzel `r`
 - schreiben, englisch *write*, Kürzel `w`
 - ausführen, englisch *execute*, Kürzel `x`

Bei Katalogen bedeutet das `x`-Bit, daß der Katalog zugänglich ist, das `w`-Bit, daß Dateien angelegt und gelöscht werden können und das `r`-Bit, daß der Inhalt eines Katalogs aufgelistet werden kann. Zum Ändern der Rechte mit `chmod` gibt es mehrere Möglichkeiten, die in der Tabelle aufgeführt sind. Dabei können die 9 Bits `rwrxrwxrwx` auch als Oktalzahl in 3er-Gruppen angegeben werden. Machen Sie sich in diesem Zusammenhang mit den folgenden UNIX-Kommandos vertraut:

Kommando	Erklärung
<code>ls -R</code>	Listet Dateien und Kataloge <i>rekursiv</i> , d.h. auch die Dateien in ihnen.
<code>ls -rtl</code>	Listet Dateien und Kataloge in zeitlicher Reihenfolge (<i>reverse time long</i>).
<code>ls -a</code>	Listet auch „versteckte“ Dateien und Kataloge, die mit <code>.</code> beginnen (<i>all</i>).
<code>chmod 644 datei</code>	Schaltet die Rechte der Datei <code>datei</code> auf <code>-rw-r--r--</code> .
<code>chmod 700 Dir</code>	Schaltet die Rechte des Verzeichnisses <code>Dir</code> auf <code>drwx-----</code> .
<code>chmod o-rw Aufg.java</code>	Verwehrt der Welt Schreib- und Leserechte an <code>Aufg.java</code> .
<code>chmod ug-w Aufg.java</code>	Verwehrt dem Besitzer und der Gruppe die Schreibrechte an <code>Aufg.java</code> .