

Übungen zu Algorithmen und Datenstrukturen Aufgabenblatt 3

Prof. Dr. Steffen Staab,
Dipl.–Inform. Gerd Gröner und Dipl.–Inform. Christoph Ringelstein
Arbeitsgruppe ISWeb, Fachbereich Informatik, Universität Koblenz–Landau

Abgabe: 19.11.2008, bis 23:59 Uhr

1 Heapsort (6 Punkte)

Implementieren Sie den in der Vorlesung vorgestellten Sortieralgorithmus Heapsort. Als Eingabe erhält der Algorithmus einen Array von zu sortierenden Zahlen. Das Ergebnis ist die aufsteigend sortierte Folge der Eingabezahlen. Realisieren Sie den Algorithmus objektorientiert. Verwenden Sie selbstdefinierte Klassen für balancierte, linksbündige Binärbäume. Ihr Programm soll gut kommentiert sein.

Programmeingabe: Array von Zahlen

Programmausgabe: Array mit aufsteigend sortierter Zahlenfolge

2 Quicksort (3 Punkte)

In der Vorlesung wurde der Sortieralgorithmus QuickSort vorgestellt. Stellen Sie den Zustand des Arrays beim Eintreten und Verlassen der Zerlegefunktion für jeden Aufruf dar. Als Eingabe erhält der Algorithmus die Zahlenfolge:

45 78 38 14 63 24 103 51 12 85

Jeweils das erste Element der zu sortierenden Folge (Teilfolge) dient als Pivotelement. Beim ersten Zerlegungsschritt ist somit 45 das Pivotelement.

3 Fibonacci-Zahlen (3 Punkte)

In der Vorlesung wurde ein rekursiver Algorithmus zur Berechnung der Fibonacci-Zahlen vorgestellt:

```
fib(x) := if (x == 0) or (x == 1) then 1
         else fib(x-2) + fib(x-1) fi
```

Eine alternative Berechnung ist mittels Memorisierung möglich:

FIB:

```
var f_n1, f_n2, f_n, k, x : int
input x
f_n2 := 0;
f_n1 := 1;
for k := 2 to x do
    f_n := f_n2 + f_n1
    f_n2 := f_n1
    f_n1 := f_n
od
if (x <= 1) then output 1
else output f_n
fi
```

Bestimmen Sie für beide Verfahren die Anzahl der Additionsoperationen. Die Anzahl der Additionsoperationen soll allgemein in Abhängigkeit der Eingabe $x \in \mathcal{N}$ angegeben werden.

Wichtige Hinweise:

- Die SVN-Repositories sind unter <http://lehre2.isweb.uni-koblenz.de/svn/aud-ws0809/> zu erreichen, gefolgt vom jeweiligen Gruppennamen.
- Bitte checken Sie Ihr Programm und Ihre Texte und Grafiken ins SVN-Repository ein. Nutzen Sie als Verzeichnisname direkt unterhalb Ihres Gruppenverzeichnisses *zwingend*:

blatt03

Nur so können die Korrektoren Ihre Implementierung aus dem SVN-Repository auslesen und die Punkte verteilen.

- Achten Sie bitte darauf, dass Ihr Programm fehlerfrei übersetzt werden kann. Die Korrektoren werden Ihren Quellcode ebenfalls übersetzen und ausführen.
- Es werden nur Programme berücksichtigt, die in Java implementiert sind und direkt kompiliert werden können.

- **Bitte checken Sie KEINE .class-Dateien ein!**
- Wenn Sie Grafiken oder Textdokumente mit abgeben möchten, so verwenden Sie bitte als Dateiformat:
 - Für schematische Zeichnungen: .png oder .gif
 - Für Scans oder Photos: .png oder .jpg
 - Für Text: Plaintext (.txt) oder PDF-Dateien (.pdf). Natürlich dürfen Sie die Grafiken auch in eine PDF-Datei einbetten.

Andere Formate werden bei der Korrektur *nicht* berücksichtigt!

- Verwenden Sie bitte keine Umlaute, Leerzeichen oder Sonderzeichen in Dateinamen (auch nicht bei .pdf oder .png Dateien)!