

4. Übung:

Algorithmen und Datenstrukturen

Sommersemester 2008

30. April 2008

Abgabe bis Donnerstag, 8. Mai 2008, vor der Übung

Aufgabe 4.1:

(Punkte 5)

Gegeben sei ein Feld $A[1..n]$ von Zahlen. Der folgende (sehr einfache) Algorithmus wird BUBBLESORT genannt.

```
for r = n downto 1
do for i=1 to r-1
  do if A[i] > A[i+1] then
    vertausche A[i] und A[i+1]
  fi
od
od
```

Zeigen Sie, dass diese Implementierung das Feld A aufsteigend sortiert (in Prosa) und analysieren Sie die Laufzeit.

Aufgabe 4.2:

(Punkte 5)

Entwickeln Sie einen Algorithmus $\text{SELECT}(A, k)$, der für ein Feld $A[1..n]$ von Zahlen, das k -t größte Element berechnet. Kann man das Problem lösen ohne das Feld zu sortieren? Hinweis: Betrachten Sie folgende Idee für einen rekursiven Algorithmus. Ordne das Feld A so um, dass alle Zahlen, die kleiner oder gleich $A[1]$ sind, vor den restlichen Zahlen stehen. Wende dann denselben Algorithmus (rekursiv) auf den ersten oder den zweiten Abschnitt an.

Aufgabe 4.3:

(Punkte 4/3/3)

- a) Schreiben Sie ein Program (in Pseudo-Code oder einer Programmiersprache), das den Inhalt eines Felds A in einen dynamischen binären Baum B sortiert. Für alle Teilbäume $(T1, x, T2)$ mit der Wurzel x in B soll gelten:

- Für den Wert x_1 der Wurzel des linken Teilbaumes T_1 ist $x_1 \leq x$.
- Für den Wert x_2 der Wurzel des rechten Teilbaumes T_2 ist $x_2 > x$.

Nehmen Sie an das Feld A mit ganzen Zahlen gefüllt ist. Der erzeugte Baum ist höchstwahrscheinlich nicht vollständig.

Benutzen Sie eine Struktur der Form:

```
bin_tree_node {
  int value;
  bin_tree_node* left_child;
  bin_tree_node* right_child;
}
```

- b) Analysieren Sie die Laufzeit im schlechtesten und im besten Fall.
- c) Geben Sie ein Programm an, das einen Baum, der die Regeln aus a) erfüllt, in einen Array abbildet. Geben Sie die Laufzeit ihres Algorithmus an.