Institut für Informatik I
Prof. Dr. Heiko Röglin
Tobias Brunsch

universitätbonn

Algorithmen und Berechnungskomplexität I
Wintersemester 2010/11
Abgabe: 17.11.2010 in der Vorlesung

Übungsblatt 5

Aufgabe 5.1

4+2+2+1 Punkte

Wir betrachten folgende Variante des Sortierverfahrens *Bubblesort*, das ein Feld A mit n Einträgen aufsteigend sortiert.

```
repeat
  swapped = false
  for i = 1,...,n-1 do
    if A[i] > A[i+1] then
      Vertausche A[i] und A[i+1]
      swapped = true
    end if
  end for
until not swapped
```

(a) Sei $\chi(A)$ die Anzahl der Inversionen in A, d.h.

$$\chi(A) := |\{(i,j) : 1 \leq i < j \leq n \text{ und } A[i] > A[j]\}|.$$

Zeigen Sie, dass *Bubblesort* genau $\chi(A)$ Vertauschungen durchführt.

(b) Beweisen Sie, dass obiges Verfahren nach endlich vielen Schritten abbricht und dass das Feld A zu diesem Zeitpunkt sortiert ist.

(c) Wie viele Vergleiche werden im besten und im schlechtesten Fall benötigt? Geben Sie jeweils ein Beispiel an, für das entsprechend viele Vergleiche durchgeführt werden.

(d) Obige Variante von *Bubblesort* führt bis zu 50% überflüssige Vergleiche durch. Erklären Sie, warum.

Aufgabe 5.2

4+1+2 Punkte

Quicksort ruft eine Partitionsfunktion `Partition` mit den Parametern A (Feld) sowie l und r (Indizes) auf. Diese vertauscht einige Einträge des Feldes A und gibt einen Wert `pos` zurück. Das Feld A hat dann folgende Eigenschaften:

1. Alle Einträge $A[l], \ldots, A[\texttt{pos}-1]$ sind nicht größer als $A[\texttt{pos}]$.
2. Alle Einträge $A[\texttt{pos}+1], \ldots, A[r]$ sind nicht kleiner als $A[\texttt{pos}]$.

(a) Zeigen Sie, dass der folgende Code eine korrekte Implementierung der `Partition`-Funktion ist.

```
Partition(A, l, r)
{
  pos = l
  if l < r then
    pivot = A[l]
    for i = l+1,...,r do
      if A[i] < pivot then
        pos = pos + 1
        Vertausche A[pos] und A[i]
      end if
    end for
  Vertausche A[l] und A[pos]
  end if
  return pos
}
```

(b) Bestimmen Sie die Laufzeit obiger Implementierung in Θ-Notation bzgl. $n := r - l + 1$.

(c) Geben Sie eine einfache Implementierung der `Partition`-Funktion mit derselben Laufzeit an, die nur $O(1)$ Vertauschungen durchführt. Warum ist diese nicht für Quicksort geeignet?

Aufgabe 5.3

5+2+1 Punkte

Professor G. Mächlich möchte eine Variante seines Lieblingssortierverfahrens *Bubblesort* mit Hilfe eines Divide-and-Conquer-Ansatzes beschleunigen und hat sich folgendes Verfahren überlegt. Sei $k \geq 2$ eine natürliche Zahl (nicht Teil der Eingabe) und A ein Feld mit $n = 2 \cdot k^l$ Einträgen für eine nichtnegative ganze Zahl $l \in \mathbb{N}_0$.

```
FastBubblesort(A, l, r)
{
  n̂ = r − l + 1                                   // Anzahl der Einträge

  if n̂ = 2 then                                   // Rekursionsabbruch
    if A[l] > A[r] then Vertausche A[l] und A[r]
    return
  end if

  // „Zerlege“ A in m Teilfelder der Größe s mit:
  m = 2k
  s = n̂/m

  // Modifiziertes Bubblesort:
  for i = m − 2, ..., 0 do                        // von m − 2 bis 0 herunterzählen
    for j = 0, ..., i do                          // von 0 bis i hochzählen
      l̂ = l + j · s                               // erster Index von Teilfeld j
      r̂ = l̂ + 2 · s − 1                           // letzter Index von Teilfeld j + 1
      FastBubblesort(A, l̂, r̂)                     // Teilfelder j und j + 1
    end for                                       // gemeinsam sortieren
  end for
}
```

(a) Zeigen Sie induktiv über den Exponenten l, dass das Feld A durch den Aufruf `FastBubblesort`$(A, 1, n)$ aufsteigend sortiert wird.

Hinweis: Formulieren Sie im Induktionsschritt eine geeignete Invariante für das Ende von Schleifendurchlauf i. Erklären Sie, warum diese Invariante korrekt ist. Auf einen ausführlichen Invarianzbeweis kann an dieser Stelle verzichtet werden.

(b) Geben Sie die Laufzeit von `FastBubblesort` in Abhängigkeit von n und der Konstante k in Θ-Notation an und vergleichen Sie diese mit der Laufzeit von *Bubblesort*. Ist der Name `FastBubblesort` gerechtfertigt?

(c) Wie kann man unter Verwendung von `FastBubblesort` ein Feld der Länge $n \neq 2 \cdot k^l$ sortieren?