

Übungen zur Vorlesung Einführung in die Informatik II

Ausgabe: 9. Juli 2002

Abgabe: 15. Juli 2002, 13.00 Uhr

Die Abgabe erfolgt ausschließlich über das System eClaus! Siehe
http://www.informatik.uni-stuttgart.de/ifi/fk/lehre/ss02/info_II.02.html

Aufgabe 1: (Heapsort-Variante)**schriftlich**

Untersuchen Sie die **heapsort**-Prozedur aus dem Skript Claus, Kap. 6.1 Folie 10, mit folgender modifizierter **sink**-Prozedur (vgl. Folie 9 – gegenüber der dort beschriebenden Variante wurde der Sonderfall **if j=rechts** ignoriert und der Abbruch der **while**-Schleife entsprechend angepasst).

```

procedure sink (links, rechts: 1..n) is
  i, j: Natural; v: <Elementtyp>;
begin v:=A(links); i:=links; j:=i+i;
  while j<=rechts-1 loop
    if A(j) < A(j+1) then
      if v < A(j+1) then A(i):=A(j+1); A(j+1):=v; i:=j+1;
      else A(i):=v; i:=rechts; end if;
    else
      if v < A(j) then A(i):=A(j); A(j):=v; i:=j;
      else A(i):=v; i:=rechts; end if;
    end if;
    j:=i+i;
  end loop;
end sink;

```

- (3 Punkte) Was tut der **heapsort**-Algorithmus, wenn obige modifizierte **sink**-Prozedur verwendet wird? Wie muss die **heapsort**-Prozedur geändert werden, damit der Algorithmus das gegebene Feld **A** doch richtig sortiert? (Die Prozedur **sink** soll dabei wie oben unverändert bleiben.)
- (1 Punkt) Schätzen Sie ab, wie groß die Laufzeitunterschiede zwischen diesen beiden Varianten sind.

Aufgabe 2: (Sortierverfahren)**zum Votieren**

(5 Punkte) Sortieren Sie die Zahlenfolge 5, 8, 1, 6, 3, 4, 9, 10, 4, 5, 4, 7

- mit *Fachverteilen (radix exchange)*, betrachten Sie hierzu die Zahlen in Binärdarstellung,
- mit *Shellsort*, verwenden Sie dabei als Schrittweite die Folge 4, 2, 1,
- mit *Mergesort*,
- mit *natürlichem Einphasenmischen*,
- mit *Bucketsort* in den Grenzen von 0 bis 11.

Geben Sie die Zahlenfolgen bei allen Zwischenschritten an.

Aufgabe 3: (Stabilität von Sortierverfahren)**zum Votieren**

(2 Punkte) Welche der in der Vorlesung vorgestellten Sortierverfahren (vgl. Skript Claus, Kap. 6.1 Folie 15) sind stabil, welche nicht? Begründen Sie Ihre Antworten.

Aufgabe 4: (Median-Bestimmung)**schriftlich**

Eine mit dem Sortieren verwandte Aufgabe ist die Bestimmung des *Medians* einer Folge von Zahlen. Eine Möglichkeit ist es, die Zahlen zu sortieren. Dann steht der Median in der Mitte der sortierten Folge (Index $1+r \operatorname{div} 2$). Es ist jedoch nicht notwendig, die Folge vollständig zu sortieren, um den Median zu finden.

- a. (5 Punkte) Entwickeln Sie aus dem Quicksort-Algorithmus einen effizienten (rekursiven) Algorithmus zur Bestimmung des k -größten Elementes (der Aufruf mit $k := 1+r \operatorname{div} 2$ liefert dann den Median). Programmieren und testen Sie Ihren Algorithmus an der Zahlenfolge aus Aufgabe 2, geben Sie die Prozeduraufrufe und Zahlenfolgen bei allen Zwischenschritten an.
- b. (3 Punkte) Geben Sie Beispiele für den *best case* und *worst case* an. Ermitteln Sie experimentell den Aufwand im Mittel in Abhängigkeit der Feldgröße.

Aufgabe 5: (Isomorphie von Bäumen)**schriftlich**

Zwei gerichtete Bäume $((V, E)$ und (V', E') heißen isomorph, wenn es eine bijektive Abbildung $f : V \rightarrow V'$ gibt, so dass für alle (u, v) in E gilt: (u, v) liegt genau dann in E , wenn $(f(u), f(v))$ in E' liegt.

- a. (3 Punkte) Finden Sie einen Algorithmus, um zu zwei gerichteten Bäumen (V, E) und (V', E') zu entscheiden, ob die beiden Bäume isomorph sind. Geben Sie die Laufzeit Ihres Algorithmus an.
- b. (5 Punkte) Finden Sie einen solchen Algorithmus, der in Zeit $O(n \log(n))$ oder sogar in $O(n)$ arbeitet, wobei n die Anzahl der Knoten in V ist.