F3 – Berechenbarkeit und Komplexität

Aufgabenzettel 7 : Algorithmen

Besprechung in der Zeit vom 8.12. bis zum 11.12.2003.

Präsenzaufgabe 7 :

(a) Wieviel Zeit benötigt man maximal, um festzustellen, ob eine beliebige Zahl $k \in \mathbb{N}$ in einer Liste von n natürlichen Zahlen vorkommt?

(b) Beurteilen Sie die Aussage: „Rekursive Algorithmen sind immer effizienter als andere Verfahren.“

Übungsaufgabe 7.1 :

von
6

(a) Beschreiben Sie ein rekursives Verfahren zur Bestimmung der Höhe eines Binärbaums. Benutzen sie eine programmiersprachenähnliche Notation und nehmen sie an, dass die Eingabe des beliebigen (also nicht immer ausgeglichenen) Binärbaums B in Form einer doppelt verketteten Liste erfolgt. (Sie brauchen nicht zu testen, ob die endliche Liste B in der nachstehend beschriebenen Darstellung, tatsächlich einen Binärbaum darstellt!) Dies bedeutet (hier in einer für diese Aufgabe vereinfachten Form), dass ein innerer Knoten x durch das Tripel $(\text{linkerNachfolger}, x, \text{rechterNachfolger})$ und ein Blatt z durch $(\text{void}, z, \text{void})$ kodiert ist. Die Wurzel wird als $(\text{linkerNachfolger}, \text{root}, \text{rechterNachfolger})$ gespeichert und ist immer Element von B. Als Befehle stehen Ihnen die üblichen arithmetischen Befehle (Addition, Subtraktion, Multiplikation, Minimum zweier Werte und Maximum zweier Werte) auch Vergleiche (kleiner, größer, gleich und deren Kombinationen) zur Verfügung. Für die Behandlung der Liste und deren Elemente können Sie nachfolgende Anweisungen bzw. Abfragen benutzen, sind aber nicht verpflichtet alle zu verwenden:

1. `wähle linken Nachfolger als neuen (Wurzel-)knoten` [als **left** abgekürzt],
2. `wähle rechten Nachfolger als neuen (Wurzel-)knoten` [als **right** abgekürzt],
3. `wähle Vorgängerknoten` [als **up** abgekürzt] ,

sowie die drei Testbefehle:

4. `gibt es einen linken Nachfolger?` [als **li-succ** : $B \to \{\text{ja}, \text{nein}\}$ abgekürzt],
5. `gibt es einen rechten Nachfolger?` [als **re-succ** : $B \to \{\text{ja}, \text{nein}\}$ abgekürzt],
6. `ist dies ein Blatt?` [als **leaf** : $B \to \{\text{ja}, \text{nein}\}$ abgekürzt].

(4 Pkt.)

(b) Geben Sie die Komplexität Ihres Verfahrens als Funktion der Größe $|B|$ an.

(2 Pkt.)

Übungsaufgabe 7.2 :

von
8

Sei $k \in I\!N, k \neq 0$, eine Konstante. Geben Sie jeweils einen Algorithmus mit den geforderten Eigenschaften (a) bzw. (b) an, der die hier definierte arithmetische Funktion $f : I\!N \to I\!N$ berechnet:

$$f(n) := \begin{cases} 1 & \text{für } n = 0, \\ k & \text{für } n = 1, \\ 2f(n-1) - f(n-2) & \text{sonst.} \end{cases}$$

Beschreiben Sie auch hier die Verfahren in einer programmiersprachenähnlichen Notation.

(a) rekursiv (2 Pkt.)

(b) iterativ (2 Pkt.)

Begründen Sie kurz (mit höchstens einem Satz) welche der folgenden Eigenschaften die von Ihnen für (a) und (b) beschriebenen Verfahren haben:

(c) effektiv (1 Pkt.)

(d) effizient (1 Pkt.)

Geben Sie einen effizienteren Algorithmus an, der ohne Rekursion und Iteration auskommt, oder begründen Sie, dass es kein effizienteres Verfahren gibt, als eines Ihrer Verfahren aus (a) oder (b). (2 Pkt.)

Bisher erreichbare Punktzahl: 86