Technische Universität München — WS 2001/2002
Institut für Informatik — Übungsblatt 3
Lehrstuhl für Effiziente Algorithmen — 7. November 2001

Grundlegende Algorithmen

Abgabe: bis 14. November, 16:00 Uhr, Briefkasten bei S0314

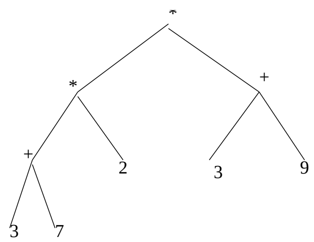
Hinweis: Unkommentierter Pseudocode wird nicht gewertet!

Aufgabe 1 (10 Punkte)

Ein arithmetischer Baum A ist durch die Datenstruktur

class treenode
  treenode left, right
  operator op
  int operand
end

und durch die beiden folgenden Regeln definiert:
(1) Falls A aus einen Knoten besteht,dann ist $A.op$ ='*const*' und der Wert des Baumes = $A.operand$.
(2) Andernfalls sind $A.left$ und $A.right$ arithmetische Bäume, $A.op$ = '+' oder $A.op$ = '$*$' und der Wert des Baumes ist Wert des linken Teilbaums $A.op$ Wert des rechten Teilbaums.
Jeder arithmetische Baum entspricht einen arithmetischen Ausdruck. Der Baum A



entspricht $(3 + 7) * 2 * (3 + 9)$.
(1) Entwerfen Sie einen arithmetischen Baum mit 5 Operatoren.
Schreiben Sie einen rekursiven Algorithmus der folgende Spezifikation erfüllt:
Eingabe: arithmetischer Baum A.
Rückgabe: Der zu A gehörende Ausdruck als String, korrekt geklammert.
Wenden Sie Ihren Algorithmus auf Ihren Baum und auf A an und beschreiben Sie den Ablauf.
Hinweis: Der zu A gehörende Ausdruck kann auch so $((3+7)*(2))*(3+9)$ geschrieben werden.

Aufgabe 2 (10 Punkte)

Schreiben Sie einen rekursiven Algorithmus der folgende Spezifikation erfüllt:

Eingabe: arithmetischer Baum A.
Ausgabe: Zahl die durch die Auswertung von A entsteht.
Wenden Sie Ihren Algorithmus auf die Bäume von Aufgabe 1 an und beschreiben Sie den Ablauf.

Aufgabe 3 (10 Punkte)

Schreiben Sie einen Divide and Conquer Algorithmus der folgende Spezifikation erfüllt:
Eingabe: unsortiertes Array $A[]$.
Ausgabe: Minimum und Maximum von $A[]$.
Es darf kein Sortieralgorithmus verwendet werden!
Wenden Sie Ihren Algorithmus auf $[7, 4, 3, 2, 9, 10, 1]$ an. Wieviele Vergleiche benötigt Ihr Algorithmus (mit Begründung).