

Kennzahl: \_\_\_\_\_

**FRÜHJAHR****46111**

Kennwort: \_\_\_\_\_

**1990**

Arbeitsplatz-Nr.: \_\_\_\_\_

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen****- Prüfungsaufgaben -****Fach:** Informatik (nicht vertieft studiert)**Einzelprüfung:** Programmentw./Systempr./Datenbanksys.**Anzahl der gestellten Themen (Aufgaben):** 1**Anzahl der Druckseiten dieser Vorlage:** 2Aufgabe

Es soll eine geschlossene Rechenstruktur für geordnete Binärbäume über den ganzen Zahlen  $\mathbb{Z}$  entwickelt werden. Dabei soll die Ordnung in einem Binärbaum  $B$  folgendermaßen festgelegt sein: Sei  $k$  ein Knoten von  $B$ ,  $L(k)$  bzw.  $R(k)$  der linke bzw. rechte Unterbaum von  $k$  in  $B$  und  $I(k)$  die zu  $k$  gehörige Zahl aus  $\mathbb{Z}$ ; dann gilt:

$$(\forall k \in B)((\forall k_l \in L(k))(I(k_l) < I(k)) \wedge (\forall k_r \in R(k))(I(k) < I(k_r))). \Rightarrow \text{Sohlte 25}$$

Aus dieser Festlegung folgt zugleich, daß die  $I(k)$  in  $B$  paarweise verschieden sind. Schließlich ordnen wir jedem Binärbaum  $B$  in der üblichen Weise seine Höhe  $h(B)$  durch folgende rekursive Definition zu: Ein leerer Binärbaum hat die Höhe 0, und die Höhe  $h(B)$  eines nichtleeren Binärbaums  $B$  mit der Wurzel  $w$  ist

$$h(B) := \max\{h(L(w)), h(R(w))\} + 1.$$

Als Höhe  $h(k)$  eines Knotens  $k$  von  $B$  bezeichnen wir (wieder wie üblich) die Höhe des Unterbaums von  $B$ , dessen Wurzel  $k$  ist.

- a) Geben Sie in PASCAL-Notation eine geeignete Datenstruktur (durch entsprechende Deklarationen) an, mit der geordnete Binärbäume mit den oben angegebenen Festlegungen dargestellt werden können! Die Knoten sollen dabei sowohl ihre zugeordnete Zahl aus  $\mathbb{Z}$  als auch ihre Höhe enthalten. Als Anker eines nichtleeren Baums soll ein Verweis (pointer) auf dessen Wurzel dienen; ein leerer Baum soll durch einen mit *nil* besetzten Verweis als Anker dargestellt werden.

Bevor Sie mit Teilaufgabe b) beginnen, überprüfen Sie, ob die angegebene Datenstruktur für eine möglichst effiziente Durchführung aller in b) geforderten Operationen geeignet ist, und begründen Sie Ihre schließliche Wahl der Datenstruktur im Hinblick auf die in b) vorgesehenen Operationen i) - vi).

INFO 12 - Datenstr.

b) Entwickeln Sie systematisch PASCAL-Prozeduren (bzw. -Funktionen) für die folgenden 6 Operationen an geordneten Binärbäumen! Dabei darf für die Operationen unter ii) - vi) vorausgesetzt werden, daß der jeweilige Binärbaum durch kreiere eingerichtet wurde, und daß er bei Aufruf der jeweiligen Operation die oben angegebenen Integritätsbedingungen (Festlegungen) erfüllt, d.h. daß er in der festgelegten Weise geordnet ist, und daß die Höhen in den Knoten entsprechend der Definition korrekt eingetragen sind.

i) kreiere(t)

Diese Operation richtet einen leeren Binärbaum ein und weist dessen Anker der Variablen t zu.

ii) leer(t)

Diese Operation ist als eine Funktion darzustellen; sie liefert den Wahrheitswert *true*, wenn der Binärbaum mit dem Anker t leer ist; andernfalls liefert die Funktion den Wert *false*.

iii) enthalten(z,t)

Diese Operation ist als eine Funktion darzustellen; sie liefert den Wahrheitswert *true*, wenn der Binärbaum mit dem Anker t einen Knoten mit der Zahl  $z \in \mathbb{Z}$  enthält; andernfalls liefert die Funktion den Wert *false*.

iv) speichere(z,t)

Diese Operation fügt in den Binärbaum mit dem Anker t einen Knoten mit der Zahl  $z \in \mathbb{Z}$  ein, falls der Binärbaum noch keinen Knoten mit z enthält; andernfalls bleibt der Binärbaum unverändert. Der resultierende Binärbaum muß wieder die Integritätsbedingungen erfüllen; sein Anker muß an t zugewiesen sein.

v) loesche(z,t)

Diese Operation entfernt den Knoten mit der Zahl  $z \in \mathbb{Z}$  aus dem Binärbaum mit dem Anker t, falls der Binärbaum einen solchen Knoten enthält; andernfalls bleibt der Binärbaum unverändert. Der resultierende Binärbaum muß wieder die Integritätsbedingungen erfüllen; sein Anker muß an t zugewiesen sein.

vi) avl(t)

Diese Operation ist als eine Funktion darzustellen; sie liefert den Wahrheitswert *true*, wenn der Binärbaum mit dem Anker t ein AVL-Baum ist; andernfalls liefert die Funktion den Wert *false*.

Info für AVL-Baum

Info für AVL-Baum