Übungen zu WS 95/96

Algorithmen und Programmierung 3

- Blatt 8-

H. Schweppe / M. Walter

Abgabe: Mo. 18.12.95 bis 12.00

***Aufgabe 1 (15 P)**

Implementieren Sie für einen Binärbaum die Operationen `SumPathLength(): cardinal`, die als Wert die Summe der Längen aller Wurzelpfade liefert und die Operation `PrintRootUpside(w: CARDINAL)`, die einen (nicht zu großen) Baum so ausgibt, daß die Wurzel oben liegt. Z.B.:

```
            6
      4           8
  3       5       7       9
```

`w` gibt die Breite der Zeichenfläche in Anzahl Zeichen an. Legen Sie die objektorientierte Implementierung zugrunde, die Sie in `~walter/alp3` finden.

***Aufgabe 2 (5 Punkte)**

Zeichnen Sie binäre Suchbäume der Höhen 2, 3, 4, 5 und 6 für die Schlüsselmenge {1,2,5,10,16,17,21}

***Aufgabe 3 (5 Punkte)**

In einem binären Suchbaum sind die Zahlen 1 bis 1000 gespeichert. Gesucht wird nach 363 mit Hilfe der Funktion `IsElement(n:cardinal): boolean`. Welche der folgenden Schlüsselsequenzen können *nicht* in dieser Folge durchsucht worden sein?

a. 2, 252, 401, 398, 330, 344, 397, 363
b. 924, 220, 911, 244, 898, 258, 362, 363
c. 925, 202, 911, 240, 912, 245, 363
d. 2, 399, 387, 219, 266, 382, 381, 278, 363
e. 935, 278, 347, 621, 299, 392, 358, 363

Aufgabe 4 (10 Punkte)

a) Prof. S. Tupid glaubt eine wichtige Eigenschaft binärer Suchbäume entdeckt zu haben. Angenommen, die Suche nach dem Schlüssel k endet erfolgreich in einem Blatt. Folgende 3 Knotenmengen lassen sich dann definieren: A als Menge der Knoten, die links vom Suchpfad liegen, B als Menge der Knoten auf dem Suchpfad zu k und C als Menge der Knoten rechts vom Suchpfad. Die Behauptung von Tupid lautet: für alle $a \in A$, $b \in B$, $c \in C$ gilt: $a \leq b \leq c$. Finden Sie ein möglichst kleines Gegenbeispiel für diese Behauptung.

b) Sei T ein binärer Suchbaum, `x` ein Blattknoten, `y` sein Vaterknoten. Zeigen Sie: `Value(y)` ist entweder der kleinste Schlüssel im Baum größer als `Value(x)` oder der größte Schlüssel in dem Baum, der kleiner `Value(x)` ist.