

Aufgabenblatt 3

Abgabe: 13.05.06, 24.00 Uhr

Bitte geben Sie Ihrem Tutor sofort Bescheid, falls Sie den Übungsschein nicht (mehr) machen möchten. Geben Sie dabei an, ob sie trotzdem noch Einzelabgaben tätigen möchten!

1. Definition eines Baumes (mittel) (schriftlich, 2+3 Punkte)

Schreiben Sie eine Ada-95-Prozedur, die einen Graphen G als Adjazenzliste als Parameter übergeben bekommt und entscheidet, ob G ein Baum mit Wurzel w ist, wobei $w \in V$ der erste Knoten in der Adjazenzliste sei.

- a) Beschreiben Sie umgangssprachlich die Idee Ihrer Prozedur.
- b) Implementieren Sie Ihren Algorithmus.

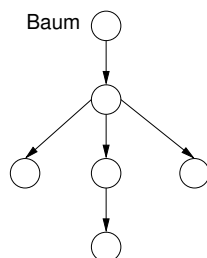
2. Allgemeine Baumnotation I (mittel) (schriftlich, 6 Punkte)

Für die Ausgabe von beliebigen Bäumen (ohne Berücksichtigung von Nullzeigern auf Kinder) sei folgende rekursive Prozedur gegeben (Datenstruktur: Adjazenzliste):

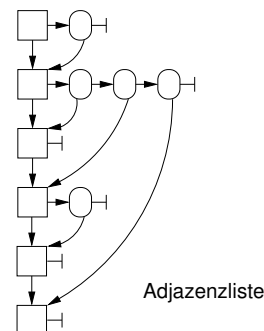
```

procedure Rek(Z : refKnoten) is
  K : refKante:= Null;
begin
  if Z /= Null then
    Put("("); K:= Z.nextKante;
    while K /= Null loop
      Rek(K.nextKnoten);
      K:= (K.nextKante);
    end loop;
    Put(")");
  end if; end Rek;

```



Klammernotation: (((()())()))

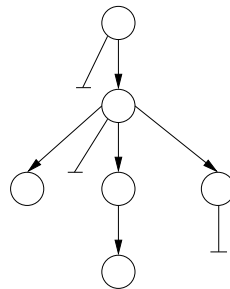


Schreiben Sie ein Ada-Programm, das einen beliebigen Baum als Zeichenkette einliest und falls die Eingabe eine korrekte Klammerfolge ist, den dazugehörigen Baum erstellt (Datenstruktur: Adjazenzliste).

3. Allgemeine Baumnotation II (mittel) (schriftlich, 3 Punkte)

Wie muss die Prozedur geändert werden, wenn auch Nullzeiger auf Kinder eines Knotens berücksichtigt werden sollen? Ist dies auch ohne ein zusätzliches Zeichen in der Klammernotation möglich?

Geben Sie im eClaus-System die veränderte Prozedur und deren Ausgabe für folgenden Baum ab:

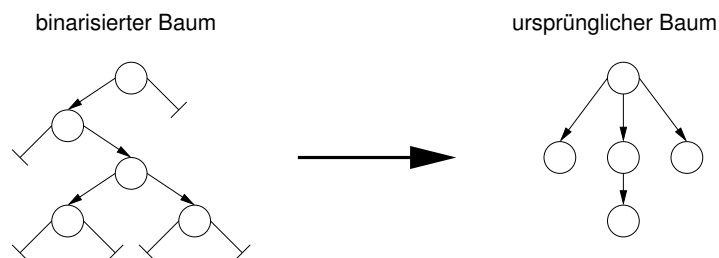


4. Baumumwandlung (leicht) (schriftlich, 3 Punkte)

Schreiben Sie ein Ada-Programm, das einen Binärbaum (Datenstruktur: Binärbaum, siehe Kapitel 8, Seite 40) in den gleichen Binärbaum umwandelt, der aber als Datenstruktur eine Adjazenzliste verwendet.

5. Entbinarisierung (mittel–schwer) (schriftlich, 6 Punkte)

Schreiben Sie Ada-Funktion, das einen Binärbaum (Datenstruktur: Binärbaum, siehe Kapitel 8, Seite 40) als das Ergebnis eines binarisierten Baumes interpretiert und falls möglich den dazu gehörenden Baum als Zeiger auf eine Adjazenzliste zurückgibt.



6. Binärbaum auf Gleichheit prüfen (leicht) (schriftlich, 2 Punkte)

Schreiben Sie ein Ada-Programm, das zwei geordnete Binärbäume G_1 und G_2 in der Klammernotation vom Benutzer einliest und auf Gleichheit prüft. Prüfen Sie dabei, ob die vom Benutzer eingegebene Klammerfolge eine korrekte Klammerung ist.

- Müssen Sie die Binärbäume im Speicher anlegen?
- Welche Zeitkomplexität hat Ihr Algorithmus?