

Übungen zur Vorlesung Einführung in die Informatik II

Ausgabe: 7. Mai 2002

Abgabe: 13. Mai 2002, 12.00 Uhr

Die Abgabe erfolgt ausschließlich über das System eClaus! Siehe
http://www.informatik.uni-stuttgart.de/ifi/fk/lehre/ss02/info_II_02.html

Aufgabe 1: (Einfach verkettete Listen)**schriftlich**

Der Inhalt einfach verketteter Listen lässt sich leicht ausgeben, in dem man beginnend mit dem Anker der Liste, diese sequentiell durchläuft, bis man auf einen `null` Zeiger trifft. Falls das letzte Element nicht `null` als Nachfolger hat, sondern ein beliebiges Element der Liste, so entsteht ein Zyklus. Die Ausgabe des Inhalts mit obigem Algorithmus führt dann in eine Endlosschleife.

- a. (3 Punkte) Schreiben Sie ein Ada 95 Programm, das einfach verkettete Listen erzeugt. Dabei soll die Anzahl der Elemente der Liste zufällig sein, der Inhalt jedes Elements eine zufällige Zahl aus dem Bereich 1 bis 42. Das letzte Element der Liste soll entweder auf `null` zeigen oder auf ein zufällig gewähltes Element der Liste.
- b. (3 Punkte) Schreiben Sie eine Ada 95 Funktion, die zu einer gegebenen einfach verketteten Liste erkennt, ob diese einen Zyklus hat oder nicht.
- c. (2 Punkte) Schreiben Sie eine Ada 95 Funktion, die zu einer gegebenen einfach verketteten Liste, die einen Zyklus enthält, das letzte Element des Zyklus erkennt und den Nachfolger dieses letzten Elements auf `null` umsetzt (und somit den Zyklus auflöst).

Schätzen Sie jeweils den Aufwand ihrer Algorithmen ab. Kommentieren Sie ihre Abgaben bitte so, dass ihr Tutor nachvollziehen kann, welche Ideen Sie beim Entwurf der Algorithmen hatten.

Die Aufwandsabschätzung sollte ebenfalls als Kommentarzeilen im Programmcode erfolgen (schreiben Sie Sonderzeichen wie Summenzeichen einfach aus, also z.B. $\sum_{i=1}^n$ als "Summe(i=1 bis n)").

Hinweis: Das Programm soll für die Liste keinerlei Zusatzinformation wie z.B. die Anzahl der Elemente verwalten. Machen Sie sich klar, dass Sie direkt auf den Zeigern arbeiten müssen, da die Werte der Listenelemente nicht paarweise verschieden sein müssen.

Hinweis 2: Es gibt Lösungen, die mit maximal drei Zeigern auskommen und nur linearen(!) Aufwand haben.

Aufgabe 2: (Intervallschachtelung)**zum Votieren**

In der Vorlesung wurde als Beispiel zur Intervallschachtelung ein Programm zur binären Suche vorgestellt. Dabei wird getestet, ob das gesuchte Element genau in der Mitte des Feldes liegt. War dies nicht erfolgreich, wird analog im linken oder rechten Teilfeld weitergesucht. Sucht man hingegen z.B. im Telefonbuch einen Eintrag, so wird man nicht stets das Element in der Mitte des Feldes untersuchen, sondern den Index in Abhängigkeit vom gesuchten Element wählen. Bei der Suche nach "Claus, V." würde man z.B. bei einem Telefonbuch mit 260 Seiten zunächst ungefähr auf Seite 30 suchen.

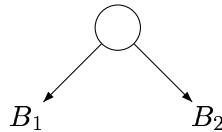
- (3 Punkte) Wandeln Sie das Programm 1 aus dem Beispiel Intervallschachtelung so ab, dass der untersuchte Index abhängig vom gesuchten Element gewählt wird. Wie lange dauert so die Suche nach dem Element 42 in dem Feld A: array (1..127) of integer mit $A(i)=i$, $1 \leq i \leq 127$. Wie lange hätte die binäre Suche gedauert?
- (2 Punkte) Welchen Aufwand hat die modifizierte Intervallschachtelung im schlechtesten Fall? Konstruieren Sie dazu ein Beispiel.

Aufgabe 3: (Catalansche Zahlen)

zum Votieren

Geordnete binäre Bäume (vgl. Suchbäume, Lagally Skript vom 5.2.02) sind folgendermaßen definiert:

- I. Der leere Baum ist ein solcher Baum (in der Programmierung durch null dargestellt).
- II. Wenn B_1, B_2 geordnete binäre Bäume sind, dann auch



Beispiele mit n Knoten:

• $n = 1$

• $n = 2$

• $n = 3$

Es sei C_n die Anzahl aller geordneter binärer Bäume mit n Knoten. Es gilt $C_0 = 1, C_1 = 1, C_2 = 2, C_3 = 5, C_4 = 14, C_5 = 42, C_6 = 132, C_7 = 429, C_8 = 1430, \dots$

- a. (3 Punkte) Finden Sie eine Formel der Form $C_{n+1} = C_n \cdot f(n)$ (Hinweis: auch rationale Formeln sind erlaubt).
- b. (2 Punkte) Finden Sie eine Rekursionsformel für C_n .
- c. (3 Punkte) Zeigen Sie: C_n ist auch die Zahl der korrekten Klammerungen für n Klammerpaare. Z.B.

$$n = 3: ((())), (()), (())(), ()(), ()()()$$

Hinweis: Die Klammerpaare "entsprechen" in dieser Reihenfolge den oben angegebenen fünf geordneten binären Bäumen.

Es werden nur 20 Punkte des Aufgabenblattes gewertet. In eClaus gibt es nun eine Multiple Choice Aufgabe, in der Sie angeben können, ob Sie wenigstens eine Aufgabe votieren wollen. Diese Aufgabe wird dann mit der in der Übung erreichten votierten Punkteanzahl bewertet.

Das Eintragen in die Übungsgruppen ist abgeschlossen. Die Listen (incl. Rauminformation) sind weiterhin unter

<https://inf2.informatik.uni-stuttgart.de/uebungsgruppen-bin/inf2/groups>

einzusehen. Ein- und Umtragen ist dann nur noch bei freier Kapazität direkt über Stefan.Lewandowski@informatik.uni-stuttgart.de möglich.

Die Folien zur Vorlesung liegen zum Kopieren im Semesterapparat und bei der Fachschaft, sowie online unter

http://www.informatik.uni-stuttgart.de/ifi/fk/lehre/ss02/info_II.02.html

Dort werden auch alle weitere Informationen zur Vorlesung und den Übungen bekanntgegeben, insbesondere gibt es dort Informationen zu dem System eClaus, über das die Abgabe der Übungen erfolgt.

Fragen zur Vorlesung und den Übungen, sowie Anregungen und Kritik können auf dem Schwarzen Brett

<http://fachschaft.informatik.uni-stuttgart.de/forum/>

diskutiert werden.