Bergische Universität GH Wuppertal
Fachbereich Mathematik
Prof. Dr. A. Frommer, Holger Arndt

Informatik II: Algorithmen und Datenstrukturen

SS 2002

5. Übungsblatt

Aufgabe 15 (Liste mit Feldern, 6P)

Eine einfach verkettete Liste lässt sich nicht nur mit Zeigern, sondern auch mit Feldern (siehe Vorlesung) implementieren. Dieses kann z. B. durch folgende Definitionen realisiert werden:

```
#define MAX_N 10                /* maximale Laenge der Liste */

typedef int Eintrag_typ;                /* Typ der Eintraege */

typedef struct Liste
{
  int anf;                                /* Anfang der Liste */
  int z;                    /* Nummer des aktuellen Elements */
  int z_vorg;                     /* Nummer des Vorgaengers */
  Eintrag_typ daten[MAX_N];
                           /* Speicherung der Listeneintraege */
  int nachf[MAX_N];          /* Verweis auf Speicherplatz des
                                 nachfolgenden Listenelements */
} Liste;
```

a) Implementieren Sie die in der Vorlesung vorgestellten Operationen `create`, `insert`, `delete`, `reset`, `advance`, `out_of_list` für die oben angegebene Realisierung der Datenstruktur.
Dokumentieren Sie Ihr Vorgehen!
Hinweis: Sie müssen auch über die noch freien Plätze der beiden Arrays Buch führen. Erweitern Sie hierzu den obigen Typ `Liste` geeignet.

b) Schreiben Sie ein Hauptprogramm, das unter Verwendung der oben angegebenen Operationen folgende Schritte nacheinander ausführt:

- Erzeugen einer leeren Liste,
- einfügen der Zahlen $1, 2, \ldots, 10$ in dieser Reihenfolge jeweils am Anfang der Liste,
- löschen des ersten Listenelements,
- löschen des fünften Listenelements,
- einfügen der Zahl 50 an der Stelle 4, d. h. als viertes Listenelement,
- einfügen der Zahl 99 am Ende der Liste (`out_of_list` benutzen!).

Lassen Sie nach jedem Schritt den Wert des Anfangs `anf` und die Elemente der Arrays `daten` und `nachf` ausgeben.

Aufgabe 16 (Polynommultiplikation, 4T+3P)

Gegeben seien zwei Polynome p und q vom Grad $n-1$ mit $n = 2^N$ ($N \in \mathbb{N}$), also $p(x) = a_0 + a_1 x + \ldots + a_{n-1}x^{n-1}$ und $q(x) = b_0 + b_1 x + \ldots + b_{n-1}x^{n-1}$. Es soll $r = pq$ berechnet werden.
Sei $m = 2^{N-1}$, $p_l(x) = a_0 + a_1 x + \ldots + a_{m-1}x^{m-1}$, $p_r(x) = a_m + a_{m+1}x + \ldots + a_{n-1}x^{m-1}$ (q_l, q_r für q analog). Schließlich seien z_l, z_r, z_t definiert durch:

$$\begin{aligned} z_l &= p_l q_l \\ z_r &= p_r q_r \\ z_t &= (p_l + p_r)(q_l + q_r) \end{aligned}$$

a) Zeigen Sie:
$r(x) = z_l(x) + (z_t(x) - z_l(x) - z_r(x))x^m + z_r(x)x^n$

b) Entwerfen Sie einen auf a) beruhenden Divide-and-Conquer-Algorithmus zur Berechnung von r.

c) Implementieren Sie Ihren Algorithmus aus b) in C.
Hinweis: Stellen Sie Polynome als Felder ihrer Koeffizienten dar.

d) Wieviele Multiplikationen benötigt Ihr Algorithmus? Wieviele Multiplikationen sind dagegen beim „üblichen Ausmultiplizieren“ nötig?

(Dieses Prinzip wird auch bi Langzahlarithmetiken verwendet.)

Aufgabe 17 (Das 0/1-Rucksack-Problem, 6T)

Betrachtet wird das bereits aus der Vorlesung bekannte „0/1-Rucksack-Problem“: Gegeben ist ein Rucksack mit Fassungsvermögen f und n Gegenstände mit Volumina $V_1, \ldots, V_n$, wobei die Mitnahme von Gegenstand i einen Gewinn von G_i erbringt. Gesucht sind Koeffizienten $x_1, \ldots, x_n \in \{0, 1\}$, so dass der Gesamtgewinn $G = \sum_{i=1}^{n} x_i G_i$ maximal wird unter der Nebenbedingung $V = \sum_{i=1}^{n} x_i V_i \leq f$.

Es soll nun versucht werden, die Füllung des Rucksacks schrittweise aufzubauen: Entscheide zuerst über die Mitnahme von Gegenstand 1, dann Gegenstand 2 usw.

Im Gegensatz zum Backtracking-Ansatz wird nicht nur eine Füllung im Auge behalten. Stattdessen werden immer alle „optimalen“ Teilfüllungen betrachtet, die sich aus den ersten i Gegenständen bilden lassen.

a) Überlegen Sie, wann eine Teilfüllung aus den ersten i Gegenständen „suboptimal“ ist, d. h. wann sie sich nicht zu einer optimalen Gesamtfüllung erweitern lässt.

b) Konstruieren Sie hiermit einen Algorithmus, der auf dem Prinzip des Dynamischen Programmierens beruht.

c) Zeigen Sie den Ablauf des Verfahrens an folgendem Beispiel: $n = 4$, $f = 40$,

i	1	2	3	4
G_i	11	27	31	43
V_i	1	11	21	17

Abgabe: Mi., 22.05.2002, 14 Uhr