Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik (20)
Johann Wolfgang Goethe-Universität Frankfurt am Main

Praktische Informatik 1

Wintersemester 1997/98

Übungsblatt Nr. 9

Abgabe: Freitag 6. Februar 1998 vor der Vorlesung

Aufgabe 1 (10 Punkte)

Wir nehmen an, daß Mengen als Listen implementiert sind. Berechnen Sie die Liste aller Teilmengen einer Menge (d.h. die Potenzmenge) mit dem "divide and conquer"-Verfahren. Was ist der Zeitbedarf des Algorithmus? Gibt es einen logarithmischen Faktor?

Aufgabe 2 (10 Punkte)

Beweisen Sie, daß der Algorithmus für das Umschichten der Türme von Hanoi $2^n - 1$ Scheibenbewegungen für Türme der Höhe n benötigt. Glauben Sie, daß es einen Algorithmus gibt, der weniger Scheibenbewegungen benötigt? Begründen Sie Ihre Antwort.

Aufgabe 3 (10 Punkte)

Schreiben Sie ein Haskell-Programm, das eine Datei liest, einen Huffman-Kode mit Quell- und Ziel-Alphabet ASCII bzw. $\{0, 1\}$ für diese Datei berechnet und die Datei mit diesem kodiert. Verwenden Sie nach Möglichkeit `arrays` und schreiben Sie den Huffman-Kode mit in die Datei. Schreiben Sie ein entsprechendes Programm, das so kodierte Dateien dekodiert. Zum Lesen und Schreiben der Datei können Sie etwa wie folgt verfahren:

```
main
 = do
    inhalt <- readFile "Datei"
    let huffmanKode = berechneHuffmanKode inhalt -- ein Array
    in
     writeFile "Datei.huff"
      ((show (elems huffmanKode))
      ++(kodiere huffmanKode inhalt))
```

Beim Dekodieren können Sie dann den Huffman-Kode etwa folgendermaßen wieder in einen `array` einlesen: `listArray (0,255) (read inhalt)`. Erst danach steht dann der kodierte Dateiinhalt. (Hinweis: Beim Kodieren könnten Sie die Ziel-Kodes der Zeichen konkatenieren um dann Folgen von je 8 Zeichen über $\{0, 1\}$ in einem `array` nachzusehen und dort das Zeichen zu ermitteln, das in die kodierte Datei geschrieben werden muß.)

Bonus-Aufgabe 4 (15 Punkte)

Gegeben eine Liste mit n Zahlen. Beschreiben Sie einen Algorithmus, der herausfindet, ob es eine Zahl gibt, die öfter als $n/2$ mal in der Liste vorkommt. Gibt es einen linearen Algorithmus? (z.B. wenn man versucht zu vermeiden, daß gleiche Elemente in der Liste nebeneinander liegen) Begründen Sie Ihre Antwort.

Aufgabe 5 (10 Punkte)

Beim "Packen" waren wir davon ausgegangen, daß das was wir einpacken wollen immer ganze "Stücke" sind. Eine Abwandlung ist, sich vorzustellen, daß von dem was verpackt werden soll auch nur ein Anteil eingepackt werden kann, z.B. von Mehl, Milch, Butter, Wein etc. Von den n "Waren" ist jeweils ein bestimmtes Volumen $g_i, 1 \leq i \leq n$ vorhanden und dieses Volumen hat einen bestimmten Preis $v_i, 1 \leq i \leq n$. Wir können dann versuchen den Wert eines Warenkorbs mit Volumen G zu maximieren. Programmieren Sie einen gefräßigen Algorithmus, der dies leistet. Hinweis: gefräßig heißt hier immer zuerst das zu nehmen, das bezogen auf die Maßeinheit am wertvollsten ist.