Entwurf und Analyse von Algorithmen

WS 1999/2000
Prof. Dr. K. Madlener
Übungsblatt 5

Aufgabe 5.1. Betrachten Sie die rekursive Prozedur $matrix(i, j)$ im Buch von Schöning auf Seite 92. Zeigen Sie, dass die Laufzeit im schlechtesten Fall in $\Theta(n^3)$ liegt, wenn man die Multiplikation natürlicher Zahlen als die wesentliche Operation mit Einheitskosten berücksichtigt. (Was ist hier eigentlich n?)

Wie kann man in der Zeile

$$q := matrix(i, k) + matrix(k + 1, j) + p_{i-1} * p_k * p_j$$

die dort verwendeten rekursiven Aufrufe unter Umständen einsparen? Hat Ihre Optimierung einen Einfluss auf die Laufzeit im schlechtesten Fall?

Aufgabe 5.2. Geben Sie einen optimalen binären Suchbaum für die Elemente

$$\langle 1, 2, 3, 4, 5, 6, 7, 8, 9\rangle$$

mit der Wahrscheinlichkeitsverteilung

$$\langle 0.01, 0.15, 0.16, 0.32, 0.21, 0.02, 0.02, 0.07, 0.04\rangle$$

an.

Aufgabe 5.3. Wir wollen uns mit einer in der Vorlesung besprochenen Methode zur Effizienzsteigerung des Dynamischen Programmierens befassen (in der Vorlesung an Hand der optimalen binären Suchbäume demonstriert). Dazu soll folgender Satz gezeigt werden:

Sei $w(i, j) \in \mathbb{R}$ für $0 \le i < j \le n$ und sei $c(i, j)$ definiert durch

$$c(i, i) = w(i, i) = 0 \qquad \text{und}$$
$$c(i, j) = w(i, j) + \min_{i<k\le j}(c(i, k - 1) + c(k, j)) \quad \text{für } i < j.$$

Wenn nun w die Vierecksungleichung

$$w(i, j) + w(i', j') \le w(i', j) + w(i, j') \quad \text{für } i \le i' \le j \le j'$$

erfüllt und monoton bezüglich der Teilmengenrelation auf Intervallen ist (d. h., wenn $w(i, j') \ge w(i', j)$ für $i \le i' < j \le j'$), dann kann die oben definierte Funktion c in Zeit $O(n^2)$ berechnet werden.

Zum Beweis des Satzes gehen Sie wie folgt vor:

- Zeigen Sie zunächst: Wenn w die Vierecksungleichung erfüllt und monoton auf Intervallen ist, dann erfüllt auch c die Vierecksungleichung.

- Sei dann $c_k(i,j)$ eine Abkürzung für $w(i,j) + c(i,k-1) + c(k,j)$ und sei weiter $K(i,j) = \max\{k : c_k(i,j) = c(i,j)\}$ für $i < j$ sowie $K(i,i) = i$. Es ist also $K(i,j)$ der größte Index, mit dem das Minimum in der Definition von $c(i,j)$ erreicht wird. Zeigen Sie nun: Erfüllt c die Vierecksungleichung, dann ist K monoton, d. h. es gilt
$$K(i,j) \leq K(i,j+1) \leq K(i+1,j+1) \quad \text{für } i \leq j.$$

Wie zeigt man nun den ursprünglichen Satz? Was hat der Satz für die Effizienz des Verfahrens zur Bestimmung optimaler binärer Suchbäume zur Folge?

Aufgabe 5.4. Wir betrachten das Problem der Änderungsdistanz zweier endlicher Zeichenketten. Es bezeichne `I` die Einfügeoperation, `D` die Löschoperation, `R` die Ersetzungsoperation und `M` die nichts ändernde (Match-)Operation. Beispiel:

```
RIMDMDMMI
v intner
wri t ers
```

Es wird also `vintner` in `writers` geändert, und zwar durch Ersetzen von `v` durch `w`, Einfügen von `r`, Überlesen von `i` (dies stimmt nämlich bei beiden Zeichenketten überein, d. h. „matcht“), Löschen von `n`, Überlesen von `t`, Löschen von `n`, Überlesen von `e` und `r` sowie Einfügen von `s`.

Wir wollen nun eine endliche Zeichenkette über dem Alphabet $\{\mathtt{I}, \mathtt{D}, \mathtt{R}, \mathtt{M}\}$, die eine Transformation einer Zeichenkette in eine andere beschreibt, ein Änderungsskript zwischen diesen beiden Zeichenketten nennen. Weiter heiße die minimale Anzahl der oben definierten Operationen, die nötig ist, um eine Zeichenkette in eine gegebene andere Zeichenkette zu überführen, die Änderungsdistanz dieser beiden Zeichenketten, wobei allerdings die `M`-Operation nicht gezählt werde.

Das Änderungsdistanz-Problem besteht nun darin, die Änderungsdistanz zwischen zwei gegebenen endlichen Zeichenketten zusammen mit einem Änderungsskript, das die Transformation beschreibt, zu bestimmen.

Es bezeichne, für zwei endliche Zeichenketten S_1 und S_2, $D(i,j)$ die Änderungsdistanz von $S_1[1..i]$ zu $S_2[1..j]$. Die Änderungsdistanz von S_1 zu S_2 ist also $D(n,m)$, wenn $|S_1| = n$ und $|S_2| = m$ ist.

Zeigen Sie, dass folgende Rekursionsgleichung korrekt ist (für $i, j > 0$):
$$D(i,j) = \min(D(i-1,j) + 1, D(i,j-1) + 1, D(i-1,j-1) + t(i,j)),$$
wobei $t(i,j) = 1$, falls $S_1[i] \neq S_2[j]$, und sonst 0 ist. Geben Sie auch sinnvolle Randbedingungen an, so dass diese Rekursionsgleichung gelöst werden kann.

Entwerfen Sie einen Algorithmus, der $D(n,m)$ in einer Zeit in $O(nm)$ berechnet.

Zeigen Sie auch, wie man (nachdem man $D(n,m)$ berechnet hat) ein optimales Änderungsskript in einer Zeit in $O(n+m)$ berechnen kann.

Informationen zur Vorlesung:
http://www-madlener.informatik.uni-kl.de/ag-madlener/lectures/eaa/ws1999-2000/eaa.html