

Institut für Theoretische Informatik
Peter Widmayer
Thomas Tschager
Antonis Thomas

13. April 2016

Datenstrukturen & Algorithmen**Blatt 7****FS 16****Aufgabe 7.1** *Häufigkeiten.*

Achtung: In der ursprünglichen Aufgabenstellung sind die Schlüssel K und L fälschlicherweise vertauscht.

- a) Wir möchten einen langen Text möglichst kurz binär kodieren. Der Text ist aus fünf verschiedenen Buchstaben zusammengesetzt. Für jeden Buchstaben ist gegeben, wie häufig er im Text vorkommt. Erstellen Sie einen optimalen Kodierungsbaum und kodieren Sie damit das Wort **AGGLGSK**.

Schlüssel	A	G	K	L	S
Häufigkeit	9	8	8	7	7

- b) Nun möchten wir einen optimalen Suchbaum erstellen. Wir gehen von einer alphabetischen Ordnung der Schlüssel aus. Neben den Zugriffshäufigkeiten der Schlüssel aus a) sind die folgenden Zugriffshäufigkeiten der Intervalle zwischen zwei Schlüssel gegeben. Beispielsweise wird in 18% der Anfragen nach einem Schlüssel, der echt zwischen A und G liegt, gesucht. Erstellen Sie einen optimalen Suchbaum und geben Sie an, welche gewichtete Pfadlänge der Suchbaum hat.

Schlüsselintervall	(_,A)	(A,G)	(G,K)	(K,L)	(L,S)	(S,_)
Häufigkeit	0	18	7	0	19	17

Aufgabe 7.2 *Palindrome aufzählen.*

Diese Aufgabe behandelt erneut dynamische Programmierung. Bitte berücksichtigen Sie bei der Beschreibung Ihrer Lösung die auf Blatt 6 genannten vier Aspekte.

Ein *Palindrom* ist eine Zeichenkette, die sich von vorne wie von hinten gleich liest, also z.B. das Wort **RENTNER**. Formal ist ein Palindrom eine Zeichenkette $\langle a_1, \dots, a_n \rangle$, wobei entweder $n = 1$ gilt, oder aber es sind $a_1 = a_n$ und $\langle a_2, \dots, a_{n-1} \rangle$ ein Palindrom (für $n = 2$ fordern wir lediglich $a_1 = a_2$). Ein Array $A[1..n]$ speichere eine Zeichenkette der Länge n . Ein Teilarray $A[i..j]$, $1 \leq i \leq j \leq n$, heisst *Palindrom in A*, falls $\langle A[i], \dots, A[j] \rangle$ ein Palindrom ist.

Beispiel: Das Array **[L, A, R, A]** enthält die Palindrome **A, R, L** sowie **ARA** (das Palindrom **A** kommt doppelt vor). Das Array **[A, N, N, A]** enthält die Palindrome **A, N, NN** sowie **ANNA** (die Palindrome **A** und **N** kommen doppelt vor).

- a) Sei A ein Array, das eine Zeichenkette der Länge n speichert. Entwerfen Sie einen Algorithmus nach dem Prinzip der dynamischen Programmierung, der alle Paare (i, j) ausgibt, für die $\langle A[i], \dots, A[j] \rangle$ ein Palindrom ist. Geben Sie auch die Laufzeit Ihrer Lösung an.

Beispiel: Für die Eingabe [L, A, R, A] werden die Paare (1, 1), (2, 2), (3, 3), (4, 4), (2, 4) ausgegeben. Für die Eingabe [A, N, N, A] werden die Paare (1, 1), (2, 2), (3, 3), (4, 4), (2, 3), (1, 4) ausgegeben. Es wird keine spezielle Reihenfolge gefordert, in der die Paare ausgegeben werden müssen.

Hinweis: Beachten Sie, dass die Aufgabe ohne dynamische Programmierung durch triviale Aufzählung aller Palindrome in Zeit $\mathcal{O}(n^3)$ lösbar ist. Gesucht wird also ein effizienterer Algorithmus.

- b) Angenommen, der Algorithmus aus a) hat die DP-Tabelle bereits berechnet. Beschreiben Sie detailliert, wie Sie aus der DP-Tabelle ein längstes Palindrom in A ablesen können. Geben Sie auch die maximal benötigte Laufzeit an.

Aufgabe 7.3 *Längste gemeinsame Teilfolge (Programmieraufgabe).*

In dieser Aufgabe soll ein Verfahren implementiert werden, das eine *längste gemeinsame Teilfolge* mit dynamischer Programmierung berechnet. Für dieses Problem sind zwei Zeichenketten $A = a_1 \cdots a_n$ und $B = b_1 \cdots b_m$ gegeben, wobei $a_1, \dots, a_n, b_1, \dots, b_m$ Zeichen aus einem Alphabet Σ sind. Gesucht wird die Länge einer längsten Zeichenkette, die eine (nicht notwendigerweise zusammenhängende) Teilfolge sowohl von A als auch B ist. Sind zum Beispiel $A = \text{“AGCAT”}$ und $B = \text{“GAC”}$, dann gibt “AC”, “GC” und “GA” längste gemeinsame Teilfolgen. Die Länge einer längsten gemeinsamen Teilfolge ist also 2.

Eingabe Die erste Zeile der Eingabe enthält lediglich die Anzahl t der Testinstanzen. Danach folgen genau zwei Zeilen pro Testinstanz. Die erste Zeile enthält die Sequenz A , und die zweite die Sequenz B . Das verwendete Alphabet ist $\Sigma = \{A, B, \dots, Z\}$.

Ausgabe Für jede Testinstanz soll lediglich eine Zeile ausgegeben werden. Sie enthält die Länge einer längsten gemeinsamen Teilsequenz.

Beispiel

Eingabe:

```
2
AGCAT
GAC
ROCK
ROLL
```

Ausgabe:

```
2
2
```

Hinweis Lösen Sie diese Aufgabe mit dynamischer Programmierung. Wir stellen eine Code-Vorlage zur Verfügung, in der nur mehr die Funktionen `computeTable` und `longestCommonSubseqLen` implementiert werden müssen. In der Funktion `main` wird ausserdem demonstriert, wie Sie einen String mit der Scanner-Klasse einlesen können. Für diese Aufgabe gibt es lediglich ein Testset.

Abgabe: Am Mittwoch, den 20. April 2016 in Ihrer Übungsgruppe.