Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Institut für Theoretische Informatik
Peter Widmayer
Sandro Montanari
Tobias Pröger

17. April 2013

Datenstrukturen & Algorithmen Programmieraufgabe 8 FS 13

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 $a_1, ..., a_n, b_1, ..., b_m$ Zeichen aus einem Alphabet Σ sind. Gesucht wird eine längste Zeichenkette, die eine (nicht notwendigerweise zusammenhängende) Teilfolge sowohl von A als auch B ist. Sind zum Beispiel $A =$ "AGCAT" und $B =$ "GAC", dann gibt "AC", "GC" und "GA" längste gemeinsame Teilfolgen.

Der Algorithmus benutzt eine Tabelle $A[\cdot, \cdot]$ mit $n+1$ Zeilen und $m+1$ Spalten. Für $0 \leq i \leq n$ und $0 \leq j \leq m$ enthält der Eintrag $A[i, j]$ die Länge einer längsten gemeinsamen Teilfolge der Teilzeichenketten $a_1 \cdots a_i$ sowie $b_1 \cdots b_j$. Einträge mit $i = 0$ und $j = 0$ repräsentieren die leere Zeichenkette. Daher werden $A[i, 0]$ und $A[0, j]$ für alle i, $0 \leq i \leq n$ und für alle j, $0 \leq j \leq m$, auf 0 gesetzt. Die verbleibenden Einträge können wie folgt berechnet werden:

$$A[i,j] = \begin{cases} A[i-1, j-1] + 1 & \text{falls } a_i = b_j \\ \max\{A[i-1, j], A[i, j-1]\} & \text{ansonsten.} \end{cases} \tag{1}$$

Nachdem die Tabelle ausgefüllt wurde, enthält der Eintrag $A[n, m]$ genau die Länge k einer längsten gemeinsamen Teilsequenz. Von dort aus kann die längste gemeinsame Teilsequenz selbst durch *Backtracking* ermittelt werden. Ist $a_n = b_m$, dann wird a_n als k-tes Zeichen der längsten gemeinsamen Teilsequenz festgelegt und mit dem Eintrag $A[n-1, m-1]$ fortgefahren. Ist $a_n \neq b_m$, dann kommt a_n in der längsten gemeinsamen Teilsequenz nicht vor. Falls nun $A[n, m] = A[n-1, m]$ gilt, dann fahren wir mit $A[n-1, m]$ fort, und mit $A[n, m-1]$ ansonsten. Dieses Vorgehen wird beendet, wenn alle k Zeichen der längsten gemeinsamen Teilsequenz ermittelt wurden.

Untenstehend findet sich das Beispiel der Tabelle $A[i, j]$ für die Eingaben $A =$ "ROCK" und $B =$ "ROLL".

$A[i,j]$	∅	R	O	L	L
∅	0	0	0	0	0
R	0	1	1	1	1
O	0	1	2	2	2
C	0	1	2	2	2
K	0	1	2	2	2

(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 $\Sigma = \{A, B, ..., Z\}$.

Ausgabe Für jede Testinstanz soll lediglich eine Zeile ausgegeben werden. Sie enthält die Länge einer längsten gemeinsamen Teilsequenz gefolgt von der längsten gemeinsamen Teilsequenz, die vom obigen Algorithmus berechnet wurde.

Beispiel

Eingabe:

```
2
AGCAT
GAC
ROCK
ROLL
```

Ausgabe:

```
2 AC
2 RO
```

Hinweis Sie können das Codefragment

```
String A = scanner.next();
String B = scanner.next();
```

benutzen, um die vollständigen Zeichenketten A und B für jede Testinstanz einzulesen. Ausserdem kann für eine gegebene Zeichenkette `s` die Methode `s.length()` benutzt werden, um die Länge festzustellen, und die Methode `s.charAt(index)`, um das Zeichen an der Stelle `index` zu ermitteln.

Abgabe: Bis Mittwoch, den 24. April 2013.