

Universität Potsdam  
Institut für Informatik  
Algorithmen und Datenstrukturen

11. Aufgabenblatt

können das Dynamische Programmierung Paradigma bei einem gegebenen Algorithmus mithilfe einer Tabelle anwenden sowie visualisieren.

können einen Algorithmus nach dem Dynamische Programmierung Paradigma für ein gegebenes Problem entwerfen.

## 18 Dynamische Programmierung

1. Berechnen Sie die maximale Teilsumme der Sequenz

$$(4, -3, -2, 5, 2, 1, -5, 7)$$

nach dem Paradigma Dynamische Programmierung (`maxTeilsumme_2`). Stellen Sie die Tabelle schrittweise auf.

2. Lösen Sie das RNA-Sekundärstrukturproblem für den folgenden RNA Strang durch die Angabe der zugehörigen Tabelle (siehe Vorlesung Dynamische Programmierung, Folie 29-32):

A C A C G U G U G U G U

*Zur Erinnerung: Nur A und U sowie C und G können Bindungen eingehen.*

3. Entwerfen Sie einen Algorithmus nach dem Paradigma der dynamischen Programmierung, der die Editier-Distanz berechnet.

**Eingabe:** Zwei beliebige Strings  $u$  und  $v$ .

**Ausgabe:** Die minimalen Kosten (Definition s. weiter unten), um  $u$  in  $v$  zu transformieren.

Die folgenden Operationen dürfen dabei genutzt werden.

- delete  
löscht ein Zeichen
- insert  $x$   
fügt ein Zeichen  $x$  hinzu
- replace  $x y$   
ersetzt ein Zeichen  $x$  durch das Zeichen  $y$

Alle Operationen ändern immer nur ein Zeichen.

Eine Transformation besteht aus einer beliebigen Folge dieser Operationen. Formal ist eine Transformation über einem Alphabet  $V$  rekursiv definiert:

- (a) Für  $x, y \in V$  sind delete, insert  $x$ , replace  $x y$  Transformationen.
- (b) Wenn  $s$  und  $t$  Transformationen sind, dann auch  $st$ .

Die Kosten einer Transformation  $t$  wird von der folgenden Kostenfunktion  $K(t)$  berechnet:

$$K(t) = \begin{cases} 1, & \text{falls } t \text{ insert} \\ 1, & \text{falls } t \text{ delete} \\ 2, & \text{falls } t \text{ replace} \\ K(s) + K(r), & \text{falls } t = sr \text{ für zwei Transformationen } s \text{ und } r \end{cases}$$

- (a) Ergänzen Sie die folgenden Gleichungen zu einer Rekursionsvorschrift, die zur Berechnung der Editier-Distanz durch dynamische Programmierung geeignet ist. Hierbei gelte:  
für  $1 \leq i \leq |u|$  sei  $u_i$  der  $i$ -ter Buchstabe in  $u$ ;  
für  $1 \leq j \leq |v|$  sei  $v_j$  der  $j$ -ter Buchstabe in  $v$ .

$$E_{0,0} = 0$$

$$E_{i,0} = i$$

$$E_{0,j} = j$$

$$E_{i,j} = \min \begin{cases} E_{i-1,j-1} + 0 & \text{falls } u_i = v_j \\ & // \text{ für replace} \\ & // \text{ für insert} \\ & // \text{ für delete} \end{cases}$$

Die Werte  $E_{i,j}$  der aufzubauenden Tabelle stehen für die Editier-Distanzen der Präfixe  $u_1 \dots u_i$  und  $v_1 \dots v_j$ . Somit ist  $E_{i,0}$  die Distanz von  $u_1 \dots u_i$  zum leeren String  $\varepsilon$  und  $E_{0,j}$  die von  $\varepsilon$  zu  $v_1 \dots v_j$ . Folglich muss  $E_{0,0}$  als Distanz von  $\varepsilon$  zu sich selbst den Wert 0 haben.

- (b) Entwickeln Sie eine geeignete Tabelle und befüllen Sie diese entsprechend Ihrer Rekursionsgleichung für die folgende Eingabe:  
 $u$ : sitting  
 $v$ : kitten