technische universität dortmund

Fakultät für Informatik
Lehrstühle 2, 11, 13

Jun.-Prof. Dr. Maike Buchin
I. Kamehkhosh, N. Kriege, D. Kurz, M. Ludewig, B. Zey

Sommersemester 2018
22. Mai

DAP2 Praktikum – Blatt 8

Abgabe: 04.–08.06.

Wichtig: Der Quellcode ist natürlich mit sinnvollen Kommentaren zu versehen. Überlegen Sie außerdem, in welchen Bereichen Invarianten gelten müssen, und überprüfen Sie diese ggf. an sinnvollen Stellen mit *Assertions* (siehe Hinweis auf Blatt 2).

Wichtig: Am Donnerstag, 31.05., fallen die Praktika aufgrund eines Feiertags aus. Besuchen Sie bitte in dieser Woche ein alternatives Praktikum am Dienstag, Mittwoch oder Freitag.

Langaufgabe 8.1: Editierdistanz (4 Punkte)

Implementieren Sie einen Algorithmus zur Bestimmung der sogenannten minimalen Editierdistanz zwischen zwei Sequenzen von Buchstaben: Unter der minimalen Editierdistanz zwischen zwei Sequenzen a, b von Buchstaben versteht man die minimal benötigte Anzahl von Einfüge-, Lösch- und Ersetzungs-Operationen einzelner Buchstaben, die benötigt werden, um a in b zu transformieren.

So ist beispielsweise 4 die minimale Editierdistanz zwischen den beiden Sequenzen `baacda` und `abace`, weil sich `baacda` nicht mit drei oder weniger, aber mit Hilfe der folgenden vier Einfüge-, Lösch- und Ersetzungs-Operationen in `abace` transformieren lässt:

Anzuwendende Operation	Sequenz nach Operation
Füge 'a' an Position 1 ein	`abaacda`
Lösche 'a' an Position 4	`abacda`
Ersetze 'd' durch 'e' an Position 5	`abacea`
Lösche 'a' an Position 6	`abace`

Die Editierdistanz für zwei Sequenzen a, b mit $|a| = n, |b| = m$ ist der Wert $D_{n,m}$, welcher mit Hilfe dynamischer Programmierung und der folgenden Optimalitätsgleichung berechnet wird:

- $D_{0,0} = 0$
- $D_{i,0} = i, 1 \leq i \leq n$ und $D_{0,j} = j, 1 \leq j \leq m$
- $$D_{i,j} = \min \begin{cases} D_{i-1,j-1} + 0 & \text{falls } a_i = b_j \\ D_{i-1,j-1} + 1 & \text{(Ersetzung)} \\ D_{i,j-1} + 1 & \text{(Einfügung)} \\ D_{i-1,j} + 1 & \text{(Löschung)} \end{cases}, 1 \leq i \leq n, 1 \leq j \leq m$$

Konkret ist Folgendes zu erfüllen:

- Legen Sie eine Klasse `EditDistance` an, die die Methode `int distance(String a, String b)` enthält
- Implementieren Sie die Methode `int distance(String a, String b)` mit Hilfe der oben beschriebenen Optimalitätsgleichung zur Bestimmung der minimalen Editierdistanz
- Dem Programm sollen zwei Parameter für die Sequenzen übergeben werden und es soll die minimale Editierdistanz für die Transformation der ersten Sequenz (erster Parameter) in die zweite Sequenz (zweiter Parameter) berechnet werden,
  Beispielaufruf: `java EditDistance informatics interpolation`

Langaufgabe 8.2: Ausgabe der Editieroperationen (4 Punkte)

Das Programm aus Aufgabenteil 1 soll nun so erweitert werden, dass nicht nur die benötigte Anzahl der Operationen, sondern auch die einzelnen Operationen selbst sowie die Zwischenergebnisse der Transformation ausgegeben werden.

- Schreiben Sie eine Methode `printEditOperations`, die schrittweise die konkreten Operationen und deren jeweilige Kosten ausgibt, die für eine optimale Transformation einer Ausgangssequenz in eine Zielsequenz benötigt werden.
- Dabei soll für jeden (Zwischen-)Schritt die Sequenz mit ausgegeben werden, die nach Anwendung der Operation dieses Schrittes entsteht.
- Die Ausgabe soll dabei so wie in dem folgenden Beispiel formatiert sein:

```
Loesung fuer "baacda" --> "abace" mit Gesamtkosten 4:
=====================================================
1) Kosten 1: Fuege a an Position 1 ein --> abaacda
2) Kosten 0: b an Position 2 --> abaacda
3) Kosten 0: a an Position 3 --> abaacda
4) Kosten 1: Loesche a an Position 4 --> abacda
5) Kosten 0: c an Position 4 --> abacda
6) Kosten 1: Ersetze d durch e an Position 5 --> abacea
7) Kosten 1: Loesche a an Position 6 --> abace
```

- Erweitern Sie Ihr Programm derart, dass als zweiter Parameter "-o" übergeben werden kann. In diesem Fall soll an Stelle der einfachen Ausgabe in Form der minimalen Editierdistanz die detaillierte Ausgabe mit Hilfe der Methode `printEditOperations` erfolgen.
  Beispielaufruf: `java EditDistance informatics interpolation -o`