

Algorithmen und Datenstrukturen

Suche in Texten

Aufgabe 1: Levenshtein-Distanz

Die Levenshtein-Distanz gibt an, wie viele Editieroperationen zum Überführen des einen Strings in den andern notwendig sind. Bestimmen Sie von Hand die Levenshtein-Distanz folgender String-Paare:

AUSTAUSCH – AUFBAUSCH
 BARBAREN – BARBARA
 COCACOLA – COCAINA

Die Aufgabe muss nicht abgegeben werden.

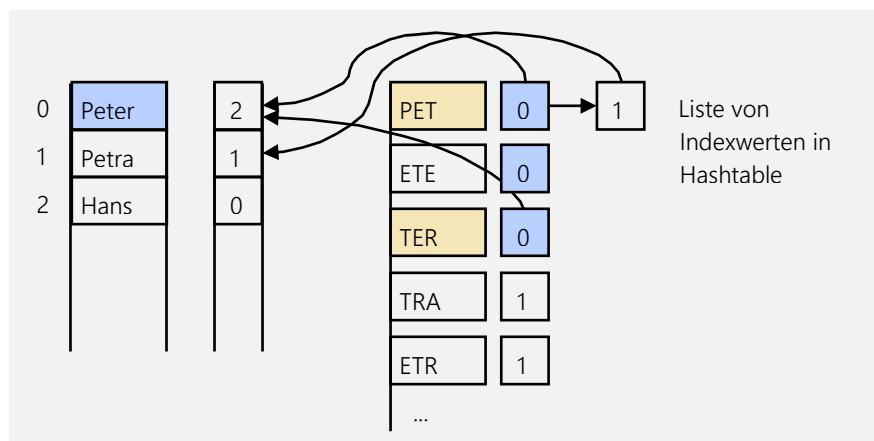
Aufgabe 2: Reguläre Ausdrücke (Regex)

- Definieren Sie reguläre Ausdrücke für eine IP-Adresse, z.B: 12.122.12.1 oder 198.168.1.1.
- Definieren Sie reguläre Ausdrücke für eine E-Mail-Adresse, z.B.: hans.muster@zhaw.ch.

Die Aufgabe muss nicht abgegeben werden.

Aufgabe 3: Fehlertolerante Suche

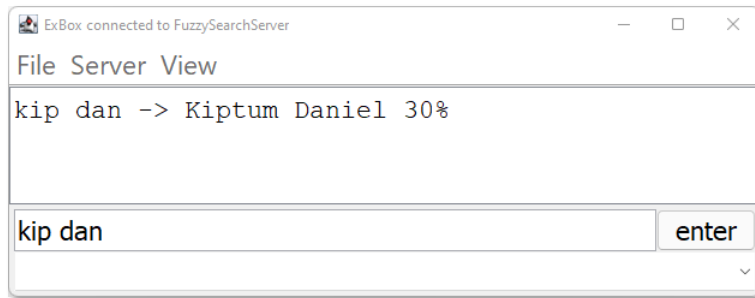
Es soll eine effiziente fehlertolerante Suche nach dem Namen (der Rangliste) implementiert werden. Für diesen Zweck wird die sogenannte Trigramm Methode angewandt. Die Idee dabei ist, dass der String in 3-Buchstaben Gruppen unterteilt wird, z.B. sind das bei „Peter“, die 3-er Gruppen „PET“, „ETE“, „TER“. Diese 3-er Gruppen werden für alle vorkommenden Namen gebildet und wie in folgender Abbildung gezeigt abgespeichert.



Die 3-er Gruppen werden in eine Hashtabelle als Schlüssel abgelegt; der dazugehörige Wert entspricht einer Liste von (Array-/Listen-)Indexwerten der entsprechenden Namen, z.B. PET kommt in Peter und Petra vor, demnach enthält die Liste 0 und 1 (Arrayindex von Peter und Petra).

Bei der Suche wird der Suchstring ebenfalls in 3-er Gruppen unterteilt. Diese Gruppen werden nun in der Hashtabelle gesucht und der Zähler des entsprechenden 2 Namens um eins inkrementiert. Das wird für alle 3-er Gruppen des Suchstrings gemacht. Derjenige Namen mit dem höchsten Zählwert am Schluss ist dann der gesuchte.

Aufgabe: Implementieren Sie einen FuzzySearchServer dem Sie eine Rangliste übergeben und einen Trigram Index erzeugen können. Es soll dann nach einem Namen mit z.B. "Kip Dan" gesucht werden können und es sollte "Kiptum Daniel" gefunden werden.



Hinweis:

- Der Name wird zuerst noch "normalisiert", i.e. alles klein geschrieben.
- Map<String, List> zur Speicherung der Trigramme verwenden.
- Ergänzen Sie das Programmgerüst.

Aufgabe 1: Levenshtein-Distanz

Die Levenshtein-Distanz gibt an, wie viele Editieroperationen zum Überführen des einen Strings in den anderen notwendig sind. Bestimmen Sie von Hand die Levenshtein-Distanz folgender String-Paare:

AUSTAUSCH - AUFBAUSCH

BARBAREN - BARBARA

COCACOLA - COCAINA

Die Aufgabe muss nicht abgegeben werden.

AUSCH		A	U	F	B	A	U	S	C	H
A	0									
U		0								
S				1						
T					2					
A						2				
U							2			
S								2		
C									2	
H										2

BARBAREN
00000012 \Rightarrow 2
BARBARA

COCACOLA
00001233 \Rightarrow 3
COCAINA

Aufgabe 2: Reguläre Ausdrücke (Regex)

- a) Definieren Sie reguläre Ausdrücke für eine IP-Adresse, z.B: 12.122.12.1 oder 198.168.1.1.
- b) Definieren Sie reguläre Ausdrücke für eine E-Mail-Adresse, z.B.: hans.muster@zhaw.ch.

Die Aufgabe muss nicht abgegeben werden.

a)

$(\backslash 2 \in \{1, 3\} \backslash \backslash .) \in \{3\} \backslash 2 \in \{1, 3\} (\backslash \backslash . \backslash)$

b)

$\cdot + @ \backslash w + \backslash \backslash . \backslash w +$