



Informationssysteme

Sommersemester 2016

Prof. Dr.-Ing. Sebastian Michel
TU Kaiserslautern

smichel@cs.uni-kl.de

Was sind Informationssysteme?

Informationssysteme sind computergestützte Programmsysteme, die Informationen ...

- **erfassen**,
- **(dauerhaft) speichern**,
- **verarbeiten**,
- **verändern**,
- **analysieren**,
- **bereit stellen und anzeigen**

Beispiel: Betriebliche Informationssysteme

- spiegeln Geschäftsmodell eines Unternehmens wider
- organisieren und unterstützen Arbeitsabläufe
- integrieren eine Vielzahl von Datenquellen
- **Gewährleistung von Konsistenz (ACID)** essenziell
- im Kern: **traditionelle Datenbanksysteme** (Oracle, DB2, ...)

Beispiel: Betriebliche Informationssysteme

- spiegeln Geschäftsmodell eines Unternehmens wider
- organisieren und unterstützen Arbeitsabläufe
- integrieren eine Vielzahl von Datenquellen
- **Gewährleistung von Konsistenz (ACID)** essenziell
- im Kern: **traditionelle Datenbanksysteme** (Oracle, DB2, ...)

Studenten	
<u>Matr</u>	Name
26120	Fichte
25403	Jonas
...	...

hören	
<u>Matr</u>	<u>VorlNr</u>
25403	5022
26120	5001
...	...

Vorlesungen	
<u>VorlNr</u>	Titel
5001	Grundzüge
5022	Glaube und Wissen
...	...

Beispiel: Betriebliche Informationssysteme (2)

Typisches Beispiel: Bankanwendung

1. Lese den Kontostand von A in die Variable a : $read(A,a)$;
2. Reduziere den Kontostand um 50 Euro: $a := a - 50$;
3. Schreibe den neuen Kontostand in die Datenbasis: $write(A,a)$;
4. Lese den Kontostand von B in die Variable b : $read(B,b)$;
5. Erhöhe den Kontostand um 50 Euro: $b := b + 50$;
6. Schreibe den neuen Kontostand in die Datenbasis: $write(B,b)$;

Was kann hier schiefgehen?

ACID-Paradigma: Atomarität, Konsistenz (Consistency), Isolation, Dauerhaftigkeit

Beispiel: Google

[Alle](#)[Bilder](#)[News](#)[Videos](#)[Shopping](#)[Mehr ▾](#)[Suchoptionen](#)

Ungefähr 70.400.000 Ergebnisse (0,93 Sekunden)

Ergebnisse für ***britney spears***

Stattdessen suchen nach: [britney spears](#)

Britney Spears

www.britneyspears.com/ ▾ [Diese Seite übersetzen](#)

149 Results - The official site. Features a photo gallery, tour information, and news about **Britney**.

Britney Spears – Wikipedia

https://de.wikipedia.org/wiki/Britney_Spears ▾

Britney Jean Spears (* 2. Dezember 1981 in McComb, Mississippi) ist eine US-amerikanische Popsängerin. Sie verkaufte in weniger als zehn Jahren mehr als ...
[Privatleben](#) - [Karriere](#) - [Rezeption](#) - [Diskografie](#)

Britney Spears (@britneyspears) • Instagram photos and ...

<https://www.instagram.com/britneyspears/> ▾ [Diese Seite übersetzen](#)

It's Britney Bitch! Get #PieceOfMe ... **britneyspears**. Verified Follow. **Britney Spears**.
It's Britney Bitch! Get #PieceOfMe tickets now! britney.lk/pomtickets. 958 posts ...

Übersicht

Die Vorlesung Informationssysteme besteht aus zwei Teilen:

Informationssuche und Data-Mining

Datenbanksysteme

Übersicht: Informationssuche und Data-Mining

Informationssuche

- Dokumentverarbeitung und Distanzmaße auf Zeichenketten
- Precision und Recall
- Boolesches Modell, Vektorraummodell, TF*IDF
- Vielfalt und Neuheit, Ähnlichkeit zwischen Dokumenten
- Latent-Semantic-Indexing (LSI)
- PageRank

Data-Mining

- “Warenkorbanalyse”: Frequent-Itemset-Mining und Association-Rules
- Clustering: K-Means-Algorithmus

Übersicht: Datenbanksysteme

- Entity-Relationship-Modellierung
- Das relationale Modell
- Entwurfstheorie
- Relationenkalküle, relationale Algebra
- SQL (Standard Query Language)
- Sichten (Views)
- Fensteranfragen und Rekursion in SQL
- DB-Schnittstellen zur Anwendungsprogrammierung: JDBC
- Benutzerdefinierte Prozeduren (UDF)
- Integritätsbedingungen und Trigger
- Anfrageverarbeitung
- Indexstrukturen (B+ Baum, Hashing)
- Kostenschätzung und regelbasierte Anfrageoptimierung
- Transaktionskonzept (ACID)
- Mehrbenutzersynchronisation und Recovery

Organisatorisches

Vorlesung und Übung

- 4 SWS Vorlesung: Wöchentlich, Dienstags und Donnerstags.
 - Dienstags, Raum 46-220, 15:30 bis 17:00 Uhr
 - Donnerstags, Raum 46-220, 10:00 bis 11:30 Uhr
- 2 SWS Übung. **Übungsleiter: Manuel Hoffmann.**
Übungsgruppenleiter: Stefan Braun, Peter Brucker und Nico Schäfer.

Klausur

Stand der Planung:

- Abschlussklausur am 03.08.2016, ~8:30–11:00 Uhr
- Zweite Klausur am 16.09.2016, ~15:30–19:00 Uhr
- Für die Zulassung zur Klausur bedarf es einer erfolgreichen Teilnahme am Übungsbetrieb.

Regeln zum Übungsbetrieb

Melden Sie sich im OLAT für die Übung an. Zugangscode: **RowStore**

- Es gibt **13 Übungsblätter**, die in der wöchentlich stattfindenden Übung besprochen werden.
- Auf diesen Übungsblättern gibt es insgesamt 40 Aufgaben, die Sie in Vorbereitung auf die Übung lösen müssen.
- **Auf diese 40 Aufgaben gibt es jeweils einen Punkt.**
- **Sie müssen insgesamt 30 Punkte erreichen**, um für die Klausur zugelassen zu werden. Die Punktevergabe funktioniert wie folgt:

Regeln zum Übungsbetrieb (2)

- Für jedes Übungsblatt gibt es eine Frist – in der Regel Montag auf Dienstag 23:59 CEST – zu der die Aufgaben als **PDF in OLAT abgegeben und** die jeweiligen Aufgaben **als bearbeitet markiert werden müssen.**
- **Geben Sie eine Aufgabe ab, erhalten Sie einen Punkt, falls diese vollständig und überwiegend richtig bearbeitet ist.**
- Sollte die Aufgabe überwiegend falsch sein, wird dieser Punkt aberkannt.
- Sollten Sie versuchen, mit der Abgabe einer leeren oder gänzlich nicht zureichenden Abgabe einen Punkt zu erlangen, werden Ihnen zusätzlich **drei Minuspunkte angerechnet.**

Regeln zum Übungsbetrieb (3)

- Die **Abgabe dient als Grundlage für die Übung**, so können etwa häufig auftretende Probleme ausführlich besprochen werden.
- Weiterhin müssen Sie in der Übung die Liste der bearbeiteten Aufgaben gegenzeichnen und **Ihre erarbeitete Lösung auf Aufforderung vorstellen** können.
- **Sollten Sie dazu nicht in der Lage sein, wird auch hier der Punkt auf die Aufgabe nicht gegeben und Ihnen ggf. drei Minuspunkte angerechnet.**
- Sie dürfen die Abgaben durchaus in Gruppenarbeit anfertigen, es muss trotzdem jeder Teilnehmer ein PDF hochladen und das Ergebnis in der Übung vorstellen können.

▼ Abgabedatum

Diese Checkliste ist mit einem Abgabedatum versehen. Nach dem Abgabedatum kann die Checkliste nicht mehr von Ihnen verändert werden.

 **Abgabedatum: 03.05.2016 00:00**

Markierung

Information

- | | |
|-----------------------------------|-----------------------------|
| <input type="checkbox"/> Erledigt | Aufgabe 1 (1 Punkte) |
| <input type="checkbox"/> Erledigt | Aufgabe 2 (1 Punkte) |
| <input type="checkbox"/> Erledigt | Aufgabe 3 (1 Punkte) |

  Abgabe

Abgabe Termin: 02.05.2016

Geben Sie Ihre Lösung ab indem Sie entweder ein Dokument hochladen oder den hier verfügbaren Editor benutzen.

Achtung: Erst durch die "Endgültige Abgabe" wird dem/der Korrektor/in ihre Lösung zugänglich gemacht.

Vorlesungsfolien und Übungsblätter

Vorlesungsfolien

- Ein in der Regel recht vollständiger Draft wird einen Tag vor der Vorlesung online gestellt.
- Stabile Version folgt dann nach der VL (bzgl. Typos, Anmerkungen, ...).

Übungsblätter

- Ausgabe Dienstags in der Vorlesung.
- Abgabe via OLAT bis Montag der darauffolgenden Woche (Abgabetermin ist auf Übungsblatt angegeben)
- Besprechung in den folgenden Übungen.

Literatur: Informationssuche

Christopher D. Manning, Prabhakar Raghavan und Hinrich Schütze.
Introduction to Information Retrieval.

Online frei verfügbar unter <http://www-nlp.stanford.edu/IR-book/>

Literatur: Datenbanksysteme

- **Alfons Kemper und Andreas Eickler: Datenbanksysteme – Eine Einführung.**
- Theo Härder und Erhard Rahm: Datenbanksysteme – Konzepte und Techniken der Implementierung.
- Elmasri, R., Navathe, S. B.: Grundlagen von Datenbanksystemen, Ausgabe Grundstudium,
- Raghu Ramakrishnan und Johannes Gehrke: Database Management Systems.

Grundlagen der Informationssuche

Grundlagen der Informationssuche

- Dokumentverarbeitung, Distanzmaße auf Zeichenketten
- Precision und Recall
- Boolesches Modell, Verarbeitung von Dokumenten
- Vektorraummodell, TF*IDF
- Vielfalt und Neuheit, Ähnlichkeit zwischen Dokumenten
- Latent-Semantic-Indexing (LSI)
- Pagerank
- Frequent-Itemset-Mining
- K-Means-Clustering

Rothkäppchen schlug die Augen auf, und als es sah wie die Sonnenstrahlen durch die Bäume hin und her tanzten, und alles voll schöner Blumen stand, dachte es 'wenn ich der Großmutter einen frischen Strauß mitbringe, der wird ihr auch Freude machen; es ist so früh am Tag, daß ich doch zu rechter Zeit ankomme,' lief vom Wege ab in den Wald hinein und suchte Blumen. Und wenn es eine gebrochen hatte, meinte es weiter hinaus stände eine schönere, und lief darnach, und gerieth immer tiefer in den Wald hinein. Der Wolf aber gieng geradeswegs nach dem Haus der Großmutter, und klopfte an die Thüre. Wer ist draußen?' 'Rothkäppchen, das bringt Kuchen und Wein, mach auf.' 'Drück nur auf die Klinke,' rief die Großmutter, 'ich bin zu schwach und kann nicht aufstehen.' Der Wolf drückte auf die Klinke, die Thüre sprang auf und er ging, ohne ein Wort zu sprechen, gerade zum Bett der Großmutter und verschluckte sie. Dann that er ihre Kleider an, setzte ihre Haube auf, legte sich in ihr Bett und zog die Vorhänge vor. Rothkäppchen aber war nach den Blumen herum gelaufen, und als es so viel zusammen hatte, daß es keine mehr tragen konnte, fiel ihm die Großmutter wieder ein und es machte sich auf den Weg zu ihr. Es wunderte sich daß die Thüre aufstand, und wie es in die Stube trat, so kam es ihm so seltsam darin vor, daß es dachte 'ei, du mein

Fragen

- Was ist ein Dokument?
- Was sind gültige "Treffer"?
- Was ist mit Dingen wie Tippfehlern?
- Ist jedes Wort gleich wichtig?
- Ist es relevant, ob ein Wort mehrfach auftritt?
- Sind sehr lange Dokumente ein Problem?
- Was ist mit dem Dokument selbst, sind z.B. manche Webseiten glaubwürdiger oder besser als andere?

Was ist ein Dokument?

Falls ein Dokument nicht in einem einfachen Textformat (z.B. ASCII, UTF-8) vorliegt muss es konvertiert werden (z.B. aus PDF, Word, HTML)

Was ist die gewünschte Granularität der Daten/Dokumente?

- Soll das Buch “Grimms Kinder- und Hausmärchen” als ein Dokument betrachtet werden, oder sollte jedes Märchen als ein separates Dokument angesehen werden?
- Sollen HTML Seiten, die jeweils nur einen Paragraph eines Dokuments darstellen aneinanderhängt werden?

Tokenisierung (Tokenization)

- Beim Tokenisieren wird Text in einzelne Tokens aufgeteilt.
- Wir betrachten ganz allgemein Terme, normalerweise einfache Worte, die möglicherweise noch normalisiert sind.
- Beispiel:

Aber	Großmutter	was	hast	du	für	ein
entsetzlich	großes	Maul	!			

Mögliche Probleme

- can't ⇒ can t
- was ist mit Web- oder Emailadressen wie <http://www.cs.uni-kl.de> oder support@ebay.com?
- Los Angeles
- Lebensversicherungsgesellschaftsangestellter
- Oder Sprachen, die gar keine Leerzeichen haben (viele Ostasiatische Sprachen)

Stopwörter

- **Stopwörter sind sehr häufig auftretende Worte, die somit kaum oder keine Informationen enthalten.**
- Daher werden diese oftmals komplett ignoriert.
- **Beispiele:** ein, eine, der, die das, von, aus
- Gegeben entweder durch explizite Liste oder implizit durch z.B. Betrachtung der häufigsten k Worte der Kollektion)

Stammformreduktion (Stemming)

- **Verschiedene Variationen eines Wortes können zusammen betrachtet werden, z.B. Plural, Adverbien, verschiedene Zeiten eines Verbs.**
- **Beispiele***: kaufen → kauf, käufer → kauf, kategorie → kategori, consistency → consist, knives → knife, consolidating → consolid.
- Eine solche **“Grundform”** eines Wortes ist nicht notwendigerweise ein richtiges (korrektes) Wort.
- Idealerweise werden Variationen des gleichen Wortes auf eine einzige Grundform reduziert.

* <http://snowball.tartarus.org/algorithms/english/stemmer.html>

Andere Ideen

- Behandlung von **diakritischen Zeichen**, z.B. ü, á, ç, ø
 - Anfragen enthalten oftmals diese Zeichen gar nicht, z.B. les miserables vs. les misérables
 - Manchmal werden solche Buchstaben auch umgeschrieben, wie für ⇒ fuer
- **Groß-/ Kleinschreibung**, z.B. United States vs. united states

Tippfehler etc – Distanzmaße auf Zeichenketten

- Oft enthalten Texte aber auch die von Benutzern gestellten Anfragen **Tippfehler**.
- Wenn ein Benutzer ein dem System unbekanntes Wort eingibt, so kann anstelle dessen mit einem korrekt geschriebenen Wort, welches möglichst nah am inkorrekten Wort liegt, weitergearbeitet werden.
- Oder man kann Vorschläge zu ähnlichen Suchbegriffen erhalten, im Sinne von “Meinten Sie”
- Was für dazu brauchen ist ein **Distanzmaß auf Zeichenketten**
- Diese sollten z.B. berücksichtigen:
 - **Extra Zeichen**, wie in Haus vs. Hauus
 - **Vergessene Zeichen**, wie in Haus vs. Hus
 - **Fehlerhafte Zeichen**, wie in Haus vs. Hius

Hamming-Editier-Distanz

- **Die Hamming-Editier-Distanz** beschreibt die Anzahl an Positionen, an denen die beide Strings x und y verschieden sind.
- **Strings unterschiedlicher Länge** werden verglichen in dem der kürzere String mit "NULL" aufgefüllt wird (z.B. uni vs. universell \rightarrow uni_____ vs. universell)

Beispiele:

- $d(\text{car}, \text{cat})=1$
- $d(\text{house}, \text{hot})=3$
- $d(\text{house}, \text{hooose})=4$

Dreiecksungleichung

- Für Strings x, y, z und Distanzmaß d gilt

$$d(x, z) \leq d(x, y) + d(y, z)$$

Längste Gemeinsame Teilsequenz

Eine **Teilsequenz** ist eine Sequenz, die von einer anderen Sequenz durch Weglassen von Zeichen aber unter Beibehaltung der Reihenfolge der Zeichen entsteht.

Eine **gemeinsame** Teilsequenz zweier Strings x und y ist ein String s , so dass **alle Zeichen von s in x und y in der gleichen Reihenfolge auftreten** (aber eben **nicht notwendigerweise zusammenhängend**).

Längste-Gemeinsame-Teilsequenz-Distanz

$$d(x,y) = \max(|x|, |y|) - \max_{s \in S(x,y)} |s|$$

wobei $S(x,y)$ die Menge aller gemeinsamer Teilsequenzen von x und y ist.

Beispiele:

- $d(\text{house}, \text{huse}) = 1$

Längste Gemeinsame Teilsequenz: Beispiel

house vs. huse

Alle Teilsequenzen von house: {h, o, u, s, e, ho, hu, hs, he, ou, os, oe, us, ue, se, hou, hos, hoe, hus, hue, hse, ous, oue, ose, use, hous, houe, hose, huse, ouse, house}

Alle Teilsequenzen von huse: {h, u, s, e, hu, hs, he, us, ue, se, hus, hue, hse, use, huse}

Gemeinsame Teilsequenzen: {h, u, s, e, hu, hs, he, us, ue, se, hus, hue, hse, use, huse}

Die längste dieser Teilsequenzen ist huse mit Länge 4.

Also ist $d(\text{house}, \text{huse}) = 5 - 4 = 1$

Levenshtein-Editier-Distanz

- Die **Levenshtein-Editier-Distanz** zwischen zwei Strings x und y ist die minimale Anzahl von **Änderungsoperationen** (**insert**, **replace**, **delete**), die benötigt werden, um x in y zu transformieren.
- Die **minimale Anzahl an Operationen** $m[i, j]$, um den Präfix-Zeichenkette $x[1 : i]$ in $y[1 : j]$ zu transformieren ist definiert durch

$$m[i, j] = \min \begin{cases} m[i-1, j-1] + (x[i] = y[j]?0 : 1) & \text{(ersetze } x[i]) \\ m[i-1, j] + 1 & \text{(lösche } x[i]) \\ m[i, j-1] + 1 & \text{(füge ein } y[i]) \end{cases}$$

Beispiele:

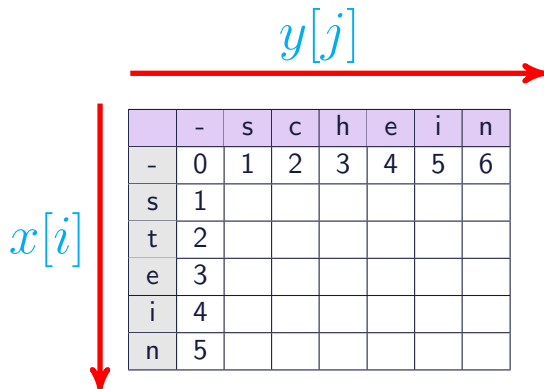
- $d(\text{hooose}, \text{house}) = 1$
- $d(\text{house}, \text{rose}) = 2$
- $d(\text{house}, \text{hot}) = 3$

Levenshtein-Editier-Distanz

Die **Levenshtein-Editier-Distanz** zwischen zwei Zeichenketten x und y entspricht $m[|x|, |y|]$ und kann mittels **dynamischer Programmierung** in $O(|x| |y|)$ berechnet werden.

Anmerkung: Die Kosten für Löschen, Einfügen und Ersetzen können auch unterschiedlich sein.

Initialisierung



	-	s	c	h	e	i	n
-	0	1	2	3	4	5	6
s	1	?					
t	2						
e	3						
i	4						
n	5						

Was ist die Distanz von $x[1:1]='s'$ zu $y[1:1]='s'$?

Drei Möglichkeiten:

- ↖ Distanz von " zu " plus Kosten 0 (weil $y[1]=x[1] = 0+0=0$)
- ↑ Distanz von " zu 's' plus Kosten 1 = $m[0,1]+1 = 2$
- ← Distanz von 's' zu " plus Kosten 1 = $m[1,0]+1 = 2$

$$m[i,j] = \min \begin{cases} m[i-1,j-1] + (x[i] = y[j]?0 : 1) & \text{(ersetze } x[i]) \quad \swarrow \\ m[i-1,j] + 1 & \text{(lösche } x[i]) \quad \uparrow \\ m[i,j-1] + 1 & \text{(füge ein } y[i]) \quad \leftarrow \end{cases}$$

	-	s	c	h	e	i	n
-	0	1	2	3	4	5	6
s	1	0	?				
t	2						
e	3						
i	4						
n	5						

Was ist die Distanz von $x[1:1]='s'$ zu $y[1:2]='sc'$?

Drei Möglichkeiten:

- ↖ Distanz von " und 's' (also $m[0,1]=1$) plus Kosten 1 für Ersetzung von 'c' durch 's'
- ↑ Distanz von " und 'sc' (also 2) plus Kosten 1 für Löschen von 's'.
- ← Distanz von 's' und 's' (also 0) plus Kosten für Einfügen von 'c'.

$$m[i,j] = \min \begin{cases} m[i-1,j-1] + (x[i] = y[j]?0 : 1) & \text{(ersetze } x[i]) \quad \swarrow \\ m[i-1,j] + 1 & \text{(lösche } x[i]) \quad \uparrow \\ m[i,j-1] + 1 & \text{(füge ein } y[i]) \quad \leftarrow \end{cases}$$

	-	s	c	h	e	i	n
-	0	1	2	3	4	5	6
s	1	0	1				
t	2						
e	3						
i	4						
n	5						

	-	s	c	h	e	i	n
-	0	1	2	3	4	5	6
s	1	0	1	2	3	4	5
t	2	1	1	2	3	4	5
e	3	2	2	2	2	3	4
i	4	3	3	3	3	2	3
n	5	4	4	4	4	3	2

Die Levenshtein-Distanz zwischen “stein” und “schein” ist also 2.

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0			
a	2				
t	3				
z	4				
e	5				

Gibt nichts zu tun

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1		
a	2				
t	3				
z	4				
e	5				

Einfügen von a

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	
a	2				
t	3				
z	4				
e	5				

Einfügen von t

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2				
t	3				
z	4				
e	5				

Einfügen von e

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1			
t	3				
z	4				
e	5				

Löschen

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0		
t	3				
z	4				
e	5				

Gibt nichts zu tun

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	
t	3				
z	4				
e	5				

Einfügen von t

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3				
z	4				
e	5				

Einfügen von e

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2			
z	4				
e	5				

Löschen

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1		
z	4				
e	5				

Löschen

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	
z	4				
e	5				

Gibt nichts zu tun

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	1
z	4				
e	5				

Einfügen von e

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	1
z	4	3			
e	5				

Löschen

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	1
z	4	3	2		
e	5				

Löschen

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	1
z	4	3	2	1	
e	5				

Löschen

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	1
z	4	3	2	1	1
e	5				

Ersetzen z durch e

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	1
z	4	3	2	1	1
e	5	4			

Löschen

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	1
z	4	3	2	1	1
e	5	4	3		

Löschen

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	1
z	4	3	2	1	1
e	5	4	3	2	

Löschen

Weiteres Beispiel

	-	k	a	t	e
-	0	1	2	3	4
k	1	0	1	2	3
a	2	1	0	1	2
t	3	2	1	0	1
z	4	3	2	1	1
e	5	4	3	2	1

Gibt nichts zu tun

C++ Code

```
1  int s1_size = s1.size();
2  int s2_size = s2.size();
3  int m[s1_size+1][s2_size+1];
4  //init
5  for (int i=0; i<=s1_size; i++) m[i][0] = i;
6
7  for (int j=0; j<=s2_size; j++) m[0][j] = j;
8
9  for (int i=1; i<=s1_size; i++) {
10     for (int j=1; j<=s2_size; j++) {
11         int min1 = m[i-1][j-1] + ((s1[i-1]==s2[j-1])?0:1);
12         //replace
13         int min2 = m[i-1][j] + 1; //delete
14         int min3 = m[i][j-1] + 1; //insert
15         m[i][j] = min(min1, min(min2, min3));
16     }
17 }
```