Informatik II: Algorithmen und Datenstrukturen SS 2013

Vorlesung 11a, Dienstag, 2. Juli 2013

(Editierdistanz, rekursive Berechnung)

Prof. Dr. Hannah Bast
Lehrstuhl für Algorithmen und Datenstrukturen
Institut für Informatik
Universität Freiburg

UNI FREIBURG

Blick über die Vorlesung heute

UNI FREIBURG

- Organisatorisches
  - Ihre Erfahrungen mit dem Ü10 (BFS, Dijkstra)
- Editierdistanz
  - Maß für die Ähnlichkeit zwischen zwei Wörtern
  - Algorithmus 1: rekursiv
  - Algorithmus 2: mittels "dynamischer Programmierung"
  - Ü11, Aufgabe 1: Algorithmus 2 implementieren

Erfahrungen mit dem Ü10 (BFS, Dijkstra)

UNI FREIBURG

- Zusammenfassung / Auszüge Stand 2. Juli 16:00
  - Test mit den "Zwischenrunden" wegen mehrfach eingefügter Knoten (ohne changeKey) war verwirrend ... einverstanden
  - Dijkstra war etwas dürftig erklärt ... hmm ?
  - Leider keine gescheite Definitionen + zu wenig Zeit
  - Könnte auch gerne noch mal etwas schwerer werden
  - Keine Motivation mehr, da 50% der Punkte schon erreicht
  - Danke für den Hinweis mit dem Gottesbeweis
  - Frühlingspunkt: steht unter Schock / nicht an der TF
  - Gource-Video für unser SVN ?

Musterlösung Ü10, Aufgabe 2

genauer in Kantenanzahl.

- BFS berechnet kürzeste Wege mit Kantenkosten 1
  - Genauer: Level i = { Knoten v mit dist(s, v) = i }
  - Beweis durch vollständige Induktion über i
  - Induktionsanfang i = 0: Level 0 = { s }, und das ist auch der einzige Knoten v mit dist(s, v) = 0
  - Induktionsschritt i → i + 1: z.z. Level $i+1 = \{ v : dist(s,v) = i+1 \}$

    IV: $\forall j \leq i$: Level $j = \{ v : dist(s,v) = j \}$

Gleichheit von Mengen also ⊆ und ⊇

„⊆": v in Level i+1 vom BFS ⇒ v über eine Kante mit Knoten w aus Level i verbunden.

IV ⇒ dist(s,w) = i

⇒ dist(s,v) ≤ i+1 ... = folgt noch nicht!

Annahme dist(s,v) = k ≤ i

IV ⇒ v ∈ Level k ≤ i ↯ v in Level i+1

⇒ dist(s,v) = i+1 ∎

„⊇": ...

Motivation

UNI FREIBURG

- Viele Anwendungen, wo man ähnliche Strings sucht
  - Dubletten in Adressdatenbanken

    Hein Blöd, 27568 Bremerhaven

    Hein Bloed, 27568 Bremerhafen

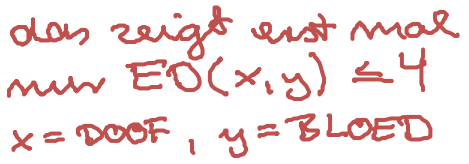
    Hein Doof, 27478 Cuxhaven
  - Produktsuche

    Memory Stik
  - Websuche

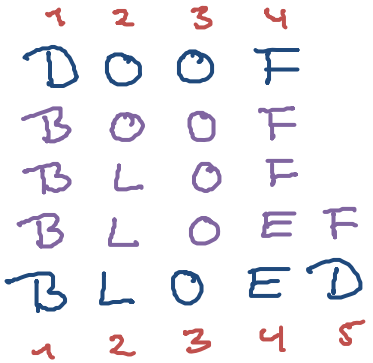
    eyjaföllajaküll

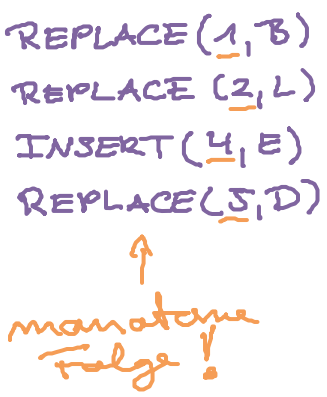
    uniwersität verien 2013
  - Gemeinsamkeit: man braucht ein Maß für die **Ehnlichkeit** zwischen zwei Strings

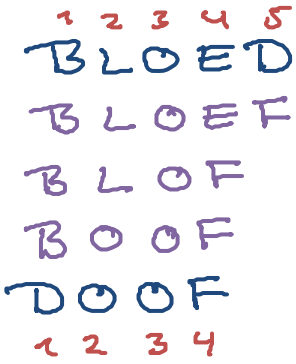
Editierdistanz 1/7

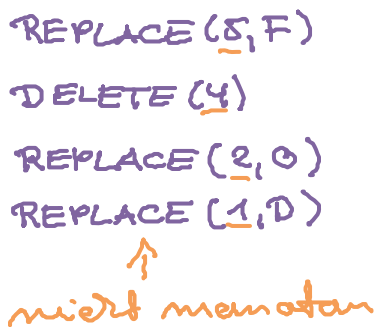


- Definition Editierdistanz, auch Levenshtein-Distanz
  - Gegeben zwei Zeichenketten (strings) x und y
  - ED(x, y) = Editierdistanz (edit distance) von x und y = die minimale Anzahl Operationen um x in y zu transformieren:
    - Einfügen eines Buchstabens (insert)
    - Ersetzen eines Buchstabens durch einen anderen (replace)
    - Löschen eines Buchstabens (delete)
    - Die Position einer Operation ist ... siehe Beispiel:









Editierdistanz 2/7

- Etwas Notation
  - Mit ε bezeichnen wir das leere Wort
  - Mit $|x|$ bezeichnen wir die Länge von x (= Anzahl Zeichen)
  - Mit $x[i..j]$ bezeichnen wir die Teilfolge der Zeichen i bis j der Zeichenkette x, wobei $1 \le i \le j \le |x|$
- Ein paar einfache Eigenschaften
  - $ED(x, y) = ED(y, x)$
  - $ED(x, \varepsilon) = |x|$
  - $ED(x, y) \ge abs(|x| - |y|)$
  - $ED(x, y) \le ED(x[1..n-1], y[1..m-1]) + 1$

man kann die Folge von Operationen "umdrehen" ... siehe Folie 6

aber nicht umbedingt =

|x|=3 |y|=7 4

$abs(x) = x > 0 \; ? \; x : -x$

$n = |x|, m = |y|$

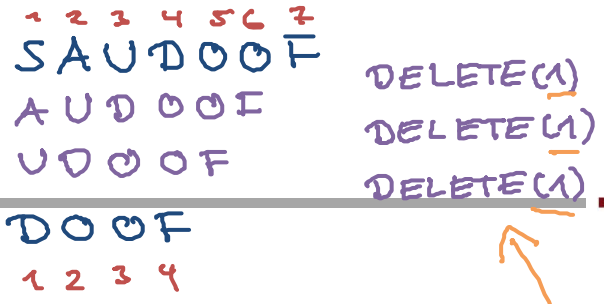
Editierdistanz 3/7

- Lösungsideen anhand von Beispielen
  - Für VERIEN → FERIEN ? ED = 1
  - Für MEXIKO → AMERIKA ? ED = 3
  - Für AAEBEAABEAREEAEBA → RBEAAEEBAAAEBBAEAE ?
  - Beobachtung: möglichst große gemeinsame Teilstrings zu finden klappt manchmal, aber nicht immer
- Rekursiver Ansatz
  - In zwei Teile / Hälften teilen? Keine gute Idee, z.B.

    ED(GRAU, RAUM) = 2 aber ED(GR, RA) + ED(AU, UM) = 4 (=2, =2)
  - Auf ein "kleineres" Problem zurückführen?

    Das probieren wir jetzt !

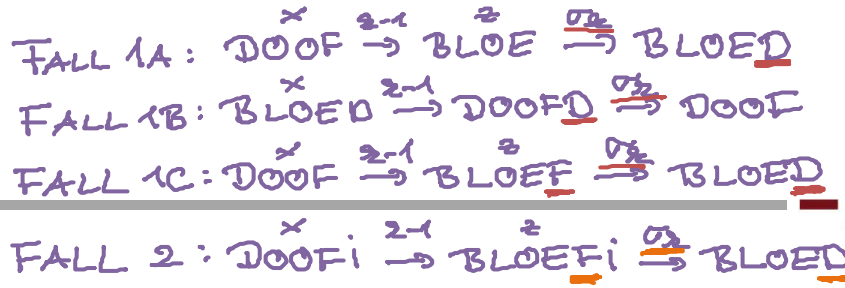
Editierdistanz 4/7



- Terminologie
  - Seien x und y unsere beiden Zeichenketten
  - Seien $\sigma_1, \ldots, \sigma_k$ eine Folge von $k = ED(x, y)$ Operationen für $x \rightarrow y$, das heißt um x in y zu überführen

    (Wir nehmen im Folgenden nicht an, dass wir die Folge schon kennen, sondern nur, dass es so eine gibt)
  - Wir betrachten im Folgenden nur **monotone** Op.-Folgen, d.h. die Position von σ_{i+1} ist $\geq$ die Position von σ_i, wobei $=$ nur dann erlaubt ist, wenn beides delete Operationen sind
  - **Lemma:** Für beliebige x und y mit $k = ED(x, y)$ gibt es eine **monotone** Folge von k Operationen für $x \rightarrow y$
  - Beweisintuition: die Reihenfolge der Operationen ist im Grunde egal, also kann man sie auch monoton anordnen

Editierdistanz 5/7



- Fallunterscheidung
  - Wir betrachten die letzte Operation σ_k
    - $\sigma_1, \ldots, \sigma_{k-1} : x \to z$ und $\sigma_k : z \to y$
    - Seien $n = |x|$ und $m = |y|$ und $m' = |z|$
    - Man beachte, dass $m' \in \{m - 1, m, m + 1\}$
  - Fall 1: σ_k macht etwas "ganz am Ende" von z, d.h. eins von:
    - Fall 1a: $\sigma_k = \text{insert}(m' + 1, y[m])$ [dann ist $m' = m - 1$]
    - Fall 1b: $\sigma_k = \text{delete}(m')$ [dann ist $m' = m + 1$]
    - Fall 1c: $\sigma_k = \text{replace}(m', y[m])$ [dann ist $m' = m$]
  - Fall 2: σ_k macht nichts "ganz am Ende" von z
    - dann $z[m'] = y[m]$ und $x[n] = z[m']$ und damit

      $\sigma_1, \ldots, \sigma_k : x[1..n-1] \to y[1..m-1]$ und $x[n] = y[m]$

Editierdistanz 6/7

UNI FREIBURG

- Wir haben also einen dieser vier Fälle
  - Fall 1a (insert am Ende): $\sigma_1, \ldots, \sigma_{k-1} : x[1..n] \rightarrow y[1..m-1]$
  - Fall 1b (delete am Ende): $\sigma_1, \ldots, \sigma_{k-1} : x[1..n-1] \rightarrow y[1..m]$
  - Fall 1c (replace am Ende): $\sigma_1, \ldots, \sigma_{k-1} : x[1..n-1] \rightarrow y[1..m-1]$
  - Fall 2 (nichts am Ende): $\sigma_1, \ldots, \sigma_k : x[1..n-1] \rightarrow y[1..m-1]$

Editierdistanz 7/7

- Daraus folgt die folgende rekursive Formel
  - Für $|x| > 0$ und $|y| > 0$ ist:

    $ED(x, y)$ = das Minimum von

    - $ED(x[1..n], y[1..m-1]) + 1$, und (insert) Fall 1A
    - $ED(x[1..n-1], y[1..m]) + 1$, und (delete) Fall 1B
    - $ED(x[1..n-1], y[1..m-1] + 1$ falls $x[n] \neq y[m]$ (replace) Fall 1C
    - $ED(x[1..n-1], y[1..m-1]$ falls $x[n] = y[m]$ Fall 2
  - Für $|x| = 0$ ist $ED(\varepsilon, y) = |y|$
  - Für $|y| = 0$ ist $ED(x, \varepsilon) = |x|$

Literatur / Links

- Editierdistanz
  - In Wikipedia (Definition + Algorithmen)

    http://en.wikipedia.org/wiki/Levenshtein_distance

    http://de.wikipedia.org/wiki/Levenshtein-Distanz