Algorithmen und Datenstrukturen

(Th. Ottmann und P. Widmayer)

Folien: Editierdistanz

Autor: Sven Schuierer

Institut für Informatik
Georges-Köhler-Allee
Albert-Ludwigs-Universität Freiburg

1 Editier-Distanz

Gegeben: zwei Zeichenketten $A = a_1 a_2 \cdots a_m$ und
$B = b_1 b_2 \cdots b_n$

Gesucht: Minimale Kosten $D(A, B)$ für eine Folge von Editieroperationen, um A in B zu überführen

Editieroperationen:

1. Ersetzen eines Zeichens von A durch ein Zeichen von B
2. Löschen eines Zeichens von A
3. Einfügen eines Zeichens von B

Einheitskostenmodell:

$$c(a, b) = \begin{cases} 1 & \text{falls } a \neq b \\ 0 & \text{falls } a = b \end{cases}$$

$a = \epsilon$, $b = \epsilon$ möglich

Dreiecksungleichung soll für c im allgemeinen gelten:

$$c(a, c) \leq c(a, b) + c(b, c)$$

$\Rightarrow$ Ein Buchstabe wird höchestens einmal verändert

2 Spur

Spur als Repräsentation von Editiersequenzen

$$
\begin{array}{rccccccccc}
A = & & \mathsf{b} & \mathsf{a} & \mathsf{a} & \mathsf{c} & \mathsf{a} & \mathsf{a} & \mathsf{b} & \mathsf{c} \\
 & & | & | & / & \mathbf{/} & & | & / & \\
B = & \mathsf{a} & \mathsf{b} & \mathsf{a} & \mathsf{c} & \mathsf{b} & \mathsf{c} & \mathsf{a} & \mathsf{c} &
\end{array}
$$

oder mit

$$
\begin{array}{rcccccccccc}
A = & - & \mathsf{b} & \mathsf{a} & \mathsf{a} & \mathsf{c} & \mathsf{a} & - & \mathsf{a} & \mathsf{b} & \mathsf{c} \\
 & & | & | & & | & \mathbf{|} & & | & & | \\
B = & \mathsf{a} & \mathsf{b} & \mathsf{a} & - & \mathsf{c} & \mathsf{b} & \mathsf{c} & \mathsf{a} & - & \mathsf{c}
\end{array}
$$

Kosten: 5

Aufteilung einer optimalen Spur $\Rightarrow$ zwei optimale Teilspuren

Bemerkung: Dynamische Programmierung erforderlich!

Berechnung der Spur

Sei $A_i = a_1 \cdots a_i$ und $B_j = b_1 \cdots b_j$

$$D_{i,j} = D(A_i, B_j)$$

Drei Möglichkeiten eine Spur zu beenden:

1. a_m wird durch b_n ersetzt:
   $D_{m,n} = D_{m-1,n-1} + c(a_m, b_n)$

2. a_m wird gelöscht: $D_{m,n} = D_{m-1,n} + 1$

3. b_n wird eingefügt: $D_{m,n} = D_{m,n-1} + 1$

3 Rekursionsgleichung

Rekursiongleichung, falls $m, n \geq 1$:

$$D_{m,n} = \min \left\{ \begin{array}{lcl} D_{m-1,n-1} & + & c(a_m, b_n), \\ D_{m-1,n} & + & 1, \\ D_{m,n-1} & + & 1 \end{array} \right\}$$

$\Rightarrow$ Berechnung aller $D_{i,j}$ erforderlich, $0 \leq i \leq m$, $0 \leq j \leq n$.

Anfangsbedingungen

Anfangsbedingungen:

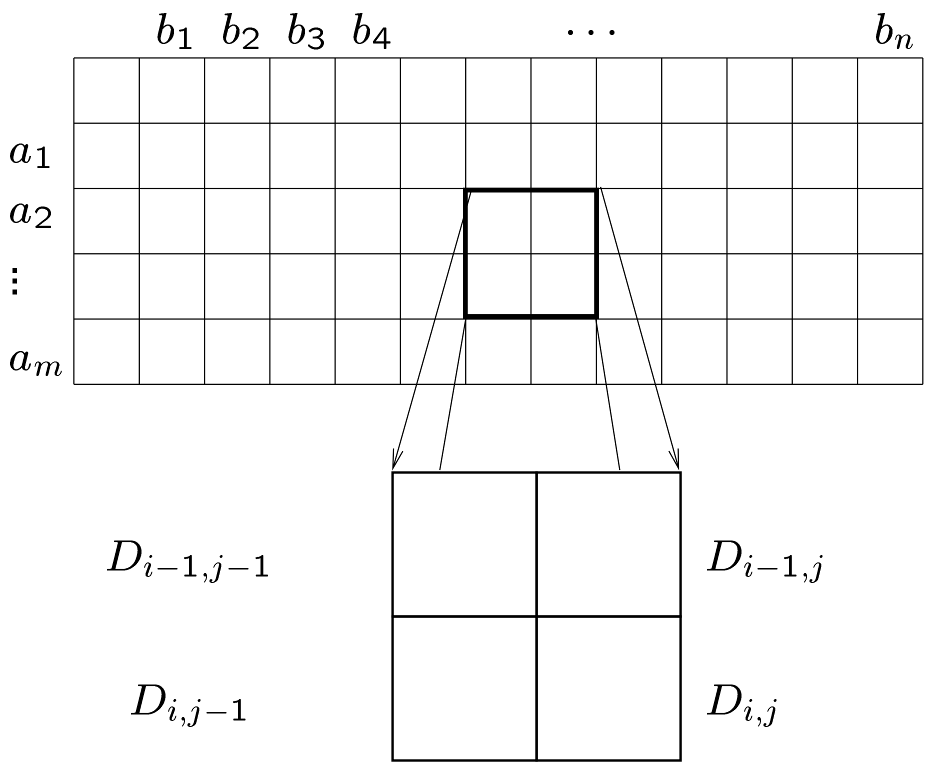
$$\begin{aligned} D_{0,0} &= D(\epsilon, \epsilon) = 0 \\ D_{0,j} &= D(\epsilon, B_j) = j \\ D_{i,0} &= D(A_i, \epsilon) = i \end{aligned}$$

Rekursionsgleichung:

$$D_{i,j} = \min \left\{ \begin{array}{lcl} D_{i-1,j-1} & + & c(a_i, b_j), \\ D_{i-1,j} & + & 1, \\ D_{i,j-1} & + & 1 \end{array} \right\}$$

4 Reihenfolge

Reihenfolge der Berechnung:



5 Algorithmus

Editierdistanz

Input: Zwei Zeichenketten $A = a_1 \cdots a_m$ und $B = b_1 \cdots b_n$

Output: Die Matrix $D = (D_{ij})$

$D[0,0] := 0$

for $i := 1$ to m do $D[i,0] = i$

for $j := 1$ to n do $D[0,j] = j$

for $i := 1$ to m do

  for $j := 1$ to n do

$$D[i,j] := \min(D[i-1,j]+1,\ D[i,j-1]+1,\ D[i-1,j-1]+c(a_i,b_j))$$

6 Beispiel

		a	b	a	c
	0	1	2	3	4
b	1				
a	2				
a	3				
c	4				

7 Editier-Operationen

Berechnung der Editieroperationen:

Editieroperationen(i, j)

Input: Berechnete Matrix D

if $i = 0$ and $j = 0$ then return

if $i \neq 0$ and $D[i, j] = D[i-1, j] + 1$

then "lösche $a[i]$"

Editieroperationen$(i-1, j)$

else if $j \neq 0$ and $D[i, j] = D[i, j-1] + 1$

then "füge $b[j]$ ein"

Editieroperationen$(i, j-1)$

else

$/* \; D[i, j] = D[i-1, j-1] + c(a[i], b[j]) \; */$

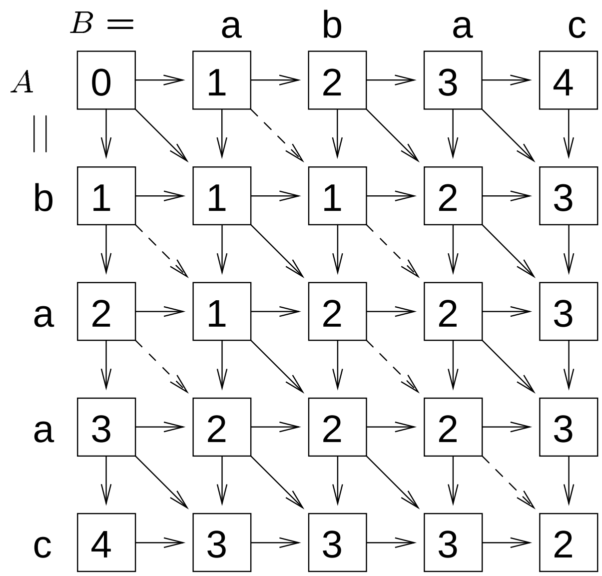
"ersetze $a[i]$ durch $b[j]$"

Editieroperationen$(i-1, j-1)$

Aufruf: Editieroperationen(m, n)

Beispiel

Editieroperationen:



Spurgraph: Übersicht über alle möglichen Spuren zu Transformation von A in B

Bemerkungen: Implizit vorher verwendet
Spurgraph: Knoten für jedes Paar (i, j)
gerichtete Kante von Knoten (i, j) zu (i_1, j), $(i, j+1)$ und $(i+1, j+1)$.
Gewichtung der Kanten entsprechend Editierkosten
Kosten nehmen entlang eines Weges monoton zu
Jeder Weg von der linken oberen Ecke zu rechten unteren Ecke entspricht einer optimalen Spur

8 Approximative Zeichenkettensuche

Gegeben: zwei Zeichenketten $P = p_1 p_2 \cdots p_m$ (Muster) und $T = t_1 t_2 \cdots t_n$ (Text)

Gesucht: Ein Intervall $[j', j]$, $1 \leq j' \leq j \leq n$, so daß das Teilwort $T_{j',j} = t_{j'} \cdots t_j$ das dem Muster P ähnlichste Teilwort von T ist, d.h. für alle anderen Intervalle $[k', k]$, $1 \leq k' \leq k \leq n$, gilt:

$$D(P, T_{j',j}) \leq D(P, T_{k',k})$$

Naives Verfahren:

for all $1 \leq j' \leq j \leq n$ do
Berechne $D(P, T_{j',j})$
wähle Minimum

Methode

for all $1 \leq j \leq n$ do

  Berechne j', so daß $D(P, T_{j',j})$ minimal ist

Für $0 \leq i \leq m$ und $0 \leq j \leq n$ sei:

$$E_{i,j} = \min_{1 \leq j' \leq j+1} D(P_i, T_{j',j})$$

Optimale Spur:

$$\begin{array}{rcccccccc} P_i = & \text{b} & \text{a} & \text{a} & \text{c} & \text{a} & \text{a} & \text{b} & \text{c} \\ & | & | & / & / & & | & / & \\ T_{j',j} = & \text{b} & \text{a} & \text{c} & \text{b} & \text{c} & \text{a} & \text{c} & \end{array}$$

Rekursionsgleichung:

$$E_{i,j} = \min \left\{ \begin{array}{l} E_{i-1,j-1} + c(p_i, t_j), \\ E_{i-1,j} + 1, \\ E_{i,j-1} + 1 \end{array} \right\}$$

Bemerkungen

Wieder drei Möglichkeiten wie die optimale Spur für $D(P_i, T_{j',j})$ endet, d.h. unabhängig von j' muß eine der drei Möglichkeiten eintreten

Aber: j' kann für $E_{i-1,j-1}$, $E_{i-1,j}$ und $E_{i,j-1}$ ganz verschieden sein.

Teilspur einer optimalen Spur ist eine optimale Teilspur, d.h. insbesondere ist j' optimal für $D(P_i, T_{j',j})$ und z.B. $E_{i,j} = E_{i,j-1} + 1$, dann ist j' auch optimal für $D(P_i, T_{j',j-1})$.

Anfangsbedingungen

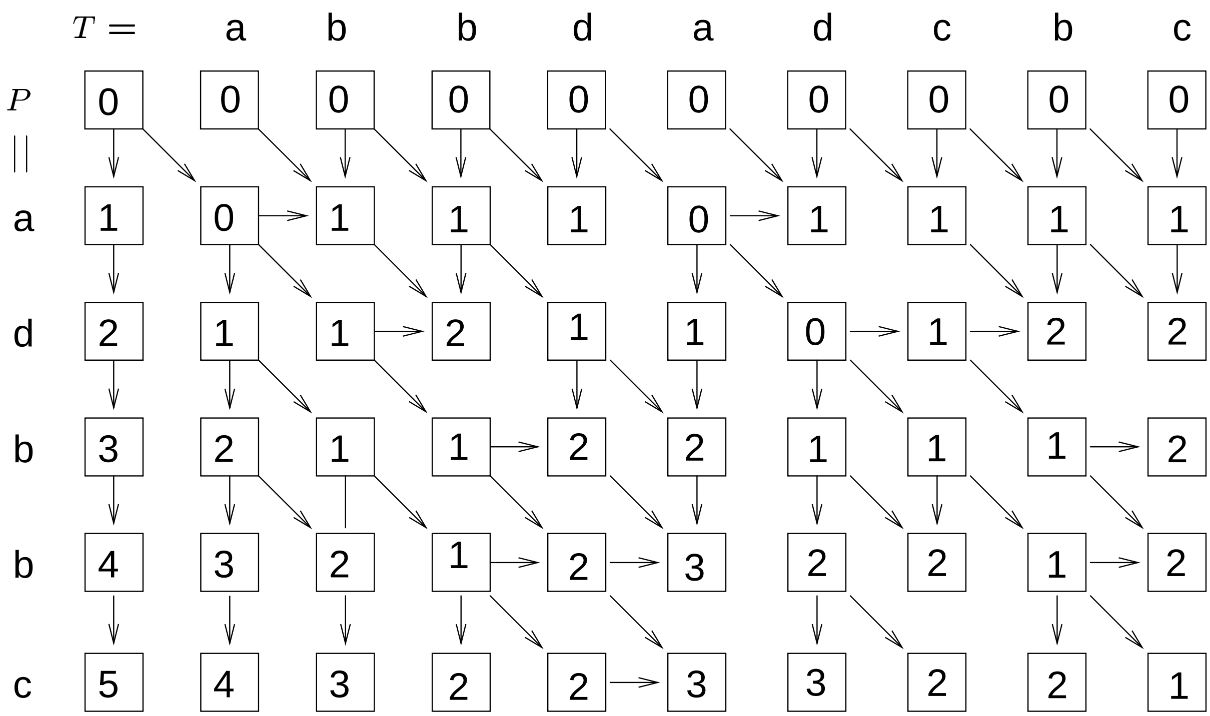
$$
\begin{aligned}
E_{0,0} &= E(\epsilon, \epsilon) = 0 \\
E_{i,0} &= E(P_i, \epsilon) = i
\end{aligned}
$$

aber

$$
E_{0,j} = E(\epsilon, T_j) = 0
$$

Beobachtung:
Die optimale Editiersequenz von P nach $T_{j',j}$ beginnt nicht mit einer Einfügung von $t_{j'}$

Abhängigkeitsgraph



Eigentlich: Suche nach Vorkommen von P in T mit Editierdistanz höchstens k

Korrektheit

Satz Gibt es im Abhängigkeitsgraphen einen Weg von $E_{0,j'-1}$ nach $E_{i,j}$, so ist $T_{j',j}$ ein zu P_i ähnlichstes, bei j endendes Teilstück von A mit

$$D(P_i, T_{j',j}) = E_{i,j}.$$

Ähnlichkeit zweier Zeichenketten

$S(A, B)$

Operationen:

1. Ersetzen eines Zeichens a durch ein Zeichen b: Gewinn $s(a, b)$
2. Löschen eines Zeichens von A, Einfügen eines Zeichens von B: Verlust $-c$

Aufgabe:
Finde eine Folge von Operationen zur Umwandlung von A in B, so daß die Summe der Gewinne maximiert wird.

Motivation

Problem aus der Biologie: Berechnung der Ähnlichkeit von Proteinen.

Jedes Protein ist eine lineare Verknüpfung von 250-2000 Aminosäuren

Es gibt 20 Aminosäuren, aus denen alle Proteine aufgebaut sind, d.h. jedes Protein kann als Zeichenkette über einem Alphabet der Größe 20 aufgefäßt werden.

Manche Aminosäuren sind sich in ihren Eigenschaften (z.B. hydrophil-hydrophob) ähnlicher als andere

Ähnlichkeit zweier Zeichenketten nur dann sinnvoll, wenn jedes Zeichen nur in einer Operation vorkommt

Ähnlichkeit von Zeichenketten

$S_{i,j} = S(A_i, B_j)$, $0 \leq i \leq m$, $0 \leq j \leq n$

Rekursionsgleichung:

$$
\begin{aligned}
S_{m,n} \quad = \max \quad & (S_{m-1,n-1} + s(a_m, b_n), \\
& S_{m-1,n} - c, S_{m,n-1} - c)
\end{aligned}
$$

Anfangsbedingungen:

$$
\begin{aligned}
S_{0,0} &= S(\epsilon, \epsilon) = 0 \\
S_{0,j} &= S(\epsilon, B_j) = -jc \\
S_{i,0} &= S(A_i, \epsilon) = -ic
\end{aligned}
$$

Ähnlichste Teilzeichenketten

Gegeben: zwei Zeichenketten $A = a_1 a_2 \cdots a_m$ und $B = b_1 b_2 \cdots b_n$

Gesucht: Ein zwei Intervalle $[i', i] \subseteq [1, m]$ und $[j', j] \subseteq [1, n]$, so daß:

$$S(A_{i',i}, B_{j',j}) \geq S(A_{k',k}, B_{l',l}),$$

für alle $[k', k] \subseteq [1, m]$ und $[l', l] \subseteq [1, n]$.

Naives Verfahren:

for all $[i', i] \subseteq [1, m]$ and $[j', j] \subseteq [1, n]$ do
    Berechne $S(A_{i',i}, B_{j',j})$

Methode

for all $1 \leq i \leq m$, $1 \leq j \leq n$ **do**

  Berechne i' und j', so daß $S(A_{i',i}, B_{j',j})$ maximal ist

Für $0 \leq i \leq m$ und $0 \leq j \leq n$ sei:

$$H_{i,j} = \max_{\substack{1 \leq i' \leq i+1, \\ 1 \leq j' \leq j+1}} S(A_{i',i}, B_{j',j})$$

Optimale Spur:

$$\begin{array}{rccccccccc}
A_{i',i} = & \mathrm{b} & \mathrm{a} & \mathrm{a} & \mathrm{c} & \mathrm{a} & - & \mathrm{a} & \mathrm{b} & \mathrm{c} \\
 & | & | & & | & \mathbf{|} & & | & & | \\
B_{j',j} = & \mathrm{b} & \mathrm{a} & - & \mathrm{c} & \mathrm{b} & \mathrm{c} & \mathrm{a} & - & \mathrm{c}
\end{array}$$

Rekursionsgleichung

:

$$H_{i,j} = \max \left\{ \begin{array}{l} H_{i-1,j-1} + s(a_i, b_j), \\ H_{i-1,j} - c, \\ H_{i,j-1} - c, \\ 0 \end{array} \right\}$$

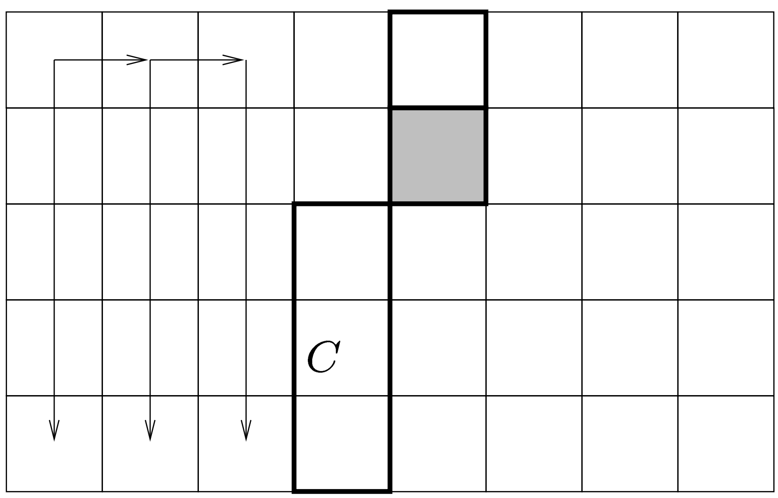
Anfangsbedingungen:

$$\begin{aligned} H_{0,0} &= H(\epsilon, \epsilon) = 0 \\ H_{i,0} &= H(A_i, \epsilon) = 0 \\ H_{0,j} &= H(\epsilon, B_j) = 0 \end{aligned}$$

9 Editier-Distanz mit linearem Platz

Berechnung von $D_{m,n}$ mit Speicherplatz $O(\min\{m, n\})$:

Sei $m \leq n$: Idee spaltenweise Berechnung von $D_{i,j}$



Editier-Distanz mit linearem Platz

Editierdistanz

Input: Zeichenketten A und B mit Länge m und n

for $i := 0$ to m do $C[j] := i$
for $j := 1$ to n do
  $/*$ Berechne $D[0, j], \ldots, D[m, j]$ $*/$
  $/*$ Inv.: $C[i] = D[i, j-1]$, $i = 0, \ldots, m$ $*/$
$F := C[0]$ $\quad /* \ = D[0, j-1] \ */$
$C[0] := C[0] + 1$ $\quad /* \ = D[0, j] \ */$
for $i := 1$ to m do
  $E := F$ $\quad /* \ = D[i-1, j-1] \ */$
  $F := C[i]$ $\quad /* \ = D[i, j-1] \ */$
  $C[i] := \min(E + c(a_i, b_j),$
    $C[i-1] + 1, F + 1)$
return $C[m]$