2021. november 2.

Karaktersorozatok hasonlósága

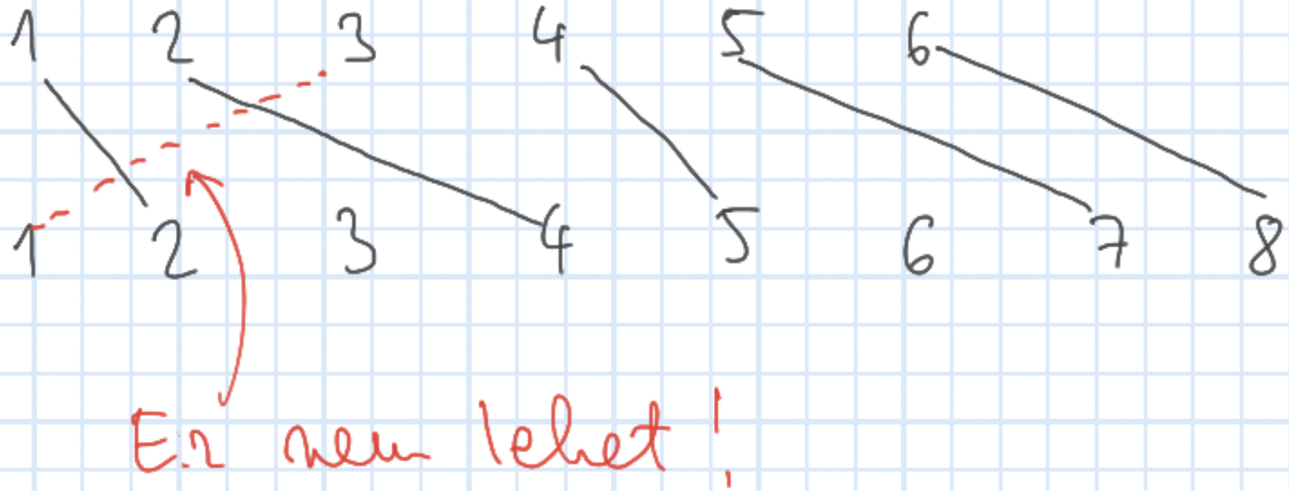
Szekvenciaillesztés (Levenstein távolság)

Adottak az $X = (x_1, x_2, \dots, x_m)$ és $Y = (y_1, y_2, \dots, y_n)$ karaktersorozatok. A hasonlóságuk egy lehetséges mértéke a következő (elég általános lesz ez):

Tekintsük az X és Y pozícióinak $\{1, 2, \dots, m\}$ és $\{1, 2, \dots, n\}$ halmazát, és legyen M egy páron-

tása e két halmaznak.

M-et reflexiós illesztésnek nevezzük ha $(i,j) \in M$, $(i',j') \in M$ és $i < i'$ esetén $j < j'$



OK!

A reflexiós illesztés egy vizualizációja (példa):

Legyen a két karaktersorozat:

1 2 3 4 5 6 7 8 9
G A T C G G C A T

C A A T G T G A A T C
1 2 3 4 5 6 7 8 9 10 11

Legyen a szerveniaillesztés:

$\{(1,1), (2,3), (3,4), (5,5), (6,7), (7,8), (8,9), (9,10)\}$

$\Rightarrow$ G - A T C G - G C A T -
C A A T - G T G A A T C

Ez a megjelenítés feltogható mint az első karak-

tersorozat áttranszformálása a másodikba, a szekvencia-illesztéssel megfelelően:

cseréljük le az első G betűt C-re, majd írjunk be egy A-t, aztán töröljünk a negyedik pozíción a C-t, és így tovább ...

Megjegyzés

Két karaktersorozat közötti szekvenciaillesztés megadható → optimális ???

Egy M szerencicillesztés költségét definiáljuk így:

- felsználunk adott $g > 0$ költséget minden olyan X- és Y-beli pozícióra, amelyek nem szerepelnek M-ben
- ha $(i,j) \in M$, akkor számoljuk fel $c[x_i, y_j] \geq 0$ költséget erre a párra, ahol a $c["p","q"]$ mennyiségek minden adottak ($c["p","p"] = 0$ tipikusan, de nem feltétlenül)

Adjuk össze ezeket!

Feladat

Adjunk hatékony algoritmust a minimális költségű szerveziállítás meghatározására!

Megjegyzés

Ha ez a költség kicsi, akkor a két karakterszorzatot hasonlóan tekintjük, ha nagy, akkor különbözőnek

Dinamikus programozás megoldás

① Részproblémák

$R[i,j]$ legyen az $X_i = (x_1, x_2, \dots, x_i)$ és $Y_j = (y_1, y_2, \dots, y_j)$ kezdőszeletek (prefixek) optimális sorrenciaillesztésének meghatározása

② optimális részstruktúra tul.

Észrevétel

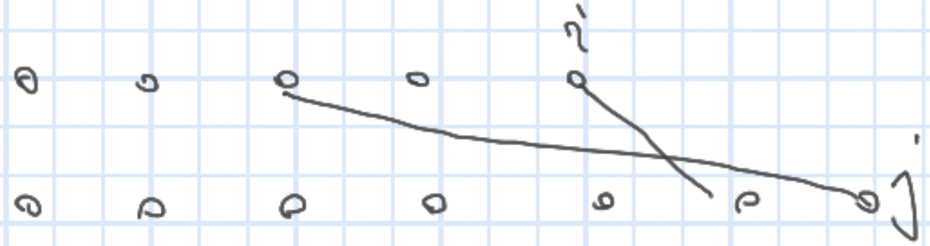
Legyen M egy sorrendiállítás X_i és Y_j között.
Ekkor három lehetőség van:

① $(i,j) \in M$

② i nem szerepel M-ben első tagként

③ j nem szerepel M-ben második tagként

Ami nem lehet:



Állítás

Legyen M opt. sorrenciaillesztés X_i és Y_j között

① ha $(i,j) \in M$, akkor $M \setminus \{(i,j)\}$ opt. sorr. illesztés X_{i-1} és Y_{j-1} között

② ha i nem szerepel M-ben első tagként, akkor M opt. sorr. illesztés X_{i-1} és Y_j között

③ ha j nem szerepel M-ben második tagként, akkor M opt. sorr. illesztés X_i és Y_{j-1} között

Biz

Indirás CUT & PASTE

Pl ① Ha $M \setminus \{(i,j)\}$ nem optimális, akkor egy másik kisebb költségű megoldásban x_i-her és y_j-her

③ Rekurzió

Jelölje $A[i,j]$ az $R[i,j]$ opt. megoldásának

költségét.

Nyilván $A[i,0] = ig$ és $A[0,j] = jg$.

Ha $i,j \neq 0$, akkor a három esetnek megfelelően

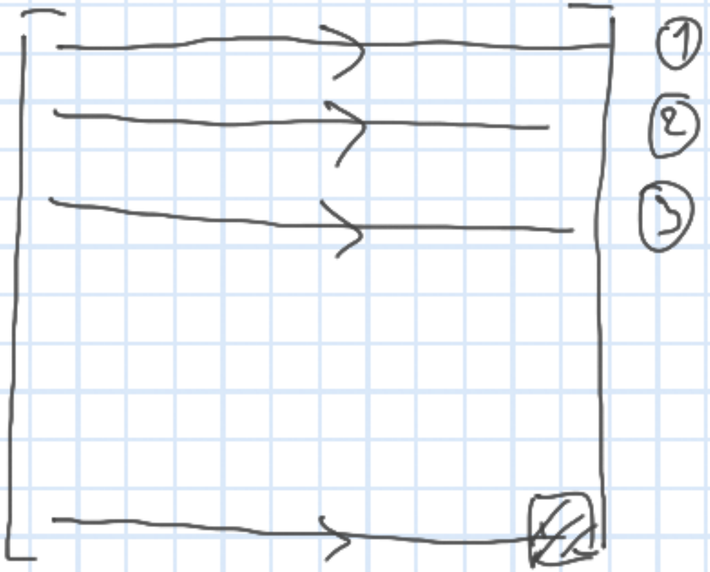
- $A[i,j] = A[i-1, j-1] + c[x_i, y_j]$
- $A[i,j] = A[i-1, j] + g$
- $A[i,j] = A[i, j-1] + g$

Nem tudjuk melyik ág fenn, így megvizsgáljuk mindhármat és a legkedvezőbbet választjuk:

$$A[i,j] = \min \begin{cases} A[i-1, j-1] + c[x_i, y_j] \\ A[i-1, j] + g \\ A[i, j-1] + g \end{cases}$$

④ optimális megoldás

$A =$



$A[m,n]$ opt. megoldás

Költség : $O(mn)$ cella a cellánként $O(1)$ számolás $\rightarrow O(mn)$

⑤ opt rekv. áll.

Minden $A[i,j]$-nél feljegyezzük, hogy a három esetből melyik volt a legkedvezőbb, ebből visszafejthető az opt. rekv. áll.