2021. november 9.

Karaktersorozatok hasonlósága

Más megközelítés (részletesen gyakorlaton)

→ leghosszabb közös részsorozat

Adottak az $X = (x_1, x_2, \ldots, x_m)$ és $Y = (y_1, y_2, \ldots, y_n)$ karaktersorozatok. Azt mondjuk, hogy a $Z = (z_1, z_2, \ldots, z_k)$ karaktersorozat közös részsorozata X-nek és Y-nak, ha

1 létezik olyan $1 \leq i_1 < i_2 < \dots < i_k \leq m$ index, hogy $x_{i_1} = z_1, x_{i_2} = z_2, \dots, x_{i_k} = z_k$.

2 létezik olyan $1 \leq j_1 < j_2 < \dots < j_k \leq n$ index, hogy $y_{j_1} = z_1, y_{j_2} = z, \dots, y_{j_k} = z_k$

Adjunk hatékony algoritmust X és Y egy leghosszabb közös részsorozatának meghatározására!

Minél hosszabb ez, annál hasonlóbbnak tekintjük X-et és Y-t.

Ez a feladat is megoldható dinamikus programozással $O(nm)$ költséggel az optimális szekvenciaillesztéssel nagyon hasonló módon

<u>Részproblémák</u>:

$R[i,j] \rightarrow$ $X_i = (x_1, x_2, \dots, x_i)$ és $Y_j = (y_1, y_2, \dots, y_j)$ egy leghosszabb közös részsorozatának meghatározása

. . .

NP-nehéz feladatok és Dinamikus Programozás

① Részhalmazösszeg

$a_1, a_2, \ldots, a_n, B$ pozitív egészek

Van-e olyan $I \subseteq \{1, 2, \ldots, n\}$, hogy $\sum_{i \in I} a_i = B$

(híres NP-teljes feladat)

② Pénzváltás

$a_1, a_2, \ldots, a_n, B$ pozitív egészek

Vannak-e olyan $\alpha_1, \alpha_2, \dots, \alpha_n$ nem negatív egészek, hogy $\sum_{i=1}^{n} \alpha_i a_i = B$

$\boxed{\text{Optimalizálás}}$

Igenlő válasz esetén keressünk olyan $\alpha_1, \alpha_2, \dots, \alpha_n$ nem negatív egészeket, melyekre $\sum_{i=1}^{n} \alpha_i$ minimális

<u>Speciális eset</u>

$a_1, a_2, \dots, a_n$ között szerepel 1 (ekkor létezik megoldás)

③ Hátizsák

$t_1, t_2, \dots, t_n$ tárgyak

t_i tárgy $\rightarrow$ w_i súly, v_i érték $(i = 1, 2, \dots, n)$

hátizsák W kapacitással

$w_1, v_1, w_2, v_2, \dots, w_n, v_n, W$ pozitív egészek

keressünk olyan $I \subseteq \{1, 2, \dots, n\}$ indexhalmazt,

hogy $\sum_{i \in I} w_i \leq W$ és $\sum_{i \in I} v_i$ maximális

② - ③ kapcsolat nem teljesen világos

① - ③ kapcsolat

Részhalmazösszeg speciális esete a hátizsák feladatnak:

Legyen $a_1, a_2, \ldots, a_n, B$ a részhalmazösszeg feladat egy bemenete (példánya), ehhez konstruálunk a hátizsákfeladathoz egy bemenetet:

$t_1, t_2, \dots, t_n$ tárgyak

$t_i \to a_i$ súly, a_i érték $\quad (i = 1, 2, \dots, n)$

hátizsák kapacitása B.

Tfh megoldottuk a hátizsák feladatot, azaz megvan egy olyan $I \subseteq \{1, 2, \dots, n\}$, amelyre $\sum_{i \in I} a_i \leq B$ és $\sum_{i \in I} a_i$ maximális

A maximum nyilván legfeljebb B.

- Ha épp B, akkor a részhalmazösszeg feladatra igen a válasz, a maximumot adó I-vel

$$\sum_{i \in I} a_i = B$$

- Ha kisebb, mint B, akkor a válasz nem!

"Következmény"

Hátizsák feladat NP nehéz (visszavezethető rá egy NP-teljes feladat)

Mi van ha egy fontos feladatról kiderül, hogy NP-nehéz?

Lehetséges megközelítés

- mekkora a mérete a tipikus feladatoknak?
  (kis méretű feladaton exponenciális algoritmus is befejeződhet belátható időn belül)
- van-e valami specialitása a tipikus feladatoknak?
  (spec. bemeneteken lehet polinomiális egy általánosan exponenciális algoritmus)

• Tényleg az optimális megoldás jó csak? Egy attól kicsit rosszabb nem teri meg?

→ közelítő algoritmusok

Dinamikus programozás megoldás a hátizsák feladatra

(nem polinomiális!)

① Részproblémák

$R[i,j]$ legyen az a feladat, amikor a $t_1, t_2, \ldots, t_i$ tárgyak közül választhatunk, és a hátizsák kapacitása j

(2) Optimális részstruktúra tulajdonság

Legyen σ egy optimális megoldása $R[i,j]$-nek

Két eset van:

(A) $t_i \notin \sigma$, akkor σ optimális megoldása $R[i-1, j]$-nek is

(B) $t_i \in \sigma$, akkor $\sigma \setminus \{t_i\}$ optimális megoldása $R[i-1, j-w_i]$-nek

Bizonyítás a vonalasor CUT & PASTE

③ Rekurzió

Jelölje $c[i,j]$ az $R[i,j]$ optimális megoldásában az összértéket

Ha $i=0$ vagy $j=0$, akkor $c[i,j]=0$ nyilván

Ha $i>0$ és $j>0$, akkor pedig

- $c[i,j] = c[i-1,j]$ ha $j < w_i$
- ha $j \geq w_i$, akkor ② -ből bármelyik

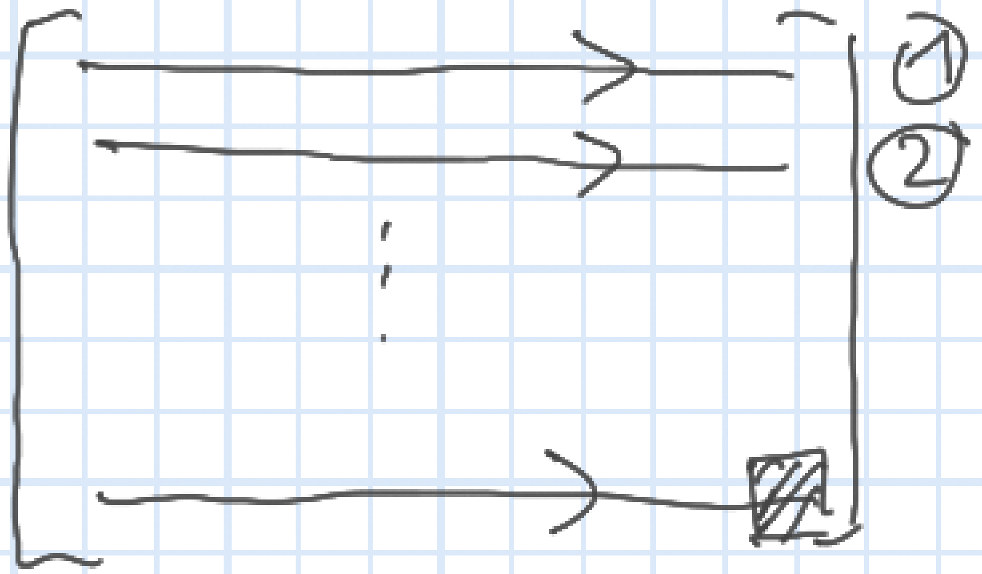
előfordulhat, mindkettőt megvizsgáljuk és a kedvezőbbet választjuk:

$$c[i,j] = \max\{v_i + c[i-1, j-w_i],\ c[i-1,j]\}$$

④ Számolás

Az $n+1$ sorból és $W+1$ oszlopból álló $c[i,j]$ táblázatot töltjük ki:

$c[i,j] =$



Opt. érték

cellánként $O(1)$ költség $\Longrightarrow$ összköltség $O(nW)$

5. Optimális "I" meghatározása

$c[i,j]$ számolásakor jegyezzük fel, hogy t_i beválasztása vagy kihagyása volt kedvezőbb

jobb alsó sarokból indulva egy optimális "I" meghatározható

Fontos

$O(nW)$ nem polinomiális

W pl. egész, mint bemenet csak $O(\log W)$ méretű, ehhez képest W exponenciálisan nagy!

Megjegyzés

Ha a súlyokra $w_i = O(n^k)$ teljesül fix k-val, akkor az alg. polinomiális.