ADATSZERKEZETEK

Mintaillesztés

MINTAILLESZTÉS

- Adott egy karakterlánc
- elemek Σ ábécéből

S karakterlánc — T minta

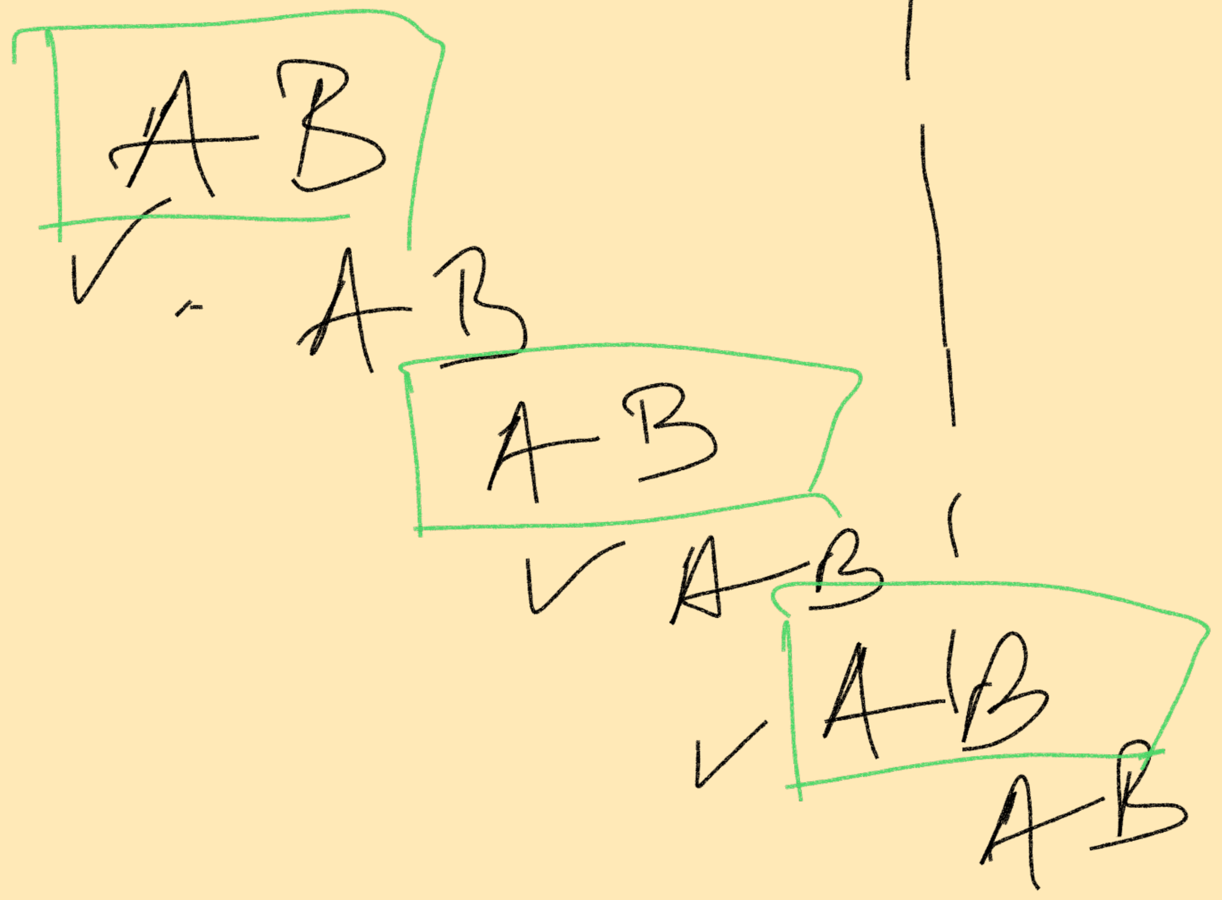
- Hol fordul elő S-ben a T?

S – n kar. $\qquad m \leq n$

T – m kar

① Egyszerű mintaillesztés (brute force)

S: ABABABC $\qquad$ T: AB



$\boxed{\text{illeszkedik}}$

$T[1..m] = S[i+1,\ldots,i+m]$

$\boxed{\text{eltolás}}$

$S[i+1,\ldots,i+m]$

i eltolást alkalmaz

$\boxed{\text{érvényes eltolás}}$

eltol + illeszkedik

- próbálkozások száma: n
- illeszkedés ellenőrzése: m

$$\boxed{O((n-m+1)m)}$$

1. 500.

$\cancel{A}\,\boxed{B}$ . . . . . B $\cancel{A}$ B

$\xrightarrow{\quad\quad\quad\quad}$

i

$\boxed{A}$B

$S[i+1] \sim$ 'A'

1, 500 helyen A van

- nincs előfeldolgozás
- nem használ fel ismereteket

for $i = 0 \to n-m+1$

Ellenőrizzük

$T[1, \ldots, m] = S[i+1, i+2, \ldots, i+m]$

// vagy illeszkedik

/// vagy az első eltérésnél befejezzük az ellenőrzést

Ha illeszkedik $\to$ kiír poz.

Minta: belső szerkezet

$$S: \overset{1.}{A}\,\overset{2.}{B}\,\overset{3.}{A}\,B\,A\,B\,C \qquad T: \boxed{\underline{ABC}}$$

AB

→ nem lesz megfelelő

+1

$\xrightarrow{+2}$ AB

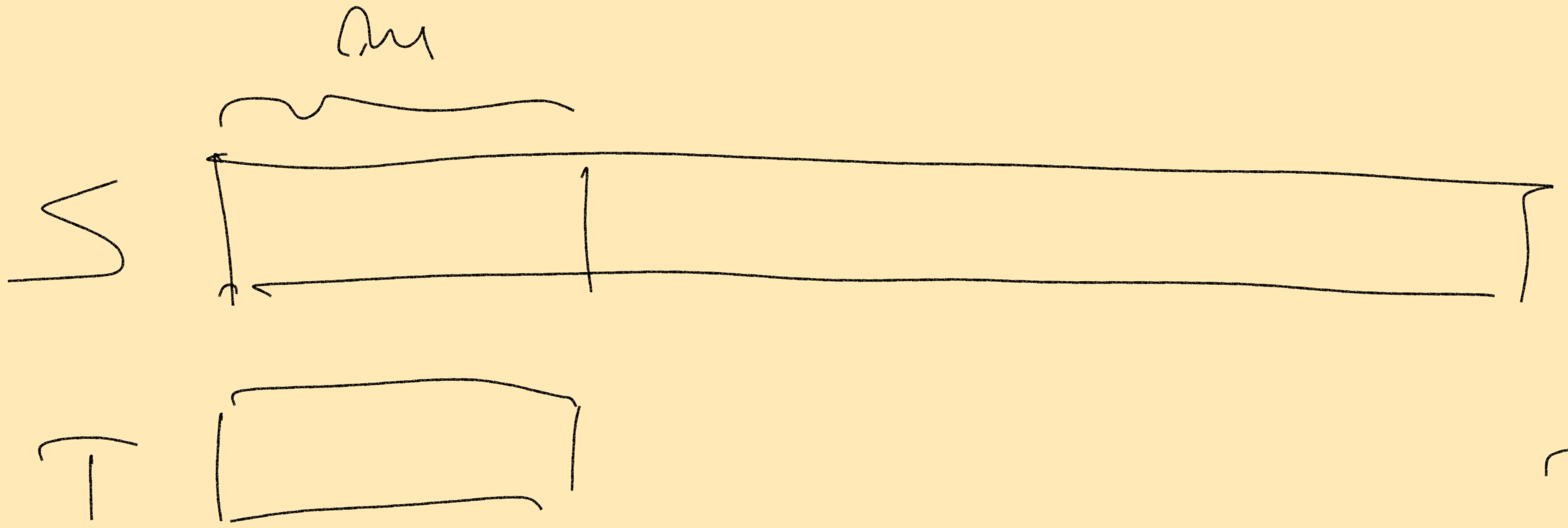
– vizsgáljuk a lehetséges eltolásokat

– csak akkor hajtjuk végre az illesztést, ha teljesülnek bizonyos alapfeltételek.

② Rabin-Karp algoritmus

S, T $\quad \Sigma$ ábécé

legyen $d = |\Sigma|$ $\Rightarrow$ S és T egy-egy d számrendszerbeli szám



T mint szám egyenlő-e $S[i+1, \ldots, i+m]$ „számmal”?

Példa

S: 1 2 5 3 7 9 8 6 2 5 8 T: 2 5

Legyen $s_i = S[i+1, \dots, i+m]$ // mint szám

$t = T[1, \dots, m]$ // mint szám

$s_0, s_1, s_2 \dots, s_{n-m+1}$ közül ki egyenlő t-vel

$$s_0 = d^{m-1} S[1] + d^{m-2} S[2] + d^{m-3} S[3] + \dots + S[m]$$

(Horner séma) $$= S[m] + d(S[m-1] + d(S[m-2] + \dots d S[0])\dots))$$

Példa

$S: 1\,2\,5\,3\,7\,9\,8\,6\,2\,5\,8$ $\qquad$ $T: 2\,5\,|\,7$

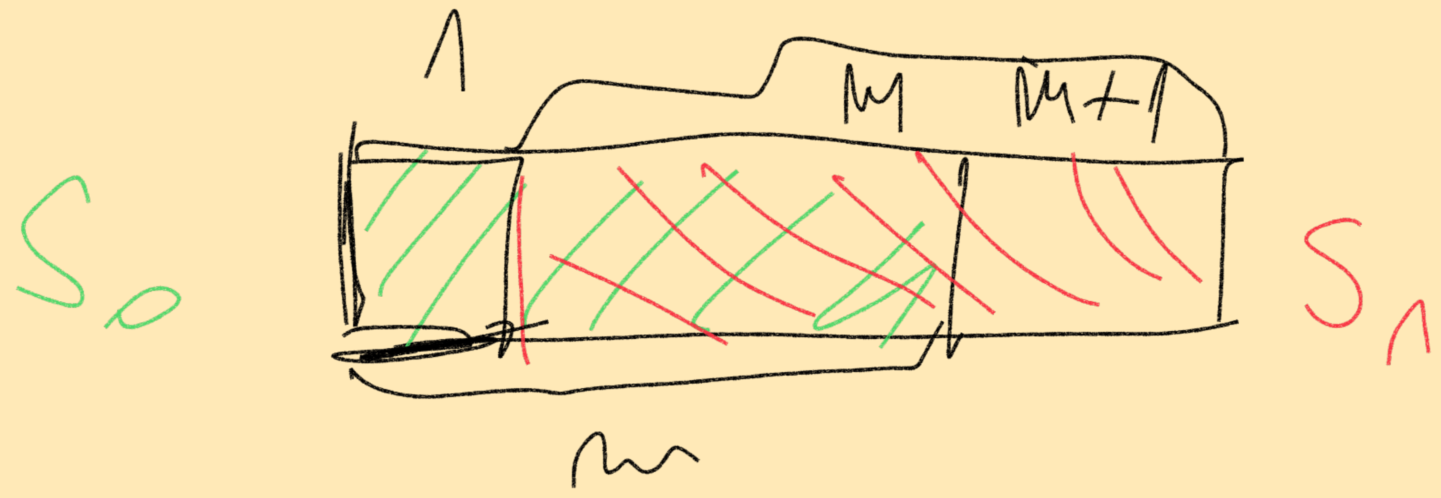
(25) $\quad S_0 = 10^1 \cdot 1 + 10^0 \cdot 2 = 12$

(258) $\quad S_0 = 10^2 \cdot 1 + 10^1 \cdot 2 + 10^0 \cdot 5 = 125 \qquad (\Theta(m))$

$S_0 \to S_1 \qquad \Theta(1) \qquad m$

$$S_1 = d\left(S_0 - d^{m-1} \cdot S[1]\right) + S[m+1]$$

$$S_i = d\left(S_{i-1} - d^{m-1} \cdot S[i+1]\right) + S[i+m+1]$$



Előnyök: illesztés konstans idő: $s_i \stackrel{?}{=} t$

Hátrány: nagy számok.

Megoldás: maradékosztályok

legyen $\boxed{q}$ tetszőleges szám

$$s_i \bmod q = t \bmod q$$

$$s_i \equiv t \pmod{q}$$

$$s_i = d\left(s_{i-1} - \boxed{d^{m-1}}\, S[i]\right) + S[i+m]$$

m számjegyű szám

d számrendszerben

$$s_i = d\left(s_{i-1} - \underline{d^{m-1}\, S[i]}\right) + S[i+m] \pmod{q}$$

$$h \equiv d^{m-1} \pmod{q}$$

$$\boxed{s_i = d\left(s_{i-1} - h\, S[i]\right) + S[i+m] \pmod{q}}$$

Mellékhatás:

$$s_i \equiv t \Rightarrow s_i \equiv t \pmod{q}$$

$$\cancel{\Leftarrow}$$

– ha egyezést találunk → teljes illesztés ellenőrzés

$\boxed{\text{Ötlet}}$: két q-val (q_1, q_2) páronként relatív prím számokkal

$\mathrm{lnko}(q_1, q_2) = 1$.

Következtetés — valós feladatoknál aránylag kicsi a hamis találat lehetősége

— q — prím és nagyobb mint s hossza

$\Rightarrow$ konstans számú találat

- Legrosszabb eset: $O((n-m+1)m)$
- Átlagos eset: $O(n) + O\left(m\left(v + \frac{n}{q}\right)\right)$

v – az érvényes eltolások száma

3. KMP (Knuth–Morris–Pratt) algoritmus

- belső szerkezetet vizsgál: a minta összetétele

S: ababc   T: abc

a b a b c

a b

+2 → a b c → találat

<u>Alg</u>: előfeldolgozás

Határozzunk meg egy $\boxed{\text{Prefix függvényt}}$

<u>Prefix függvény</u>: $q = \overline{1, m}$ határozzuk meg

P_q ($P_q = P[1 \ldots q]$) leghosszabb

valódi prefixét, ami szuffix is.

"valódi prefix": pl. ABC valódi prefixei: A, AB

	A	C	A	B	A	C	A	C
q	1	2	3	4	5	6	7	8
PF	0	0	1	0	1	2	3	2

$\boxed{\text{Prefix függvény}}$

q = 2 VP: A Sz: ε

q = 3 VP: A, AC Sz: A, CA

q = 4 VP: A, AC, ACA Sz: B, AB, CAB

q = 5 VP: A, AC, ACA, ACB Sz: A, BA, ABA, CABA

q = 6 VP: A, AC, ACA, ACB, ACBA Sz: C, AC, BAC, ABAC, CABAC

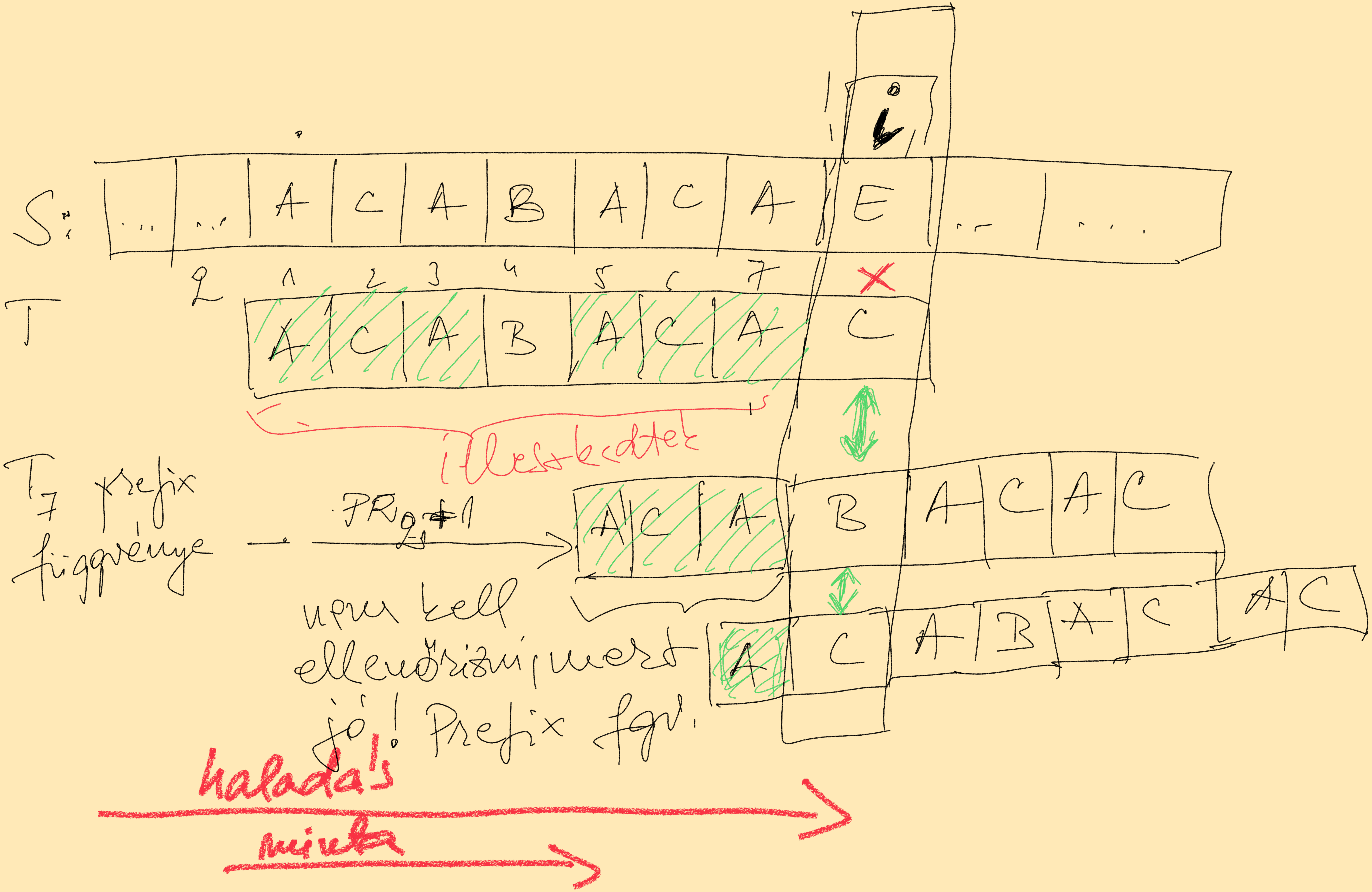
$\boxed{AC}$

$\boxed{ACA}$

$\boxed{ACAB}$

$\boxed{ACABA}$

$\boxed{ACABAC}$


S:
... ... A C A B A C A E ... ...
i
T
A C A B A C A C
1 2 3 4 5 6 7
illeszkedtek
T7 prefix függvénye
PR2+1
A C A B A C A C
nem kell ellenőrizni, mert jó! Prefix fgv.
A C A B A C A C
haladás
minta

Lehetőségek

- $S[i]$ nem egyenlő $P[q]$-val
  - $q = PR_{q-1} + 1$
  - újrakezdem $[i]$-re
  - amíg PR_{q-1} 0 lesz → léptetem i-t
- $S[i] = P[q]$ → ha $q = m$ akkor kiír
  - $q = PR_{q-1} + 1$

abbababab aba

abb

a aba

ab

→ kiír

Futási idő: előfeldolgozás: $O(m)$ (amortizált)

illesztés: $\boxed{O(n)}$

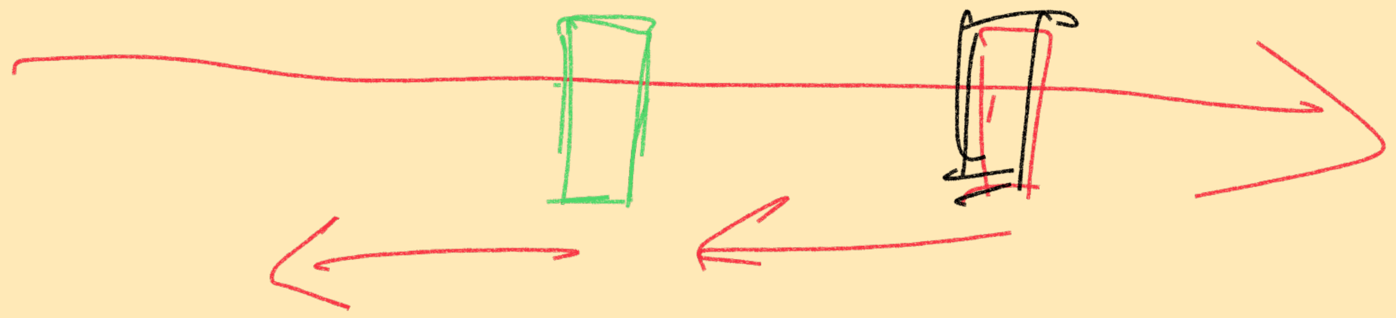
$O(m+n)$

$m \ll n$

$O(n)$

4. Boyer – Moore algoritmus

- ugrásokkal mozgatja az ablakot (mintát)
- haladási irány
  illesztési irány

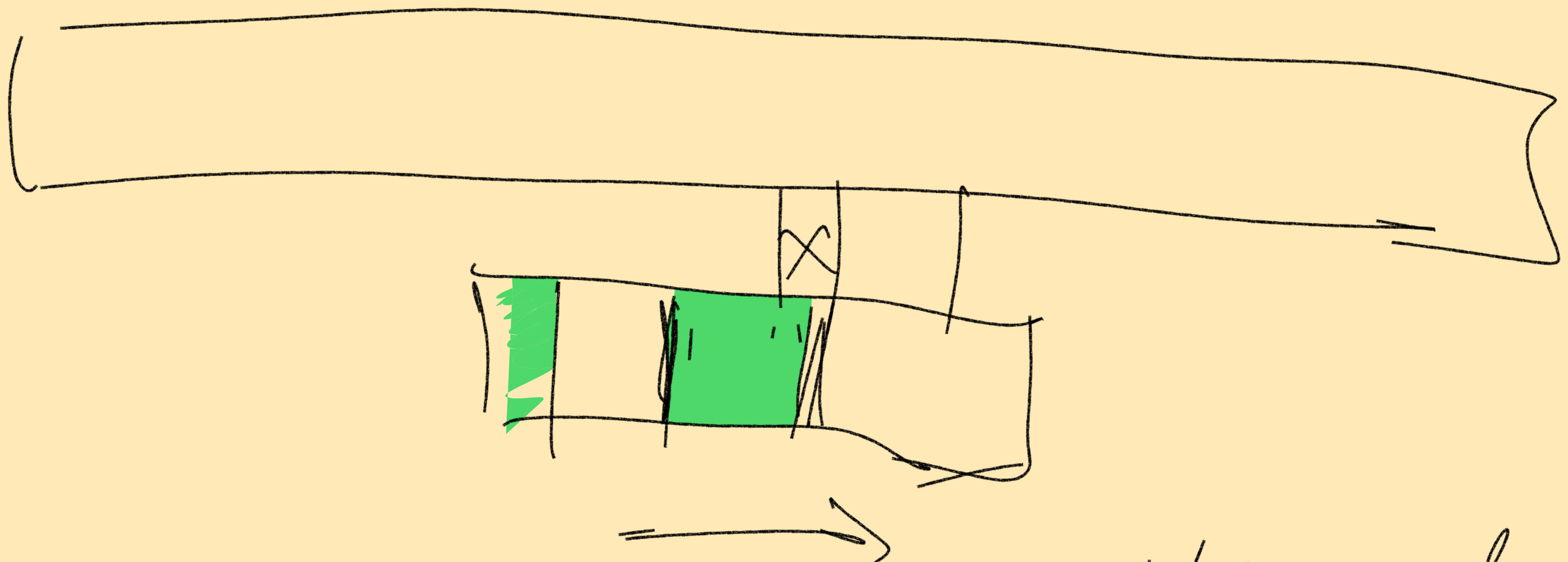


a) Rossz karakter heurisztika $S[i+m] = x$

ha $P[m] \neq x$

akkor az alkalmazott eltolás $l_1 = x$ utolsó előfordulása a mintában

b) Jó szuffix heurisztika



annyit mozdítja hátra, hogy az utolsó jó szuffix a helyére kerüljön,

ez eltolás

Tényleges eltolás: $\max(e_1, e_2)$

Futási idő $O(|\Sigma| + m)$

előfeldolgozás

legrosszabb eset

$O((n-m+1)m + |\Sigma|)$