

# Algoritmusok és adatszerkezetek 2.

Fekete István előadása alapján

Készítette: Nagy Krisztián

## 8. előadás

### VIII. Mintaillesztés (String keresés)

Minta (rövidebb szöveg) keresése egy (hosszabb) szövegben. Első előfordulás.

$H$  ábécé,

Szöveg:  $S[1..n] \in H^*$

Minta:  $M[1..m] \in H^*$

Általában:  $m \ll n$

Példa:

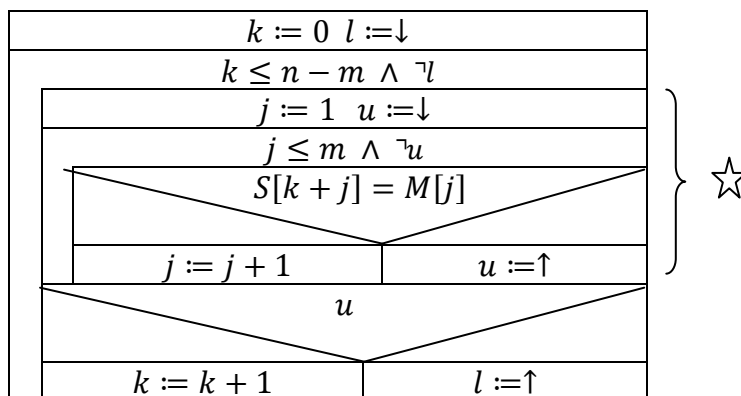
$S: A B B A B C A B$

$M: A B C$

Illesztkezés: 4. pozíción van vagy  $k = 3$  eltolás

Naív megoldás:

$M$  csúsztatása  $S$ -en (alatta vizuálisan), minden pozícióban egy programozás módszertanból ismert Lineáris keresés 2 egy nem egyenlő karakterpárig, ha ilyet találunk, akkor tovább toljuk, ha nem, akkor megtaláltuk az illesztkezést.



☆ Lineáris keresés 2 az első nem azonos karakterpárig

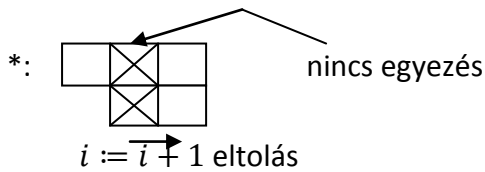
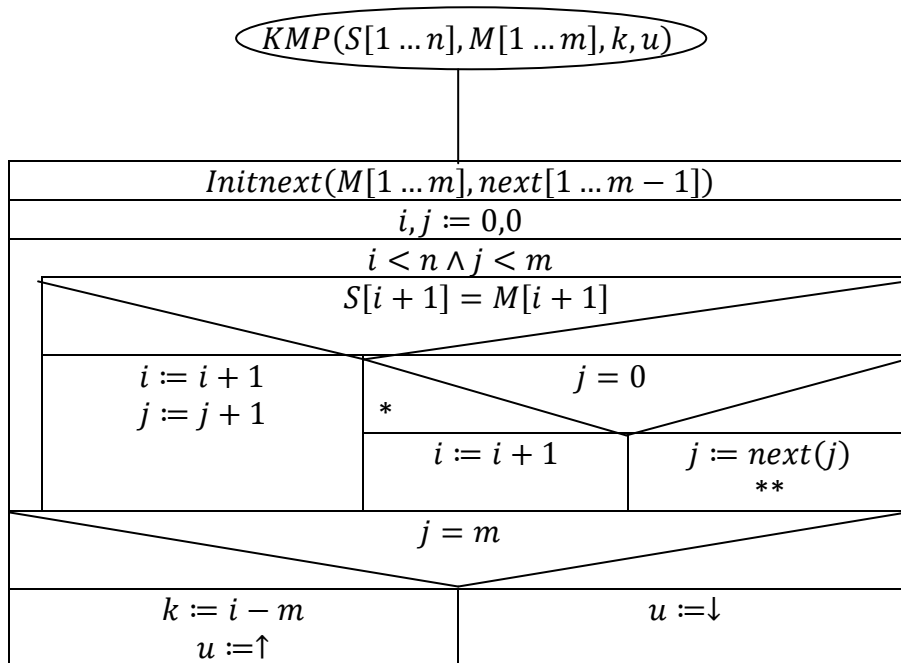
Műveletigény:  $M \notin S \ M \notin S$

$T(n, m) = \theta((n - m + 1) \cdot m) = \theta(nm) \rightarrow$  súlyos művelet igény (ne használjuk!)



Megjegyzések:

- $next(m)$  nincs értelmezve, mert ha mind az  $m$  karakter illeszkedik, akkor készen vagyunk.
- $next(j) = 0$  jelentése: nincs egyező  $p = s$  pár (Ezt az esetet a KMP külön ágon kezeli)
- $next(j)$  maximális értéke  $j - 1$ , mert  $j$ -nek nem lenne értelme. Nem valódi  $p = s$  pár  $\Rightarrow$  nem adódna eltolás
- $next(1) = 0$  mindig! (Ennek hatása a KMP-ben  $M$ -et 1-gyel jobbra csúsztatjuk)



\*\* : Például az 5. péda.  $j := next(j)$  éppen úgy állítja be  $j$ -t, hogy a következő összehasonlítandó karakterpár  $i + 1 \ j + 1$ . 5. példában:  $i + 1 \rightarrow Y \ j + 1 \rightarrow B$

Műveletigénye:  $\theta(n + m)$ -es  $\rightarrow$  egy teljes nagyságrendet javít a naív megoldáshoz képest ☺