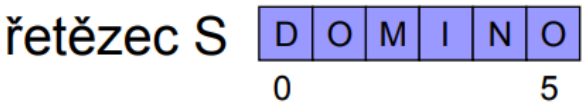
26. Prohledávání řetězců- terminologie, princip, přirozené prohledávání, KMP, chybová funkce

Co je prohledávání řetězců

- Je dán řetězec T (text) a vyhledávaný řetězec P (vzor, pattern)
- Hledáme vzor P uvnitř textu T
- Aplikace: textové editory, částečně vyhledávání na webu, zpracování obrazu, strukturní rozpoznávání aj.

Základní pojmy

- Označme S řetězec o velikosti m
- Podřetězec S[i:j] je část řetězce S mezi index i a j
- Prefix (předpona) S je podřetězec S[0:i]
- Suffix (přípona) S je podřetězec S[i:m-1], kde i je index mezi 0 a m-1

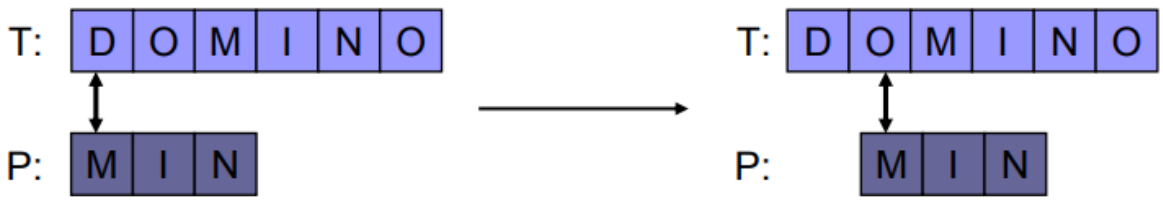


- Podřetězec S[1:3] = OMI
- Všechny prefixy S: domino, domin, domi, dom, do, d
- Všechny suffixy S: o, no, ino, mino, omino, domino

- Každé slovo je prefixem i sufixem sebe sama – takový prefix nazýváme nevlastní
- Abeceda A je konečná množina znaků, ze kterých tvoříme T a P
- Velikost abecedy je během algoritmu konstantní

Přirozené vyhledávání

- Postupně procházíme celý řetězec T a pro každou pozici testujeme, zda na ní nezačíná hledaný řetězec P



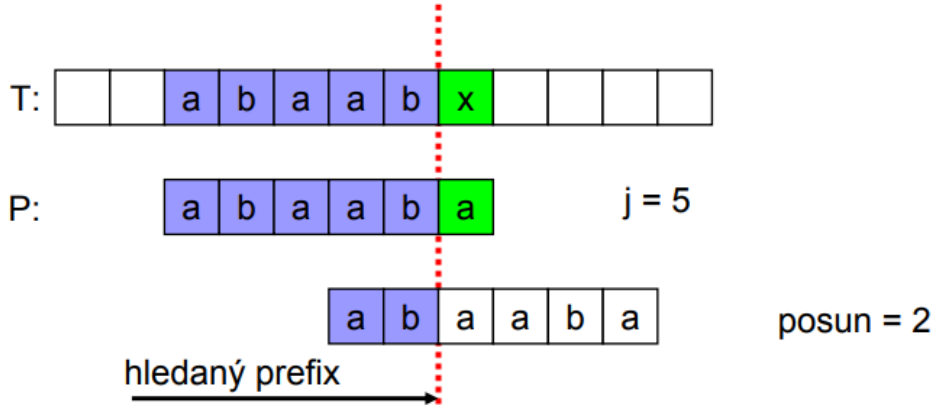
P se posouvá po 1 znaku přes T

- Algoritmus je rychlý, pokud máme velkou abecedu a naopak pomalý, pokud máme malou abecedu (například jen 0 a 1)
- Nejhorší případ:
    - T: 1111111111111111110
    - P: 110

Algoritmus KMP

- https://www.youtube.com/watch?v=pu2aO_3R118&ab_channel=GBhat
- Knuth-Morris-Pratt (KMP)
- Princip je stejný jako u přirozeného prohledávání
- Řízení procesu:
    - Nastupuje, pokud se vyskytne neshoda mezi textem a vzorem v pozici P[j]
    - Jaký je největším možný posun vzoru, abychom se vyhnuli opakovanému porovnávání

- Hledaný největší možný posun je roven délce největšího prefixu P[0:j-1], který je suffixem P[1:j-1]
- Příklad:



- Nalezneme největší prefix (start) :

  a b a a b" (P[0:j-1])
- a suffix (end) :

  "b a **a b**" (p[1:j-1])
- Hledaný řetězec : "a b"
- Nastavíme j = 2 // nová hodnota j

KMP preprocesing – chybová funkce

- Protože prefix i sufix hledáme ve vzoru, který známe předem, můžeme celou analýzu prefixů a sufixů provést předem
- Označme k jako pozici před neshodou
- Chybová funkce F(k) je definována jako nejdelší prefix P[0:k], který je také suffixem P[1:k]

Příklad chybové funkce

- P = abaaba
- F(k) velikost největšího prefixu, který je zároveň suffixem

P	a	b	a	a	b	a

k	0	1	2	3	4	5
F(k)	0	0	1	1	2	3

- Použití – určíme novou hodnotu posunu
- Př: pokud P[j] != T[i]), pak k = j-1 a

j = F(k).