

Vyhledávání v textu:

Σ := abeceda $\rightarrow$ množina všech znaků,
chceme její konečnost i konstantnost
dalších i rozumně malou, abychom mohli indexovat

Σ^* := řetězce $\rightarrow$ množina všech řetězců

$\alpha, \beta, \gamma \dots$ řetězce

$\alpha_p \dots$ zřetězení

$a, b, c \dots$ znaky

$\alpha[i]$... i -tý znak α malý

$|\alpha|$... délka řetězce

$\alpha[i:j]$... podřetězec (i tam je, j tam není)

ε ... prázdný řetězec

$\alpha[1:j]$... prefix

$\alpha[i:]$... suffix



Který podřetězec je prefixem suffixem.

Problém: dostaneme slovo σ , $S := |\sigma| \rightarrow$ což je vstupní řetězec
jehlu u , $J := |u| \rightarrow$ což je hledaný řetězec

Výstup: $\{i \mid \sigma[i:i+J] = u\}$

Triviální přístup:

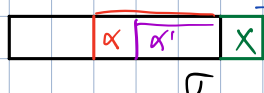
$\Theta(S \cdot J) \rightarrow$ zkoušíme každý začátek a porovnáváme s jehlou následující

Neefektivní: $\Theta(S)$, vstup: $uouuouos$, hledám $uouos$

Postupně probíráme, když nemáme shodu z hledaným, prostě zanedbáváme.
Zbytek mi jen uos , takže nic nemůžeme, přestože text $uouos$ obsahuje.

Incrementální algoritmy:

Postupně čteme vstup a podle toho dopočítáváme výstup. Pokud α není prefixem jehly, zanedbáváme α a



stav $\alpha :=$ nejdelší prefix jehly, který je suffixem sava

$\hookrightarrow$ jehly při čtení $uououubpbi$ $\rightarrow$ zde $uouo$
 $\rightarrow$ zde $uouos$
 $\rightarrow$ zde uou
 $\rightarrow$ zde uou (uouou už není prefix jehly)

α' tak dlouho, než α' je prefixem jehly

nový stav $\rightarrow \varepsilon$ (prázdný řetězec)

$\alpha'x$ suffix σx

$\hookrightarrow \alpha'$ suffix σ

$\hookrightarrow \alpha'x$ prefix jehly $\hookrightarrow \alpha'$ prefix jehly

$\} \} \text{ to bo plyne: } |\alpha'| \leq |\alpha|$

α' je suffixem α

Předpokládáme posunuti:

$\rightarrow \alpha' \neq \alpha$

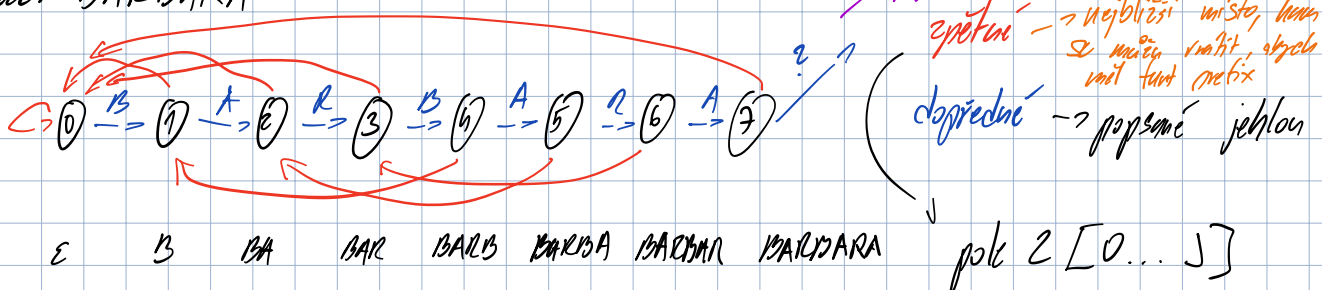
Pro slovo $\alpha := z(\alpha) :=$ nejdelší α' vlastní suffix α , který je prefixem jeho

KMP algoritmus (Knuth, Morris, Pratt):

lze popsat pomocí vyhledávacího automatu.

Řetězec: BARBARA

stav:



KMP:

Krok (i, x) :

Pokud $jehl[i] \neq x$:

Jeli $i = 0$: vrátíme 0

$i = z[i]$

Vrátíme $i+1$

Hledání jehly:

Rychlost:

Hleděj (σ)

1. $i = 0$

2. Pro $j = 0 \dots |\sigma| - 1$:

3. $i = \text{krok}(i, \sigma[j])$

4. Pokud $i = |jehl|$:

5. Ohlásím výsledek na $j - |jehl| + 1$.

Lemma: Hleděj (σ) má složitost $\Theta(|\sigma|)$.

Dů: Hleděj je zjevně lineární, pouze určit krok (i, x) může být problém.

dopředných hran je $\leq |\sigma|$

zpětných $\leq$ # dopředných $\leq |\sigma|$

Celkem tedy nejvíce $2|\sigma|$ hran projde.

Celý algoritmus je tedy $\Theta(|\sigma|)$



Pokud α je slovo a spustíme automat na vstup $\alpha[1:]$, vrátíme ve stavu $z(\alpha)$

- první chybějící nejdelší jinný suffix než celou α

- zpětný hromád se dají konstruovat až za chvilu $\rightarrow$ všechny předpoklady už budou vyhovovat.

Spustím na:

A
AR
ARB
ARBA
ARBAR
ARBARA

stav, nastit jen tohle

ARBARA

Konstrukce automatu:

- 1) $z[0] = 0, z[1] = 0$
- 2) $i = 0$
- 3) Pro $j = 2, \dots, J$:
- 4) $i = \text{Krok}(i, \text{jehlh}[j-1])$
- 5) $z[j] = i$

složitost hledj (j)

$$\theta(j) + \theta(s)$$

V: KMP najde všechny výskyty v čase $\theta(j+s)$.

Důležité nahore.

Obecněji:

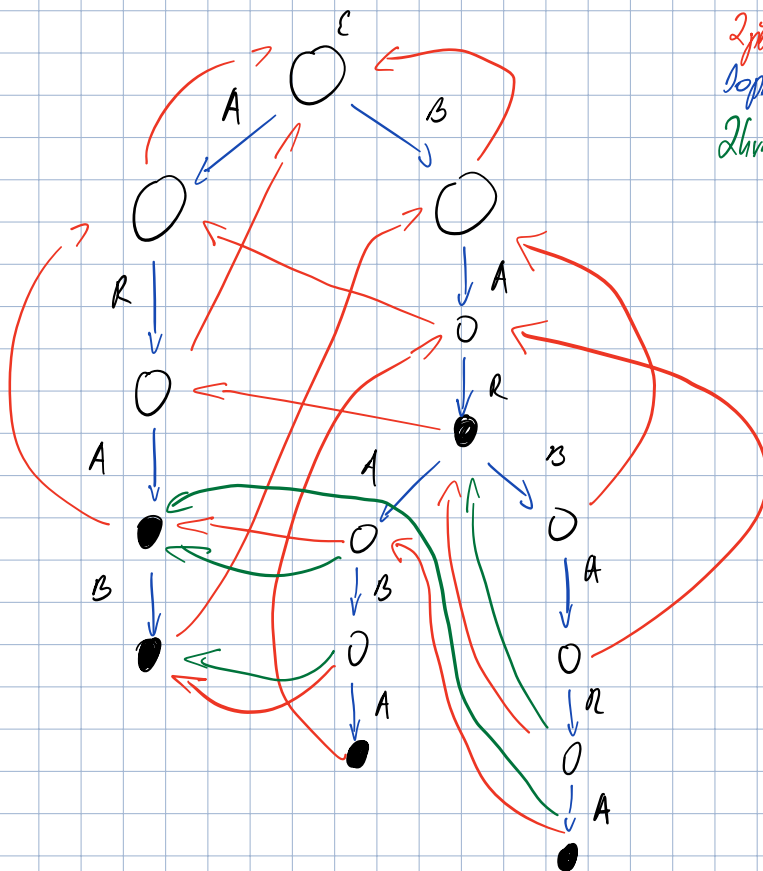
jehlh: $u_1 \dots u_n$
 délka: $J_1 \dots J_n$
 Sema θ délky S

$$\text{Cheme} = \{(i, j) \mid u_i = \sigma[j: j+|u_i|]\}$$

$$\text{čas } \theta(S + \sum J_i + V) \rightarrow \# \text{ výskytů}$$

jehlh:

ARA
 BAR
 ARAB
 BARBARA
 BARABA



zpětní
 dopřední
 zhratka

do nejbližšího
 konceho bodu
 po zpětných
 hranicích