MI-SP-SP-5 (MI-EVY) (přednášky 5 a 11)

Algoritmy přesného a přibližného vyhledávání.

Exact matching

forward matching

- ▷ naivní (brute force) algoritmus - $O(mn)$ time

- ▷ Karp-Rabin - hledání pomocí hashovací funkce - $O(mn)$ time
  - vypočítám hash vstupu (hledaného) - používá se rolling hash → jednoduše spočítám další element
  - potom udělám stejně se všemi substringy délky m
    - → můžu mít i false positives - musím ještě zkontrolovat

příklad: base = 5 mod = 10

$$p = aba \rightarrow 0 \cdot 5^2 + 1 \cdot 5^1 + 0 \cdot 5^0 = 5$$

další element spočítám jako $(prev - x \cdot 5^2) \cdot 5 + y \cdot 5^0$

- ▷ Morris-Pratt algorithm - $O(n)$ time, $O(m)$ space

$$\beta'[i] = \beta[i-1] + 1$$

  - ze předu matchuji symboly - v případě, že se netrefím, zarovnám na hodnotu $\beta'[i]$
  - hodnota říká kolikátý symbol má být po zarovnání pod současným

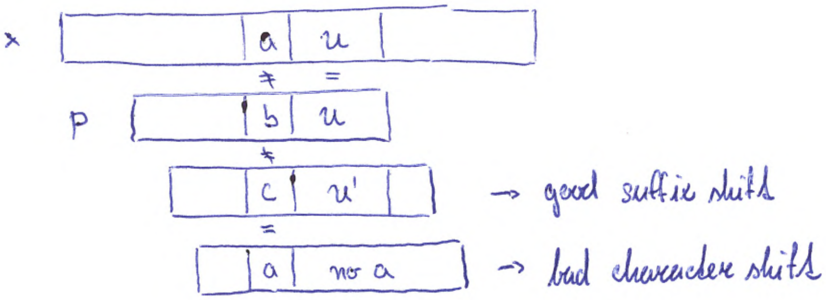
- ▷ Knuth-Morris-Pratt algoritmus
  - vylepšení o efektivnější zarovnávání

$\beta''[i]$ - KMP funkce

$\beta''[j] = j'$, $j'-1$ je délka nejdelšího borderu $p[1 \dots j-1]$ kde se $p[j'] \neq p[j]$

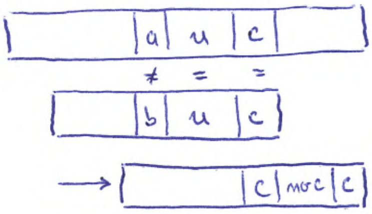
backward matching

- ▷ Boyer-Moore algorithm



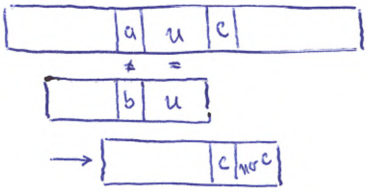
Boyer-Moore-Horsepool $O(nm)$

- BCS dle posledniho symbolu daneho patternu



Boyer-Moore-Sunday

- next symbol in text



[Zhu - Takaoka - používá dva symboly místo jednoho]

Approximate pattern Matching

- Edit distance - vzdálenost mezi dvěma stringy

○ Hamming - pouze operace replace

○ Levenshtein (edit distance) - replace, delete, insert

- dynamické programování

D	-	a	c	g	a	t
-	0	1	2	3	4	5
a	1	0	1	2	3	4
g	2	1	1	1	2	3
g	3	2	2	2	1	2
c	4	3	2	3	2	2
t	5	4	3	3	3	2

○ Weighted distance

○ Needleman - Wunsch algorithm - porovnávání sekvencí v bioinformatice

- ceny operací jsou dány tabulkou - různé záměny různě drahé

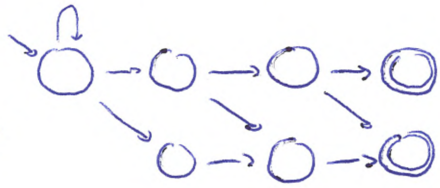
+ gap penalty - cena za mezeru (operace insert a delete)

- cena se maximalizuje

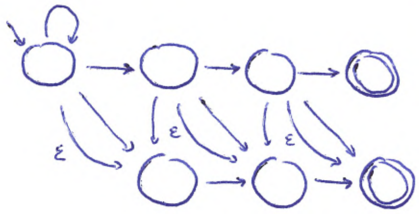
- approximate string matching - hledám slova v textu se vzdáleností menší než k

DP:

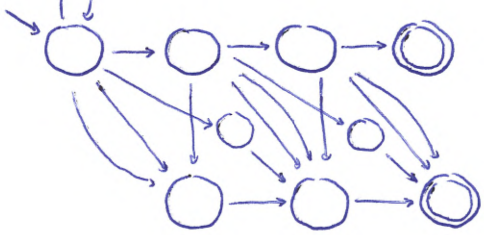
Hamming:



Levenshtein:



Damerau:



- navíc operace transpose

[+bit paralelism]

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
p[i]	a	b	a	a	b	a	b	a	a	b	a	a	b	a	
β[i]	0	0	1	1	2	3	2	3	4	5	6	4	5	6	
β'[i]	0	1	1	2	2	3	4	3	4	5	6	7	5	6	$\boxed{7}$
β''[i]	0	1	0	2	1	0	4	0	2	1	0	7	1	0	$\boxed{7}$

- KMP - koukam se na pozici $\beta'[i] = a$ → pokud $p[a] \neq p[i]$ → $\beta''[i] = a$

→ pokud $p[a] = p[i]$ → $\beta''[i] = \beta''[a]$

BMH:

p = abaabaa
6543210

c	BCS[c]
a	1 ← minimum je 1
b	2
c	7

```
a b c a b a ⓐ ⓑ a ⓐ ⓑ a a a
―――――――――――――――――――――――――――
    ≠ " " " "
a b a a b a a
              *
  a b a a b a a
      " " " " " " "
    √ a b a a b a a
                    *
        a b a a b a a
            " " " " " " "
          √ a b a a b a a
```