

# String matching

Profs. André, Salles

Departamento de Informática  
Universidade Federal de Viçosa

INF 333 - 2024/1

# String Matching

# String Matching

Dados dois strings  $A$  e  $B$ , encontre a primeira ocorrência de  $B$  em  $A$  (se existir).

Ou seja, encontre o menor  $k$  tal que, para todo  $i$ ,  $1 \leq i \leq |B|$ , temos  $a_{k+i} = b_i$ .

$A = \text{cdccdcddcdcdcddcdccdc}$

$$B = \text{cdcddcdc}cc$$

# String Matching - Força Bruta

## Força bruta

- Alinhar  $B$  com  $A$  na primeira posição
- Iniciar a comparação caractere por caractere
- A cada erro, mover  $B$  uma posição à frente e reiniciar a comparação
- Complexidade?

$A =$  c d c c d c d c d d c d c d c d d c d c d c c d c

$B =$  c d c d d c d c d c c

# String Matching - melhoria

## Pior caso

- $A = \text{dddddddddddc}$   
 $B = \text{ddddc}$

## Melhoria

- $A = \text{cdddcdddcdddddc}$   
 $B = \text{cdddd}$
- evitar comparar de novo o que já sabe!

# String Matching - KMP

$A = \text{cdccdcddcddcddcddcddcddcddc}$

$$B = \text{cdcddcdc}c$$

- Considera a falha em  $b_5 \neq a_8$  quando  $i = 3$ . O que pode ser melhorado?

$A = \text{cdc}\color{red}{\text{cd}}\text{cd}\text{ddc}\text{dc}\text{dcd}\text{ddc}\text{dc}\text{dccc}$

$$B = \text{cdcdd} \dots$$

- Suponha falha em  $b_{11} \neq a_{16}$  quando  $i = 5$ . O que pode ser melhorado?

$A = \text{cdccdc} \textcolor{red}{\text{cdcddcdc}} \textcolor{red}{\text{cdc}} \text{ddcdcdccdc}$

$$B = \text{cdcddcdc} \text{cc}$$

- Procurar o máximo sufixo de  $B[1..10]$  que é também prefixo de  $B$
- Independente de  $A$ ! Pode ser pré-processado conhecendo-se apenas  $B$ !

# String matching - pré-processamento

$B = \text{cdcdcdcdcdcc}$

- Para cada  $b_i$ , encontrar o maior prefixo de  $B[1..i-1]$  que é também sufixo
- Se o tamanho deste sufixo é  $j$ , comparar o mesmo caractere de  $A$  com  $b_{j+1}$   
(sem re-comparar os anteriores!)
- $\text{next}(i) = \text{máximo } j \text{ tal que } B[i-j..i-1] = B[1..j]$
- Quando ocorre uma falha em  $b_i$ , comparar novamente com  $b_{\text{next}[i]+1}$
- Caso especial: se a falha for em  $b_1$ , mover em  $A$

$i$	1	2	3	4	5	6	7	8	9	10	11
$B$	c	d	c	d	d	c	d	c	d	c	c
$\text{next}$	-1	0	0	1	2	0	1	2	3	4	3

- Complexidade?

# String matching - Boyer-Moore

Caracteres do string procurado verificados da direita para esquerda

- Regra do caractere errado (*bad character*)
- Regra do sufixo correto (*good suffix*)

A: HERE I S A SIMPLE EXAMPLE

B: EXAMPLE

A: HERE IS A SIM PLE EXAMPLE

B:           EXAMPLE

A: HERE IS A S I M P L E EXAMPLE

B:           EXAMPLE

A: HERE IS A SIMPLE EXA M P L E

B:                   EXAMPLE

A: HERE IS A SIMPLE E X A M P L E

B:                   EXAMPLE

# Aho–Corasick

- Permite busca simultânea por múltiplos strings

## Dicionário

- he
- she
- his
- hers

## Texto

- shers

## Dicionário

- a
- ab
- bab
- bc
- bca
- c
- caa

## Texto

- abccab

