Casamento de Padrões em Texto

Definição

- Cadeia de caracteres: sequência de elementos denominados caracteres.

- Os caracteres são escolhidos de um conjunto denominado alfabeto.
  - Ex.: em uma cadeia de bits o alfabeto é { 0, 1 }.

Definição

- Casamento de padrão: encontrar todas as ocorrências de um padrão em um texto

- Exemplos de aplicação:
  - edição de texto;
  - recuperação de informação;
  - estudo de sequências de DNA em biologia computacional.

Definição

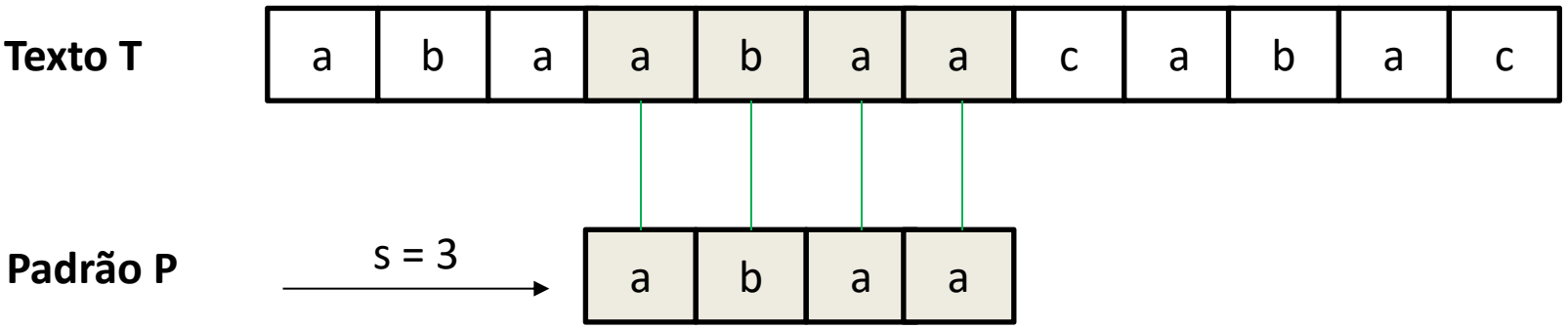
- Em um editor de texto, o usuário pode estar interessado em buscar todas as ocorrências de um padrão (uma palavra específica) no texto que está sendo editado.

- Esse problema é conhecido como casamento de cadeia de caracteres ou casamento de padrão (pattern matching).

Definição formal - notação

- **Texto**: arranjo T[1..n] de tamanho n;
- **Padrão**: arranjo P[1..m] de tamanho m ≤ n.
- Os elementos de **P** e **T** são escolhidos de um alfabeto finito **Σ** de tamanho **c**.
  - Ex.: Σ = { 0, 1 } ou Σ = {a, b, . . . , z }
- **Casamento de cadeias ou casamento de padrão**: dados duas cadeias *P* (padrão) de comprimento $|P| = m$ e *T* (texto) de comprimento $|T| = n$, onde $n > m$, deseja-se saber as ocorrências de P em T.

Exemplo



Queremos encontrar toda acorrência do padrão P = *abaa* no texto
T = *abaabaacabac*
O padrão ocorre apenas uma vez no texto, no deslocamento s = 3

Categorias de Algoritmos

- **P e T não são pré-processados:**
  - algoritmo sequencial, on-line e de tempo-real;
  - padrão e texto não são conhecidos *a priori*.
  - complexidade de tempo O (mn) e de espaço O(1), para pior caso.

Categorias de Algoritmos

- **P pré-processado:**
  - algoritmo sequencial;
  - padrão conhecido anteriormente permitindo seu pré-processamento.
  - complexidade de tempo O (n) e de espaço O (m + c), no pior caso.
  - ex.: programas para edição de textos.

Categorias de Algoritmos

- **P e T são pré-processados:**
  - algoritmo constrói índice.
  - complexidade de tempo O(log n) e de espaço é O (n).
  - tempo para obter o índice é grande, O (n) ou O (n log n).
  - compensado por muitas operações de pesquisa no texto.
  - Tipos de índices mais conhecidos:
    - Arquivos invertidos
    - Árvores trie e árvores Patricia
    - Arranjos de sufixos

Exemplos: P e T são pré-processados

- Diversos tipos de índices: arquivos invertidos, árvores trie e Patricia, e arranjos de sufixos.
- Um arquivo invertido possui duas partes: vocabulário e ocorrências.
- O vocabulário é o conjunto de todas as palavras distintas no texto.
- Para cada palavra distinta, uma lista de posições onde ela ocorre no texto é armazenada.
- O conjunto das listas é chamado de ocorrências.
- As posições podem referir-se a palavras ou caracteres.

Exemplo – arquivo invertido

```
1      7          16     22  26          36        45       53
Texto exemplo. Texto tem palavras. Palavras exercem fascínio.
```

exemplo	7
exercem	45
fascínio	53
palavras	26 36
tem	22
texto	1 16

Casamento exato

- Consiste em obter todas as ocorrências exatas do padrão no texto.
  - Ex.: ocorrência exata do padrão ***teste***.

    os testes testam estes alunos
- Os algoritmos para o casamento exato, em geral, funcionam em dois enfoques:

Casamento exato

1 - Consiste em ler os caracteres do texto um a um e, a cada passo, algumas variáveis são atualizadas de forma a identificar uma ocorrência possível (algoritmos força bruta, Knuth-Morris-Pratt e Shift-And).

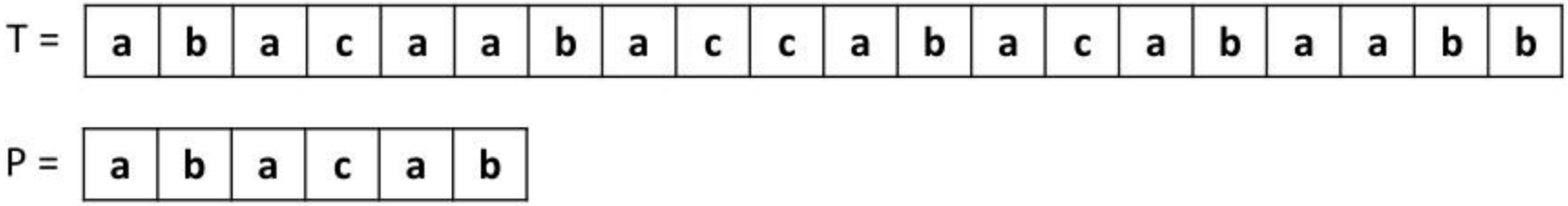
2 - Pesquisa de P em uma janela que desliza ao longo de T, pesquisando por um sufixo da janela que casa com um sufixo de P, por comparações da direita para a esquerda (algoritmos Boyer-Moore-Horspool e Boyer-Moore).

Força Bruta / Ingênuo

- É o algoritmo mais simples para casamento de cadeias.

- A ideia é tentar casar qualquer subcadeia no texto de comprimento m com o padrão.

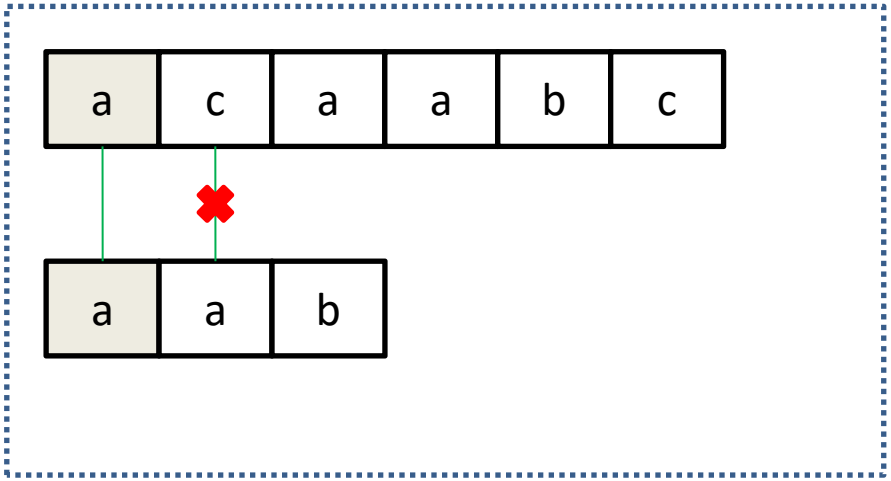
Força Bruta

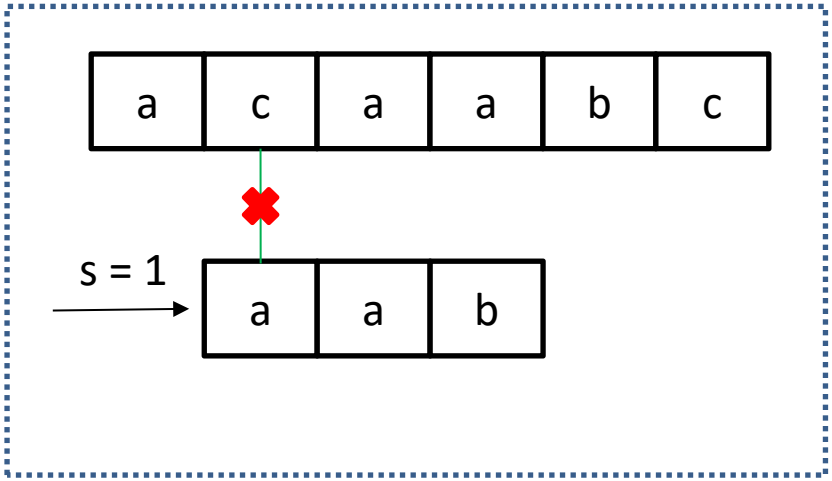


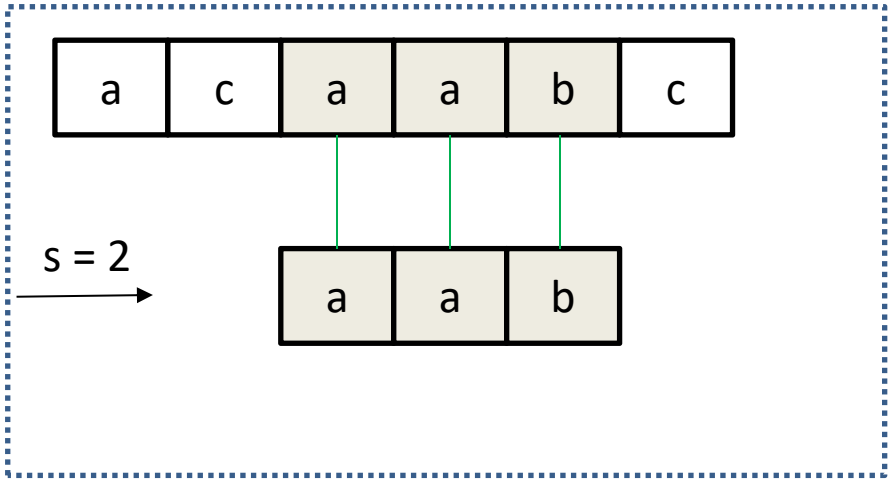
Entrada: as cadeias T (texto) com n caracteres e P (padrão) com m caracteres.

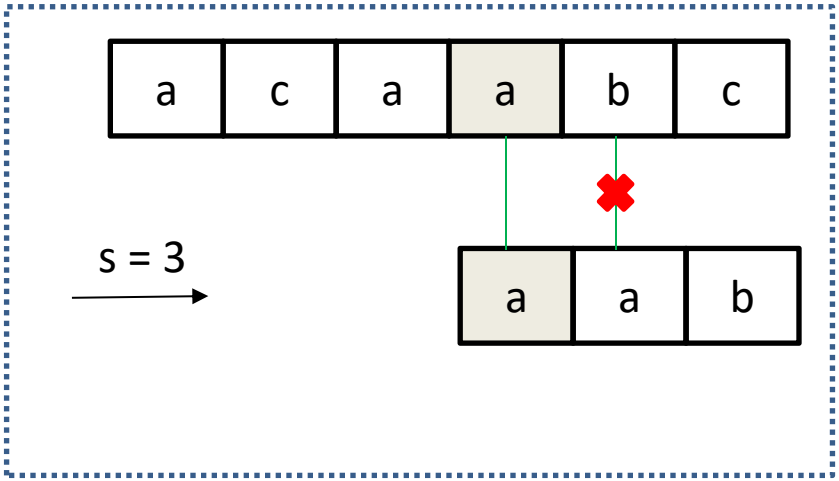
Saída: o índice da primeira substring de T igual a P, ou uma indicação de que P não é uma substring de T.

Exemplo









Força Bruta

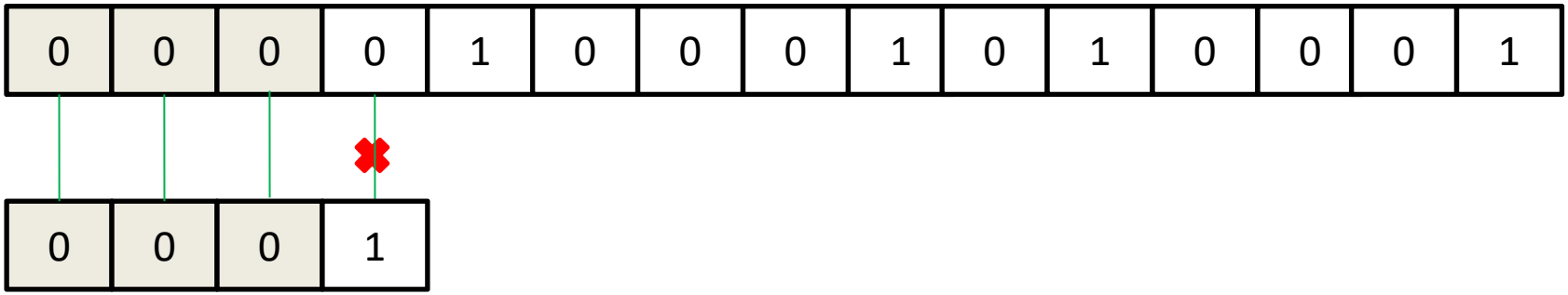
1. busca_padrao(t, n, p, m)
2. para **i** ← **0** até **n - m** faça
3. j ← 0
4. enquanto (**j < m**) e (**t[i+j] == p[j]**) faça
5. j ← j + 1
6. se **j == m** então
7. retorne i
8. retorne “padrão p não encontrado em t”

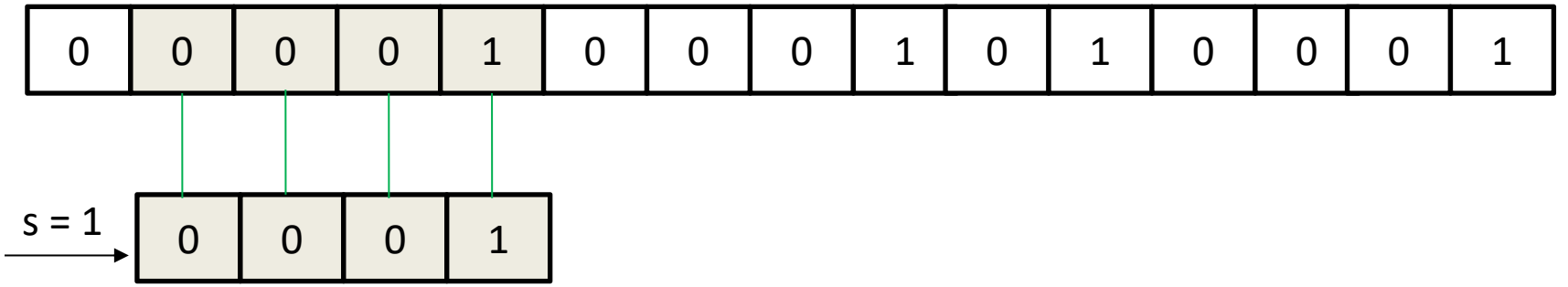
Algoritmo

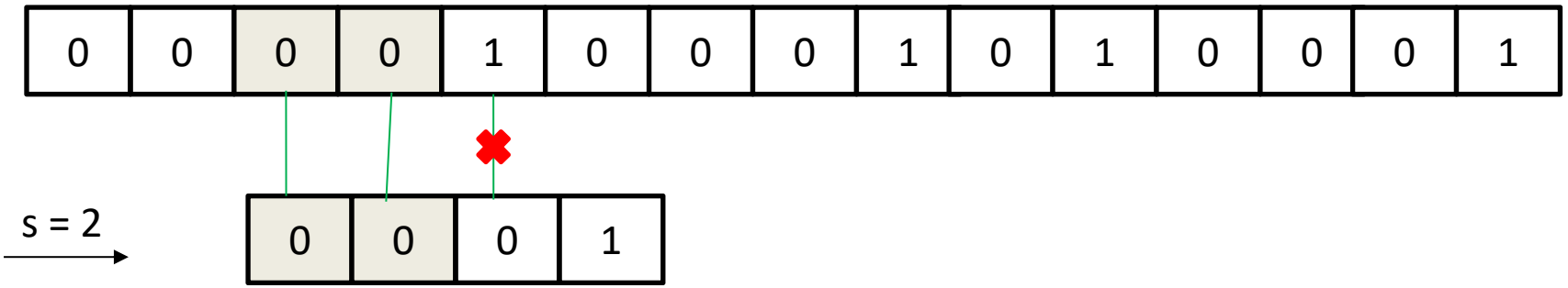
```
Naive-string-matcher(T,P)
    n = T.comprimento
    m = P.comprimento
    for s = 0 to n-m
        if P[1..m] == T[s+1 .. s +m]
        imprimir “Padrão ocorre no deslocamento” s
```

Exemplo

Quantas comparações?







TEM COMO MELHORAR O ALGORITMO DE FORÇA BRUTA?

Algoritmo de Boyer Moore

- Assim como o força-bruta, esse algoritmo consiste na comparação de caracteres.

- A diferença está no salto de posições do arrnajo que o algoritmo faz.
  - gera um número menor de comparações
  - realiza um pré-processamento para determinar o tamanho dos saltos

Algoritmo de Boyer Moore

- Pré-processamento
  - Define-se o alfabeto
  - Todas as posições da tabela do alfabeto são carregadas com o tamanho da palavra
  - Os caracteres alterados recebem o valor m-i:
    - m é o tamanho da palavra
    - i é a posição do caracter na palavra
    - faz-se o processo até o penúltimo caracter da palavra

Algoritmo de Boyer Moore

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
valor																							

T	E	S	T	E

i

m = 5

Algoritmo de Boyer Moore

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
valor	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5

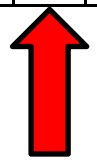
T	E	S	T	E

i

m = 5

Algoritmo de Boyer Moore

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
valor	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5



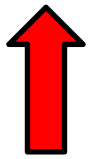
T	E	S	T	E

i

m = 5
i = 1
valor = 5 - 1 = 4

Algoritmo de Boyer Moore

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
valor	5	5	5	5	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5



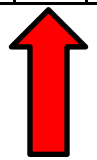
T	E	S	T	E

i

m = 5
i = 2
valor = 5 - 2 = 3

Algoritmo de Boyer Moore

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
valor	5	5	5	5	3	5	5	5	5	5	5	5	5	5	5	5	5	5	2	4	5	5	5



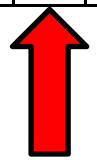
T	E	S	T	E

i

m = 5
i = 3
valor = 5 - 3 = 2

Algoritmo de Boyer Moore

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
valor	5	5	5	5	3	5	5	5	5	5	5	5	5	5	5	5	5	5	2	1	5	5	5



T	E	S	T	E

i

m = 5
i = 4
valor = 5 - 4 = 1

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres

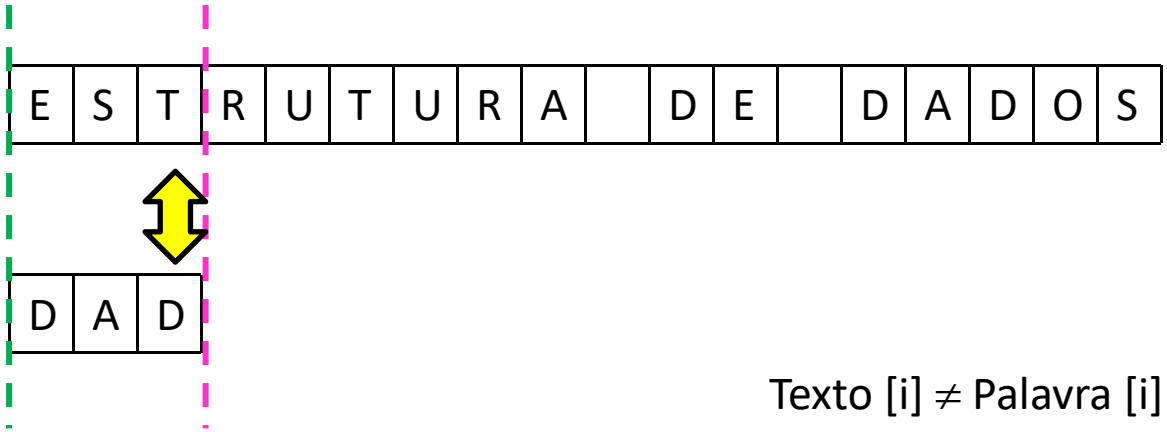
E	S	T	R	U	T	U	R	A		D	E		D	A	D	O	S

D	A	D

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres



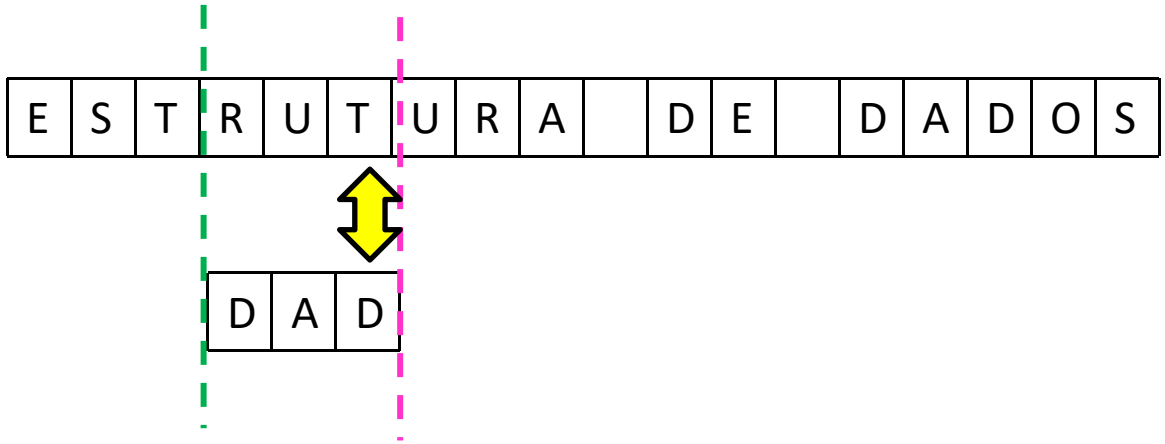
Texto [i] ≠ Palavra [i]

Salto de texto[i]

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres



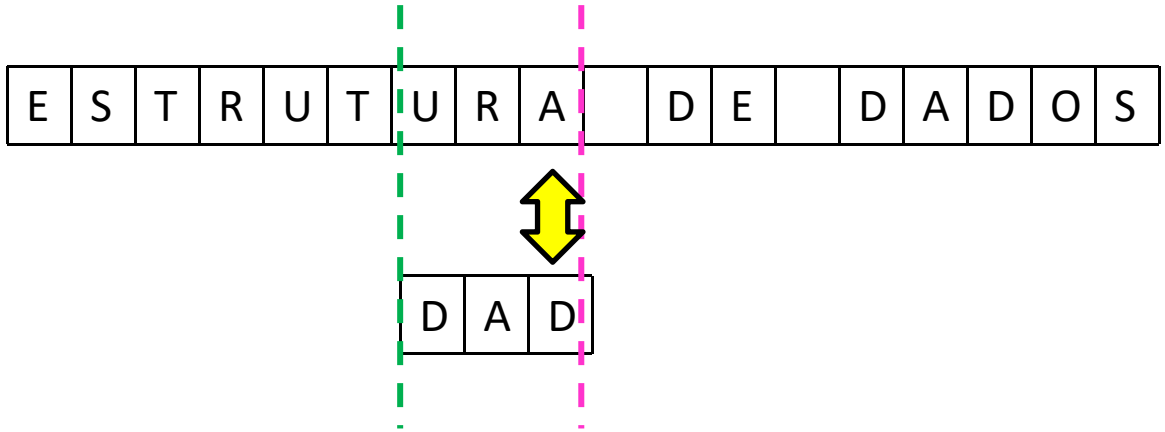
Texto [i] ≠ Palavra [i]

Salto de texto[i]

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres



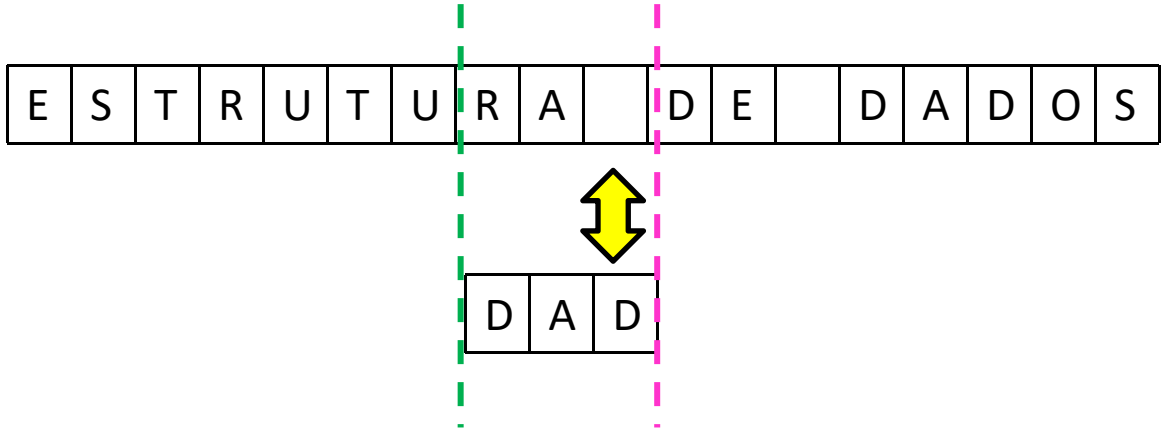
Texto [i] ≠ Palavra [i]

Salto de texto[i]

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres



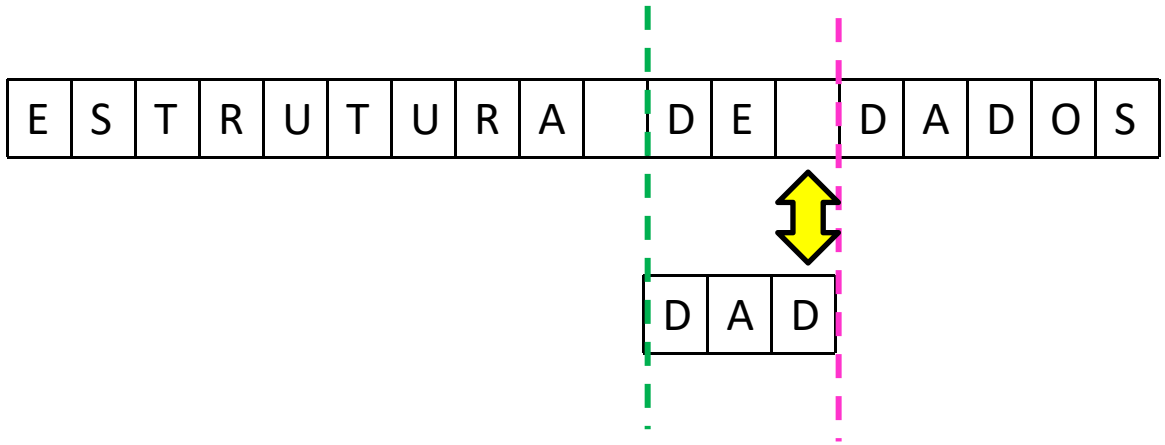
Texto [i] ≠ Palavra [i]

Salto de texto[i]

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres



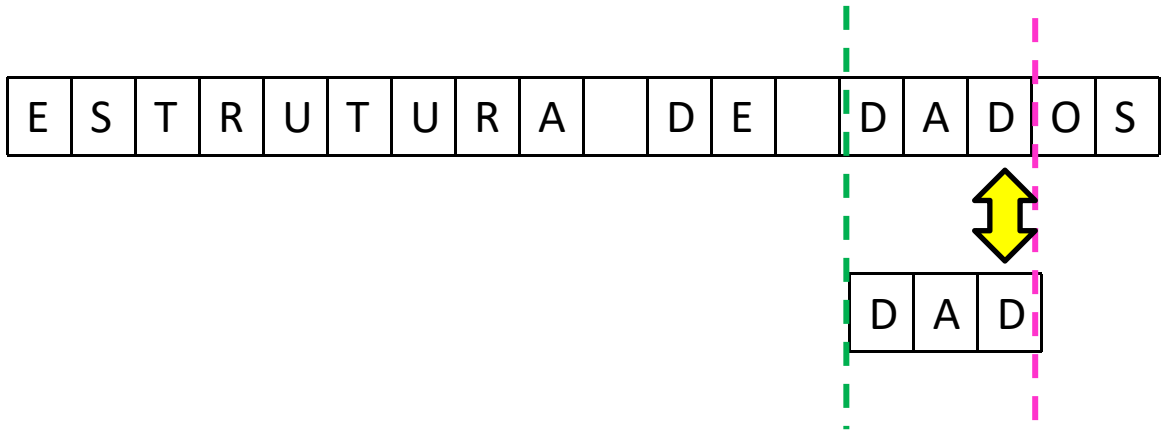
Texto [i] ≠ Palavra [i]

Salto de texto[i]

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres

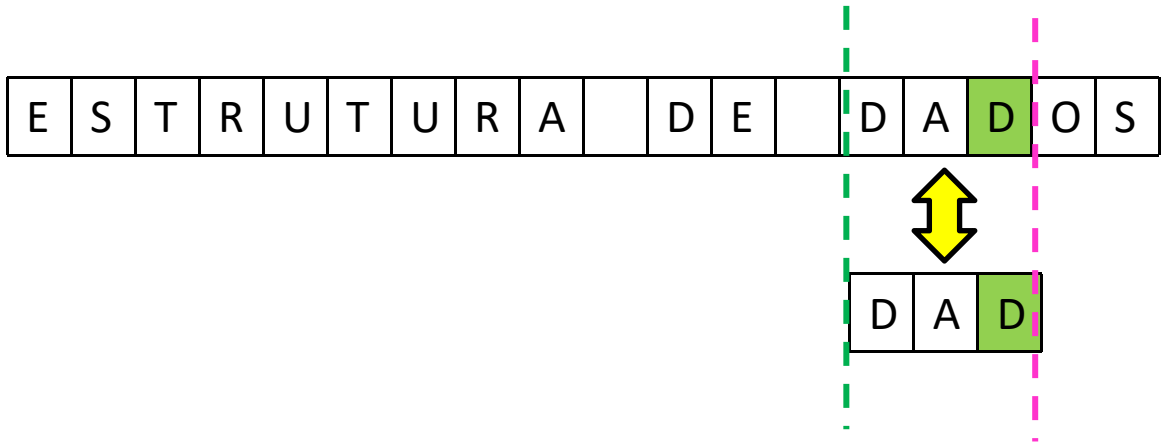


Texto [i] = Palavra [i]

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres

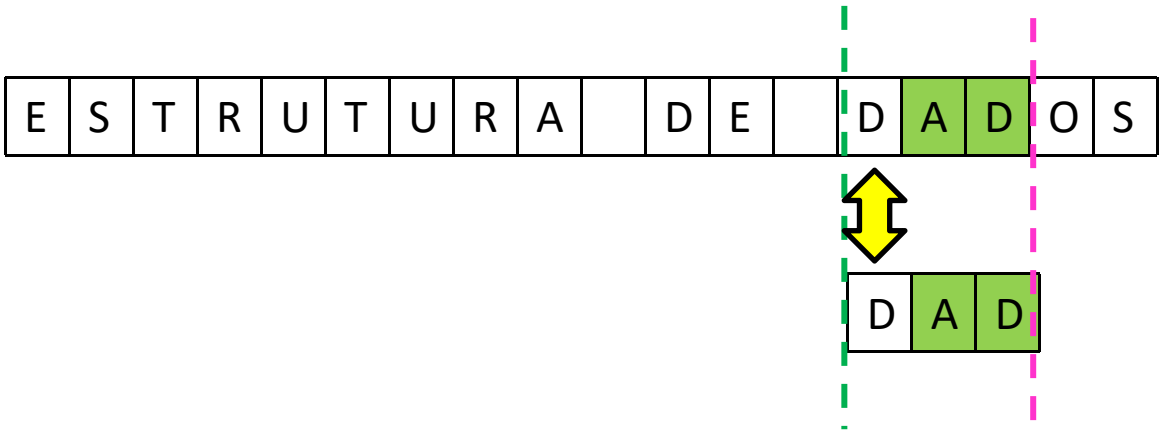


Texto [i] = Palavra [i]

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres

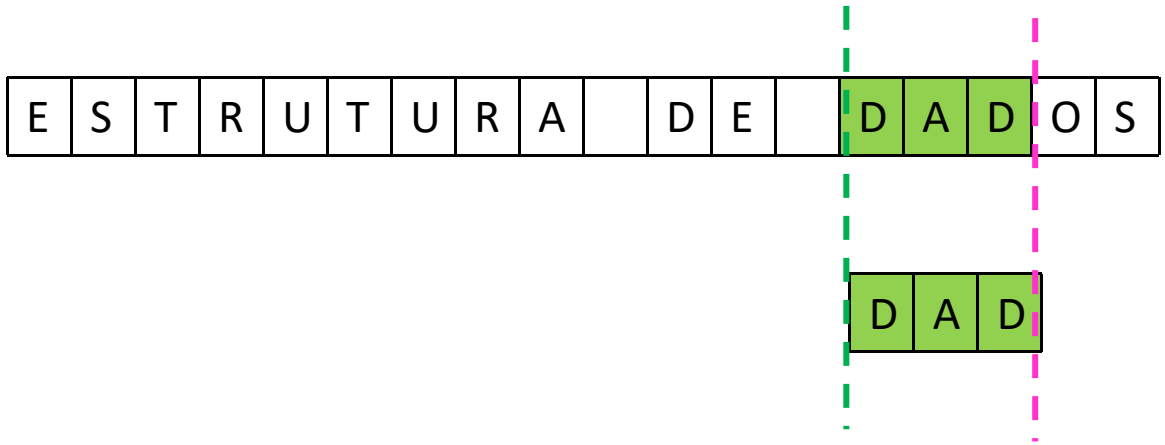


Texto [i] = Palavra [i]

EXEMPLO

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	*
valor	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

* Outros caracteres



Exercícios

- Considere o texto abaixo:

ABACAABACCABACABAABB

Pesquise o padrão ABACAB utilizando o método força bruta e depois o método BM.Diga quantas comparações cada método faz.

Exercícios

- Considerando a seguinte cadeia de DNA:

  T = “CTAATCGCTTAATCAAACGC”

- Realize o passo a passo do algoritmo Boyer-Moore para verificar se o padrão P = “ATCA” ocorre em T.

Construa o vetor L e ilustre todos os alinhamentos e comparações dos caracteres do padrão P em T.

Algoritmo Boyer Moore

```
int boyerMoore(char palavra[], int m, char texto[], int n)
{
        int Alfabeto[256], i, pos, r, ocorrencias = 0;
        //Pré processamento
        for (i = 0; i < 256; i++)Alfabeto[i] = m;
        for (i = 1; i < m; i++) Alfabeto[palavra[i-1]] = m - i;

        //Busca por ocorrencias da palavra
        pos = m;
        while (pos <= n)
        {
                r = 1;
                while (r <= n && palavra[m - r] == texto[pos - r]) r++;

                if (pos == n) pos++; //chegou ao final do texto

                if (r > m) ocorrencias++; //encontrou uma palavra
                else pos += Alfabeto[text[pos-r]];//avança casas
        }
        return ocorrencias;
}
```

Material de apoio

- Livros:
  - CORMEN T. H. et al., **Algoritmos: Teoria e Prática**. Tradução da 2ª ed., Rio de Janeiro: Campus, 2002.
  - SEDGEWICK, Robert; WAYNE, Kevin. **Algorithms**. Addison-Wesley Professional, 2011.
- Websites:
  - https://www.ime.usp.br/~pf/estruturas-de-dados/aulas/kmp.html
  - https://pt.wikibooks.org/wiki/Algoritmos/Reconhecimento_de_padr%C3%B5es/Algoritmo_KMP
  - https://homepages.dcc.ufmg.br/~nivio/cursos/pa02/tp3/tp32/tp32.html
  - https://www.youtube.com/watch?v=4Xyhb72LCX4
  - https://www.youtube.com/watch?v=JITD8C2wLQY