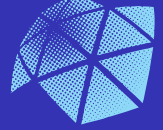


# Laboratório de Algoritmos e Técnicas de Programação II

Aula 05 - Manipulação de strings e reconhecimento de padrões

Álvaro Magri Nogueira da Cruz

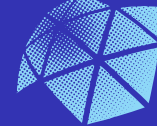


- 1 Introdução
- 2 Exemplos de funções da `string.h`
- 3 Algoritmos de casamento de padrões
  - Força-bruta ou ingênuo
- 4 Exercícios
- 5 Referências



## Observações

- A biblioteca de tratamento de strings é a `<string.h>`;
- Possui funções de:
  - Cópia;
  - Concatenação;
  - Determinação de comprimento;
  - Comparação.
- Com ela podemos manipular strings rapidamente, minimizando linhas de código!



## Algumas funções

Protótipo da função	Descrição da função
<code>char *strcpy(char *s1, const char *s2)</code>	Copia string s2 no array s1. O valor de s1 é retornado
<code>char *strncpy(char *s1, const char *s2, size_t n)</code>	Copia no máximo n caracteres da string s2 no array s1. O valor de s1 é retornado.
<code>char *strcat(char *s1, const char *s2)</code>	Acrescenta a string s2 ao array s1 . O primeiro caractere de s2 sobrescreve o caractere nulo de finalização de s1 . O valor de s1 é retornado.
<code>char *strncat(char *s1, const char *s2, size_t n)</code>	Acrescenta no máximo n caracteres da string s2 ao array s1 . O primeiro caractere de s2 sobrescreve o caractere nulo de finalização de s1 . O valor de s1 é retornado.
<code>int strcmp(const char *s1, const char *s2)</code>	Compara a string s1 com a string s2. A função retorna 0, menor do que 0 ou maior do que 0 se s1 for igual, menor ou maior do que s2, respectivamente
<code>int strcmp(const char *s1, const char *s2, size_t n)</code>	Compara até n caracteres da string s1 com a string s2. A função retorna 0, menor do que 0 ou maior do que 0 se s1 for igual, menor ou maior do que s2, respectivamente

# Exemplo: Copiando valores



```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char x[] = "Laboratorio de Prog II";
6     char y[25];
7     char z[25];
8     strcpy(y, x);
9     strncpy(z, x, 11);
10    z[11] = '\0';
11    printf("A string x eh: %s\nA string y eh: %s\nA string z
12          eh: %s\n", x, y, z);
13    return 0;
}
```

# Exemplo: Concatenando valores



```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char s1[30] = "Laboratorio de Prog II";
6     char s2[] = " - 2020";
7     char s3[40] = "Aula de ";
8     printf("s1: %s\ns2: %s\n\n", s1, s2);
9     strcat(s1, s2);
10    printf("strcat(s1,s2): %s\n", s1);
11    strncat(s3, s1, 11);
12    printf("strcat(s3,s1): %s\n", s3);
13    return 0;
14 }
```

# Exemplo: Comparando valores



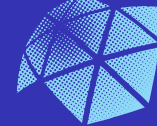
```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char *s1 = "Laboratorio de Prog II";
6     char *s2 = "Laboratorio de Prog II";
7     char *s3 = "Laboratorio de Prog I";
8     printf("s1: %s\ns2: %s\ns3: %s\n\n", s1, s2, s3);
9     printf("strcmp(s1, s2): %d\nstrcmp(s1, s3): %d\nstrcmp(s3, s2): %d\n", strcmp(s1, s2), strcmp(s1, s3), strcmp(s3, s2));
10    printf("\nstrncmp(s1, s2, 8): %d\nstrncmp(s1, s3, 22): %d\nstrncmp(s3, s2, 22): %d\n", strncmp(s1, s2, 8), strncmp(s1, s3, 22), strncmp(s3, s2, 22));
11    return 0;
12 }
```

# Exemplo: Comprimento de strings



```
1 #include <stdio.h>
2 #include <string.h>
3
4 int main(){
5     char *s1 = "Laboratorio de Prog II";
6     char *s2 = "Algoritmos e Tecnicas de Prog II";
7     char *s3 = "UNESP - Rio Preto";
8     printf("s1: '%s' -> tamanho %d\ns2: '%s' -> tamanho %d\n
9     ns3: '%s' -> tamanho %d\n", s1, strlen(s1), s2, strlen(s2
10    ), s3, strlen(s3));
11     return 0;
12 }
```

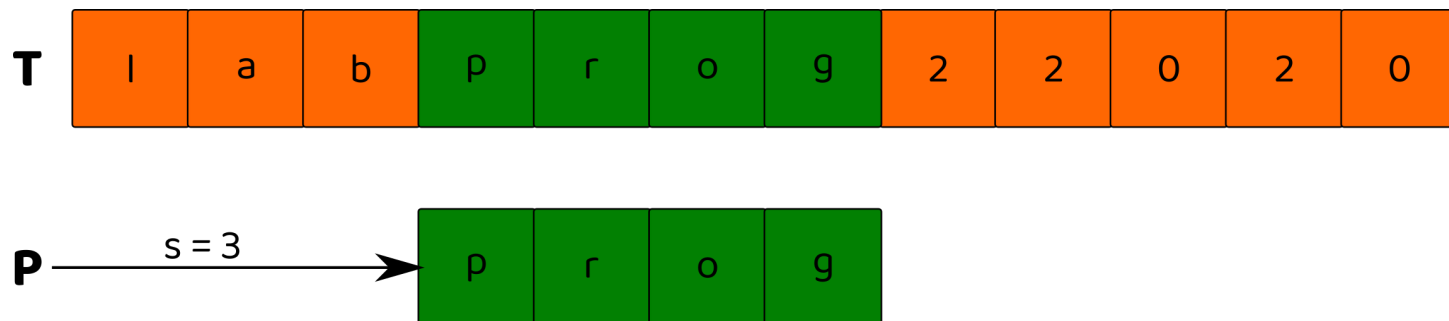
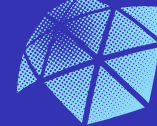




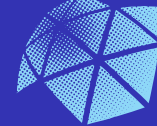
## O problema

- Casamento de padrões é um aspecto importante no contexto de processamento de textos;
- É alvo de estudo na Ciência da Computação há décadas;
- Problema que consiste em determinar a(s) ocorrência(s) de um determinado string (padrão) em um texto.
  - Dado um **texto**  $T[1 \dots n]$ ,  $T[i] \in \Sigma$ ,
  - e um **padrão**  $P[1 \dots m]$ ,  $p[i] \in \Sigma$
  - achar todos os deslocamentos válidos de  $P$  em  $T$ , onde deslocamento válido ( $s$ ) significa:
    - $T[s + 1 \dots s + m] = P[1 \dots m]$ .

# Algoritmos de casamento de padrões II



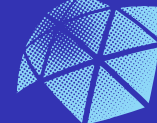
- Janela deslizante;
- Uma janela de tamanho  $m$  percorre o texto de tamanho  $n$ ;
- Continua-se até que  $m$  deslize por todo texto.



## Algoritmos de casamento de padrões

- Força-bruta ou ingênuo;
- KMP (Knuth-Morris-Pratt);
- Boyer-Moore.

# Força-bruta ou ingênuo I



- Buscar a palavra **ABRA** no texto **ABACADABRAC**.

i	j	i+j	0	1	2	3	4	5	6	7	8	9	10
txt →			A	B	A	C	A	D	A	B	R	A	C
0	2	2	A	B	R	A	← pat						
1	0	1		A	B	R	A						
2	1	3			A	B	R	A					
3	0	3				A	B	R	A				
4	1	5					A	B	R	A			
5	0	5						A	B	R	A		
6	4	10							A	B	R	A	

entries in black match the text

entries in red are mismatches

entries in gray are for reference only

return i when j is M

match

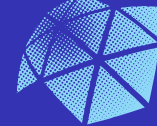
# Força-bruta ou ingênuo II



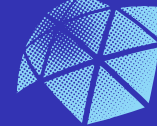
```
1 int forcaBruta(char *padrao, char *texto) {  
2     int j, M = strlen(padrao);  
3     int i, N = strlen(texto);  
4     for (i = 0; i <= N - M; i++) {  
5         for (j = 0; j < M; j++)  
6             if (padrao[j] != texto[i+j])  
7                 break;  
8         if (j == M) return i;  
9     }  
10    return -1;  
11 }
```



- 1 Implemente o algoritmo KMP;
- 2 Implemente o algoritmo Boyer-Moore;
- 3 Crie um arquivo de texto e faça uma função para efetuar a busca do padrão que o usuário desejar. Imagine como se você abrisse um editor de texto e fosse utilizar a ferramenta localizar, isto é, implemente a ferramenta localizar.



- ❶ Deitel H. M., e Deitel P.J; **“C: Como programar”**. 6.ed. **Pearson Prentice Hall, 2011. 818p.**
- ❷ Jean Paul Tremblay & Richard P. Bunt. **“Ciência dos Computadores - Uma abordagem algorítmica”**. McGraw-Hill.
- ❸ Jaime Evaristo. **“Aprendendo a Programar / Programando em Turbo Pascal”**. Edufal - Editora da Univ. Federal de Alagoas. Maceió, 1996.
- ❹ Harry Farrer et al. **“Pascal Estruturado (da série “Programação Estruturada de Computadores” )”**. Editora Guanabara Dois. Rio de Janeiro, 1985.
- ❺ Stephen O’Brien. **“Turbo Pascal 6 Completo e Total”**. Makron Books.



- ⑥ Celes, W., Cerqueira, R., Rangel, J.L. “Introdução a Estrutura de Dados”. Elsevier, 2004.
- ⑦ Feofiloff, P. “Algoritmos em Linguagem C”. Elsevier, 2009. 208p.
- ⑧ Schildt, H. “C Completo e Total”. 3<sup>a</sup> ed. Pearson. 1996. 852p.