Algoritmos básicos em Strings

Técnicas de Programação - 2025/02

Última atualização: 18/08/2025 09:04

Instruções

Vamos agora tentar **escrever** pseudo código. Cada exercício abaixo tem espaço para escrever e algumas entradas de teste. A ideia é só partir pra implementação **depois** de ter testado com sucesso para as entradas do material impresso.
Não se esqueça de consultar o cheat sheet para ver como traduzir o pseudo código para Java e os slides do dia 2 para ver quais operações são possíveis.

Prefixo

Dada uma string S, verifique se a string Q é um prefixo de S. Ou seja, se os primeiros $Q.length$ caracteres de S são exatamente iguais à string Q.

Algorithm 1: Prefixo

Input : string S, string Q
Output: bool

Entradas de teste:

- $S =$ "aa", $Q =$ "a"
- $S =$ "abcaas sdfsd a", Q="abc"
- $S =$ "a", $Q =$ "abc"
- $S =$ "aabc", $Q =$ "abc"

Sufixo

Dada uma string S, verifique se a string Q é um sufixo de S. Ou seja, se os últimos $Q.length$ caracteres de S são exatamente iguais à string Q.

Algorithm 2: Sufixo

Input : string S, string Q
Output: bool

Entradas de teste:

- $S =$ "aa", $Q =$ "a"
- $S =$ "abcaas sdfsd a", Q="abc"
- $S =$ "a", $Q =$ "abc"
- $S =$ "aabc", $Q =$ "abc"

Busca de termo 1

Dada uma string S, a string Q está contida em S em qual posição? Devolva a **primeira** posição em que isso acontece. Se a string Q não estiver contida em S devolva -1. Em Java este método se chama `indexOf`.

Algorithm 3: Busca

Input : string S, string Q
Output: int

Dica

O algoritmo *Prefixo* pode te dar inspiração para este exercício. Você não irá chamá-lo diretamente, mas ele pode fazer parte da *Busca*

Entradas de teste:

- $S =$ "aa", $Q =$ "a"
- $S =$ "saaabcaas sdfsd a", Q="abc"
- $S =$ "a", $Q =$ "abc"
- $S =$ "aabcbca abc bda", $Q =$ "abc"

Busca de termo 2

Dada uma string S, a string Q está contida em S em qual posição? Devolva a **última** posição em que isso acontece. Se a string Q não estiver contida em S devolva -1. Em Java este método se chama `lastIndexOf`.

Algorithm 4: BuscaReversa

Input : string S, string Q
Output: int

Dica

O algoritmo *Sufixo* pode te dar inspiração para este exercício. Você não irá chamá-lo diretamente, mas ele pode fazer parte da *Busca*

Entradas de teste:

- $S =$ "aa", $Q =$ "a"
- $S =$ "saaabcaas sdfsd a", Q="abc"
- $S =$ "a", $Q =$ "abc"
- $S =$ "aabcbca abc bda", $Q =$ "abc"

Substituição

Dada uma string S e duas string Q e Nq, devolva uma nova string $S2$ em que a primeira ocorrência de Q é substituída por Nq. Em Java este método é chamdado de `replace`.

Algorithm 5: Substitui

Input : string S, string Q, string Nq
Output: string

Dica

Este exercício é parecido com a busca, mas como agora temos que criar uma nova string precisamos usar a operação de concatenação. Você pode, inclusive, chamar o algoritmo *Busca* em seu pseudo código.

- $s_1 + s_2$ cria uma nova string com as duas strings anteriores concatenadas
- $s_1 =$ "" cria uma string vazia (que você pode usar para ir concatenando o resto!)

Entradas de teste:

- $S =$ "abc bla bla", $Q =$ "a", $Nq =$ "b"
- $S =$ "abc bla bla", $Q =$ "bla", $Nq =$ "bu"
- $S =$ "abc bla bla", $Q =$ "a", $Nq =$ "bca"

Substituir todos

Dada uma string S e duas string Q e Nq, devolva uma nova string $S2$ em que a **todas** ocorrência de Q são substituídas por Nq. Em Java este método é chamdado de `replaceAll`.

Algorithm 6: SubstituiTodos

Input : string S, string Q, string Nq
Output: string

> Aviso
>
> Este é um exercício **difícil**. As entradas abaixo são mais fáceis que as dos testes de unidade e vão ajudá-los a entender problemas comuns no algoritmo de vocês.

Entradas de teste:

- $S =$ "abc bla bla", $Q =$ "a", $Nq =$ "b"
- $S =$ "abc bla bla", $Q =$ "bla", $Nq =$ "bu"
- $S =$ "abc bla bla", $Q =$ "a", $Nq =$ "bca"