

MAC0121 Algoritmos e Estruturas de Dados I

[Painel](#) / [Meus cursos](#) / [2º semestre 2017](#) / [mac121.2017ii](#) / [Geral](#)
/ [Segmento comum mais longo e subpalíndromo mais longo](#)

Segmento comum mais longo e subpalíndromo mais longo

Neste exercício, você terá de escrever dois programas, `LongestCommonSegment.java` e `LongestPalindrome.java` (e mais uma classe auxiliar `SuffixArray2.java`; veja abaixo).

(I) Segmento comum mais longo

Sejam dados dois strings `s` e `t`. Dando `s` e `t` como entrada ao seu programa `LongestCommonSegment.java`, seu programa deve ter como saída um string `u` que ocorre como segmento tanto em `s` como em `t` e, adicionalmente, dentre as várias possibilidades, o string `u` deve ter o maior comprimento possível.

Exemplo. Suponha que `s` e `t` sejam os strings

```
bccbaccccbaacaaa  
baccabbacbbbabcb
```

A saída `u` de seu programa poderia então ser

```
bacc
```

Seu programa deve ter dois modos de execução. Em um dos modos (especificado por não haver argumentos de linha de comando), os dois strings `s` e `t` devem ser dados na entrada padrão, um por linha. No segundo modo de execução, seu programa deve receber o nome de dois arquivos na linha de comando, arquivos esses que contém os strings de entrada. Por exemplo, considere os arquivos `s.in`, `t.in` e `st.in` anexos. Seu programa poderia então ser executado das seguintes duas formas:

```
$ java-introcs LongestCommonSegment < st.in  
bacc  
$ java-introcs LongestCommonSegment s.in t.in  
bacc  
$
```

Você pode experimentar seu programa usando sequências aleatórias geradas pelo programa `Generator.java`:

```
$ java-introcs Generator 2 1000000 abc | java-introcs LongestCommonSegment
bcbaaabccbcbaabbcaabbccbcbb
$ java-introcs Generator 2 3000000 abc | java-introcs LongestCommonSegment
babacaccccbacbcacbccbcabbc
$ java-introcs Generator 2 9000000 abc | java-introcs LongestCommonSegment
abbacbbbcbcbccbccaaabacacaaababc
$
```

Você pode também encontrar trechos repetidos em textos, fazendo, por exemplo,

```
$ java-introcs LongestCommonSegment anna2.txt jean2.txt
it had been raining all the morning, and
$
```

Acima, anna2.txt contém o texto de Anna Karenina, de Tolstoy, do projeto Gutenberg, mas sem o trecho inicial e final do arquivo que falam sobre o projeto Gutenberg (jean2.txt é o texto de Les Misérables, de Hugo).

Para evitar que espaços em branco e mudanças de linha interfiram quando dois textos são comparados, elimine "espaços contíguos" e "espaços no começo e no fim do string", fazendo

```
s = s.replaceAll("\\s+", " ").trim();
t = t.replaceAll("\\s+", " ").trim();
```

nos seus dois strings `s` e `t` a serem processados.

(II) Subpalíndromo mais longo

Uma palíndromo é um string que coincide com seu reverso (o string lido de trás para frente). Por exemplo, são palíndromos as palavras noon, civic, radar, level, rotor, kayak, reviver, racecar, redder, madam, e refer (em português, sopapos é um belo palíndromo).

Escreva um programa chamado LongestPalindrome.java que, ao receber um string `s` de entrada, imprime um segmento `u` de `s` que é um palíndromo e que tem comprimento maior possível. Por exemplo, se `s` é o string em `s.in`, então seu programa poderia imprimir a saída `aacaa`.

Novamente, seu programa deve ter dois modos de execução, como ilustrado abaixo:

```
$ java-introcs LongestPalindrome < s.in
aacaa
$ java-introcs LongestPalindrome s.in
aacaa
$
```

LongestPalindrome.java deve eliminar espaços contíguos e espaços no começo e no fim do string de entrada (use `replaceAll("\\s+", " ").trim()`).

Você pode usar Generator para gerar entradas para seu programa:

```
$ java-introcs Generator 1 1000000 ACTG | java-introcs LongestPalindrome
TCAACGGGGCCACCGGGGCAACT
$ java-introcs Generator 1 2000000 ACTG | java-introcs LongestPalindrome
CCTCATGGATCTAGGTACTCC
```

```
$ java-introcs Generator 1 4000000 ACTG | java-introcs LongestPalindrome  
ACCTGATGTAAATGTAGTCCA
```

```
$
```

Importante. Estude a classe `SuffixArray.java` de Sedgewick e Wayne (anexo). Escreva uma classe similar `SuffixArray2.java` para escrever seus programas `LongestCommonSegment.java` e `LongestPalindrome.java` (assim, você deve entregar três arquivos neste exercício).

[s.in](#)[st.in](#)[SuffixArray.java](#)[t.in](#)

Status de envio

Status de envio	Nenhuma tentativa
Status da avaliação	Não há notas
Data de entrega	sexta, 22 Set 2017, 23:55
Tempo restante	2 dias 13 horas
Última modificação	-
Comentários ao envio	► Comentários (0)

Enviar

Fazer alterações em seu envio

Você acessou como Édio Cerati Neto (Sair)
mac121.2017ii

Português - Brasil (pt_br)
English (en)
Português - Brasil (pt_br)