# AULA 21

# Busca de palavras (string matching)

CLRS 32

# Busca de palavras em um texto

Dizemos que um vetor $P[1\,..\,m]$ ocorre em um vetor $T[1\,..\,n]$ se

$$P[1\,..\,m] = T[s+1\,..\,s+m]$$

para algum $s$ em $[0\,..\,n-m]$.

Exemplo:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $T$ | $x$ | $c$ | $b$ | $a$ | $b$ | $b$ | $c$ | $b$ | $a$ | $x$ |

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $P$ | $b$ | $c$ | $b$ | $a$ |

$P[1..4]$ ocorre em $T[1..10]$ com deslocamento 5.

O valor $s$ é um descolamento válido

# Busca de palavras em um texto

Problema: Dados $P[1\,..\,m]$ e $T[1\,..\,n]$, encontrar todos os deslocamentos válidos.

Exemplo:

Para $n = 10$, $m = 4$, e

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $T$ | $b$ | $b$ | $a$ | $b$ | $a$ | $b$ | $a$ | $c$ | $b$ | $a$ |

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $P$ | $b$ | $a$ | $b$ | $a$ |

Os deslocamentos válidos são 1 e 3.

# Simulação

$P =$ a b a b b a b a b b a

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | a | b | a | a | b | a | b | a | b | b | a | b | a | b | a | b | b | a | b | a | b | b | a | $T$ |
| 1 | a | b | a | b | | | | | | | | | | | | | | | | | | | | |
| 2 | | a | | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | a | b | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | a | b | a | b | b | | | | | | | | | | | | | | | | |
| 5 | | | | | a | | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | a | b | a | b | b | a | b | a | b | b | | | | | | | | | |
| 7 | | | | | | | a | | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | a | b | a | | | | | | | | | | | | | | |
| 9 | | | | | | | | | a | | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | a | | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | a | b | a | b | | | | | | | | | | |
| 12 | | | | | | | | | | | | a | | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | a | b | a | b | b | a | b | a | b | b | a | |

# Naive-String-Matcher

Recebe $P[1\,..\,m]$ e $T[1\,..\,n]$, e devolve todos os deslocamentos válidos.

NAIVE-STRING-MATCHER $(P, m, T, n)$

```
1   para i ← 1 até n − m faça
2       q ← 0
3       enquanto q < m e P[q + 1] = T[i + q] faça
4           q ← q + 1
5       se q = m
6           então "P ocorre com deslocamento i − 1"
```

Relação invariante: na linha 3 vale que

(i0) $P[1\,..\,q] = T[i\,..\,i{+}q{-}1]$

# Consumo de tempo

| linha | consumo de todas as execuções da linha |
|---|---|
| 1 | $\Theta(n-m+1)$ |
| 2 | $\Theta(n-m)$ |
| 3 | $\mathrm{O}((n-m)(m+1)) = \mathrm{O}((n-m)m)$ |
| 4 | $\mathrm{O}((n-m)m)$ |
| 5 | $\Theta(n-m)$ |
| 6 | $\mathrm{O}(n-m)$ |
| total | $\Theta(3(n-m)+1) + \mathrm{O}(2(n-m)m + n - m)$ $= \mathrm{O}((n-m+1)m)$ |

# Conclusões

O consumo de tempo do algoritmo NAIVE-STRING-MATCHER é $\mathrm{O}((n-m+1)m)$.

No pior caso, o consumo de tempo do algoritmo NAIVE-STRING-MATCHER é $\Theta((n-m+1)m)$.

# Naive-String-Matcher

NAIVE-STRING-MATCHER $(P, m, T, n)$

```
 1   i ← 1   q ← 0
 2   enquanto i ≤ n − m + 1 faça
 3         se q = m então ▷ caso 1
 4               "P ocorre com deslocamento i − 1"
 5               q ← 0
 6               i ← i + 1
 7         senão se P[q+1] = T[i+q] então ▷ caso 2
 8               q ← q + 1
 9         senão se P[q+1] ≠ T[i+q] então ▷ caso 3
10               q ← 0
11               i ← i + 1
```

Relação invariante: na linha 2 vale que

(i0) $P[1\,..\,q] = T[i\,..\,i{+}q{-}1]$

# Consumo de tempo

Cada linha do algoritmo consome tempo $\Theta(1)$.

Caso 1 e caso 3 ocorrem $n - m + 1$ vezes (total).

Para cada valor de $i$ o número de ocorrências do caso 2
é $\leq m$.

Logo, o número total de iterações das linhas 2–13
é $\leq (n - m + 1)m$.

Portanto, o consumo de tempo algoritmo é $\Theta((n - m + 1)m)$.

Mas, isto já sabíamos. . .

# Simulação

$P =$ a b a b b a b a b b a

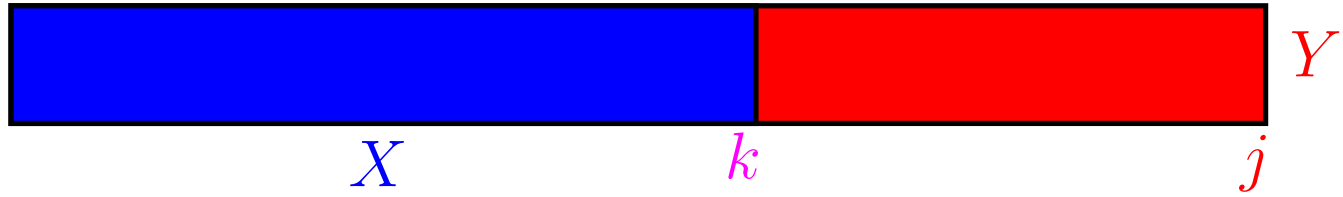
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | | | | | | $T$ |
| 1 | a | b | a | b | | | | | | | | | | | | | | | | | | | |
| 2 | | a | | | | | | | | | | | | | | | | | | | | | |
| 3 | | | a | b | | | | | | | | | | | | | | | | | | | |
| 4 | | | | a | b | a | b | b | | | | | | | | | | | | | | | |
| 5 | | | | | a | | | | | | | | | | | | | | | | | | |
| 6 | | | | | | a | b | a | b | b | a | b | a | b | b | | | | | | | | |
| 7 | | | | | | | a | | | | | | | | | | | | | | | | |
| 8 | | | | | | | | a | b | a | | | | | | | | | | | | | |
| 9 | | | | | | | | | a | | | | | | | | | | | | | | |
| 10 | | | | | | | | | | a | | | | | | | | | | | | | |
| 11 | | | | | | | | | | | a | b | a | b | | | | | | | | | |
| 12 | | | | | | | | | | | | a | | | | | | | | | | | |
| 13 | | | | | | | | | | | | | a | b | a | b | b | a | b | a | b | b | a |

# Prefixos e sulfixos

$X[1\,..\,k]$ é prefixo de $Y[1\,..\,j]$ se

$$k \leq j \quad \text{e} \quad X[1\,..\,k] = Y[1\,..\,k].$$

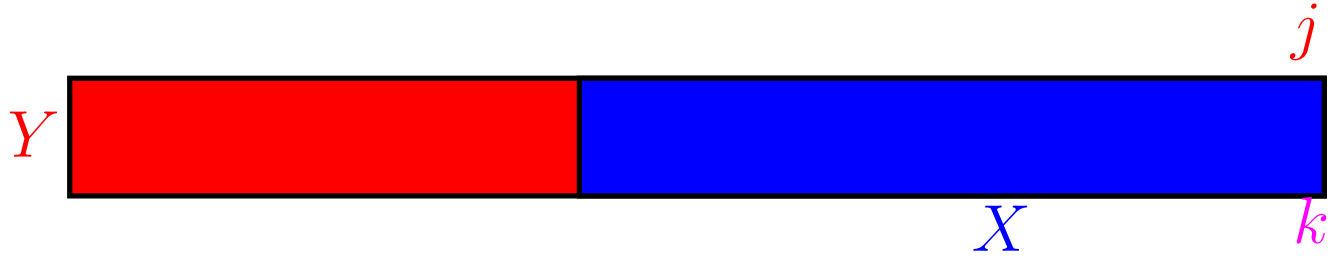
Se $k < j$, então $X[1\,..\,k]$ é prefixo próprio.




$X[1\,..\,k]$ é sulfixo de $Y[1\,..\,j]$ se

$$k \leq j \quad \text{e} \quad X[1\,..\,k] = Y[j-k+1\,..\,j].$$

Se $k < j$, então $X[1\,..\,k]$ é sulfixo próprio.



# Exemplos

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $P$ | $a$ | $b$ | $a$ | $b$ | $b$ | $a$ | $b$ | $a$ | $b$ | $b$ | $a$ |

$P[1\,..\,1]$ é prefixo próprio de $P[1\,..\,11]$

$P[1\,..\,3]$ é prefixo próprio de $P[1\,..\,7]$

$P[1\,..\,3]$ é sulfixo próprio de $P[1\,..\,8]$

$P[1\,..\,5]$ é sulfixo próprio de $P[1\,..\,10]$

$P[1\,..\,1]$ é sulfixo próprio de $P[1\,..\,11]$

# Nova simulação

$P =$ a b a b b a b a b b a

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | a | b | a | a | b | a | b | a | b | b | a | b | a | b | a | b | b | a | b | a | b | b | a | $T$ |
| 1 | a | b | a | b | | | | | | | | | | | | | | | | | | | | |
| 4 | | | a | b | | | | | | | | | | | | | | | | | | | | |
| 4 | | | | a | b | a | b | b | | | | | | | | | | | | | | | | |
| 8 | | | | | | a | b | a | b | b | a | b | a | b | b | | | | | | | | | |
| 15 | | | | | | | | | | | a | b | a | b | b | | | | | | | | | |
| 15 | | | | | | | | | | | | | a | b | a | b | b | a | b | a | b | b | a | |
| $i$ | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |

# Função prefixo

$\pi[q] :=$ maior comprimento de um prefixo próprio de $P[1\,..\,q]$ que é sulfixo de $P[1\,..\,q]$

$:= \max\{j : j < q$ e $P[1\,..\,j]$ é sul£xo de $P[1\,..\,q]\}$

| $q$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $P$ | a | b | a | b | b | a | b | a | b | b | a |
| $\pi$ | 0 | 0 | 1 | 2 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

| $q$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $P$ | a | b | a | b | a | b | a | b | c | a |
| $\pi$ | 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 0 | 1 |

# KMP-Matcher

KMP-MATCHER $(P, m, T, n)$

```
 0   π ← COMPUTE-PREFIX-FUNCTION(P, m)
 1   i ← 1   q ← 0
 2   enquanto i ≤ n + 1 faça
 3         se q = m então ▷ caso 1
 4               "P ocorre com deslocamento i − m"
 5               q ← π[q]
 6         senão se P[q+1] = T[i] então ▷ caso 2
 7               q ← q + 1
 8               i ← i + 1
 9         senão se P[q+1] ≠ T[i] e q > 0 então ▷ caso 3
10               q ← π[q]
11         senão se P[q+1] ≠ T[i] e q = 0 então ▷ caso 4
12               i ← i + 1
```

Suponha $T[n+1] =$ "símbolo que não ocorre em $P[1\,..\,m]$"

# Correção

Relação invariante: na linha 2 vale que

(i0) $P[1\,..\,q] = T[i-q\,..\,i-1]$

Na linha 2 vale ainda que

(i1) $P[1\,..\,k]$ não é sulfixo de $T[1\,..\,i]$, para $k \geq q+2$.

Invariante $(i1)$ implica que

$$i-m, i-m+1, \ldots, i-q-1$$

não são deslocamentos válidos, ou seja

$$P[1\,..\,m] \neq T[\,..\,i+1], P[1\,..\,m] \neq T[\,..\,i+2], \ldots$$
$$P[1\,..\,m] \neq T[\,..\,i+m-q-1].$$

# Consumo de tempo

Exceto a linha 0, cada linha do algoritmo consome tempo $\Theta(1)$.

Caso 2 e caso 4 ocorrem $n + 1$ vezes (total).

Para cada valor de $i$, número de ocorrências do caso 1 e caso 3 é $\leq m$.

Logo, o número total de iterações das linhas 2–12 é $\leq n + 1 + (n + 1)m = (n + 1)(m + 1)$.

Portanto, o consumo de tempo total das linhas 1–12 é $\mathrm{O}(nm)$.

# Consumo de tempo

Exceto a linha 0, cada linha do algoritmo consome tempo $\Theta(1)$.

Caso 2 e caso 4 ocorrem $n + 1$ vezes (total).

Para cada valor de $i$, número de ocorrências do caso 1 e caso 3 é $\leq m$.

Logo, o número total de iterações das linhas 2–12 é $\leq n + 1 + (n + 1)m = (n + 1)(m + 1)$.

Portanto, o consumo de tempo total das linhas 1–12 é $\mathrm{O}(nm)$.

EXAGERO!

# Consumo de tempo

Exceto a linha 0, cada linha do algoritmo consome tempo $\Theta(1)$.

Caso 2 e caso 4 ocorrem $n+1$ vezes (total).

O número de ocorrências do caso 1 e caso 3 é $\leq$ número de ocorrências do caso 2, pois

- $q$ nunca é negativo;
- $q$ incrementado no caso 2;
- $q$ é decrementado no caso 1 e no caso 3.

Logo, o número total de iterações é $\leq n+1+n+1 = 2n+2$.

Portanto, o consumo de tempo das linhas 1–12 é $\Theta(n)$.

# KMP-Matcher

KMP-MATCHER $(P, m, T, n)$

```
0   π ← COMPUTE-PREFIX-FUNCTION(P, m)
1   i ← 1   q ← 0
2   enquanto i ≤ n faça
3       enquanto q > 0 e P[q+1] ≠ T[i] faça
4           q ← π[q]
5           se P[q+1] = T[i] então
6               q ← q + 1
7           se q = m então
8               "P ocorre com deslocamento i − m"
9               q ← π[q]
```

# Compute-Pre£x-Function (grosseiro)

COMPUTE-PREFIX-FUNCTION $(P, m)$

```
 0   π[1] ← 0
 1   q ← 2
 2   enquanto q ≤ m faça
 3         j ← q − 1
 4         repita
 5               i ← j
 6               k ← q
 7               enquanto i > 0 e P[i] = P[k] faça
 8                     i ← i − 1    k ← k − 1
 9               se i > 0
10                     então j ← j − 1
11         até que i = 0
12         π[q] ← j
13   devolva π
```

# Consumo de tempo

| linha | consumo de todas as execuções da linha |
|---|---|
| 0-1 | $\Theta(1)$ |
| 2 | $\Theta(m)$ |
| 3 | $\Theta(m-1)$ |
| 4-6 | $\mathrm{O}(m^2)$ |
| 7-8 | $\mathrm{O}(m^3)$ |
| 9-11 | $\mathrm{O}(m^2)$ |
| 12 | $\Theta(m-1)$ |
| 13 | $\mathrm{O}(m)$ |
| total | $\Theta(3m-1)+\mathrm{O}(m^3+2m^2+m)$ |
| | $=\mathrm{O}(m^3)$ |

# Subestrutura ótima

Suponha que $P[1\,..\,j+1]$ é o maior prefixo próprio de $P[1\,..\,q]$ que é sulfixo de $P[1\,..\,q]$ ($\pi[q] = j+1$).

# Subestrutura ótima

Suponha que $P[1\,..\,j+1]$ é o maior prefixo próprio de $P[1\,..\,q]$ que é sulfixo de $P[1\,..\,q]$ ($\pi[q] = j+1$).

Portanto,

- $P[j+1] = P[q]$ e
- $P[1\,..\,j]$ é prefixo próprio e sulfixo de $P[1\,..\,q-1]$.

Logo, $P[1\,..\,j]$ é prefixo e sulfixo de $P[1\,..\,\pi[q-1]]$.

# Subestrutura ótima

Suponha que $P[1\,..\,j+1]$ é o maior prefixo próprio de $P[1\,..\,q]$ que é sulfixo de $P[1\,..\,q]$ ($\pi[q] = j+1$).

Portanto,

- $P[j+1] = P[q]$ e
- $P[1\,..\,j]$ é prefixo próprio e sulfixo de $P[1\,..\,q-1]$.

Logo, $P[1\,..\,j]$ é prefixo e sulfixo de $P[1\,..\,\pi[q-1]]$.

Se $j \neq \pi[q-1]$, então

# Subestrutura ótima

Suponha que $P[1\,..\,j+1]$ é o maior prefixo próprio de $P[1\,..\,q]$ que é sulfixo de $P[1\,..\,q]$ ($\pi[q] = j+1$).

Portanto,

- $P[j+1] = P[q]$ e
- $P[1\,..\,j]$ é prefixo próprio e sulfixo de $P[1\,..\,q-1]$.

Logo, $P[1\,..\,j]$ é prefixo e sulfixo de $P[1\,..\,\pi[q-1]]$.

Se $j \neq \pi[q-1]$, então

$P[1\,..\,j]$ é prefixo próprio e sulfixo de $P[1\,..\,\pi[q-1]]$.

Logo, $P[1\,..\,j]$ é prefixo e sulfixo de

$$P[1\,..\,\pi[\pi[q-1]]] = P[1\,..\,\pi^2[q-1]].$$

# Subestrutura ótima (cont.)

Se $j \neq \pi^2[q-1]$, então

# Subestrutura ótima (cont.)

Se $j \neq \pi^2[q-1]$, então

$P[1\,..\,j]$ é prefixo próprio e sulfixo de $P[1\,..\,\pi^2[q-1]]$.

Logo, $P[1\,..\,j]$ é sulfixo de

$$P[1\,..\,\pi[\pi[\pi[q-1]]]] = P[1\,..\,\pi^3[q-1]].$$

# Subestrutura ótima (cont.)

Se $j \neq \pi^2[q-1]$, então

$P[1\,..\,j]$ é prefixo próprio e sulfixo de $P[1\,..\,\pi^2[q-1]]$.

Logo, $P[1\,..\,j]$ é sulfixo de

$$P[1\,..\,\pi[\pi[\pi[q-1]]]] = P[1\,..\,\pi^3[q-1]].$$

Se $j \neq \pi^3[q-1]$, então

$P[1\,..\,j-1]$ é prefixo próprio e sulfixo de $P[1\,..\,\pi^3[q-1]]$.

Logo, $P[1\,..\,j]$ é prefixo e sulfixo de $P[1\,..\,\pi^4[q-1]]$.

Se $j \neq \pi^4[q-1]$, então ...

# Recorrência

$\pi[q] :=$ maior comprimento de um prefixo próprio de $P[1\,..\,q]$ que é sulfixo de $P[1\,..\,q]$

$:= \max\{j : j < q$ e $P[1\,..\,j]$ é sul£xo de $P[1\,..\,q]\}$

| $q$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P$ | '?' | a | b | a | b | b | a | b | a | b | b | a |
| $\pi$ | -1 | 0 | 0 | 1 | 2 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

$\pi[0] = -1$

$\pi[1] = 0$

$\pi[q] = \max\{\pi^k[q-1] + 1 : P[\pi^k[q-1]+1] = P[q]\}$

Suponha $P[0] =$ "coringa".

# Compute-Pre£x-Function (grosseiro)

COMPUTE-PREFIX-FUNCTION $(P, m)$

```
 0   π[1] ← 0
 1   q ← 2
 2   enquanto q ≤ m faça
 3         j ← q − 1
 4         repita
 5               i ← j
 6               k ← q
 7               enquanto i > 0 e P[i] = P[k] faça
 8                     i ← i − 1    k ← k − 1
 9               se i > 0
10                     então j ← j − 1
11         até que i = 0
12         π[q] ← j
13   devolva π
```

# Compute-Pre£x-Function (melhorado)

COMPUTE-PREFIX-FUNCTION $(P, m)$

```
 0   π[1] ← 0
 1   q ← 2
 2   enquanto q ≤ m faça
 3         j ← π[q − 1]
 4         repita
 5               i ← j + 1
 6               k ← q
 7               enquanto i > 0 e P[i + 1] = P[k] faça
 8                     i ← i − 1    k ← k − 1
 9               se i > 0
10                     então j ← π[j]
11         até que i = 0
12         π[q] ← j + 1
13   devolva π
```

# Compute-Pre£x-Function (melhor ainda)

COMPUTE-PREFIX-FUNCTION $(P, m)$

```
 0   π[1] ← 0
 1   q ← 2
 2   enquanto q ≤ m faça
 3         j ← π[q − 1]
 4         repita
 5               i ← j + 1
 6               k ← q
 7               se P[i] = P[k]
 8                     então i ← 0
10                     senão j ← π[j]
11         até que i = 0
12         π[q] ← j + 1
13   devolva π
```

# Compute-Pre£x-Function (limpo)

COMPUTE-PREFIX-FUNCTION $(P, m)$

```
0   π[1] ← 0
1   q ← 2    j ← 0
2   enquanto q ≤ m faça
3       enquanto j > 0 e P[j + 1] ≠ P[q] faça
4           j ← π[j]
5       se P[j + 1] = P[q]
6           então j ← j + 1
7       π[q] ← j
8   devolva π
```

# Compute-Pre£x-Function (limpo)

COMPUTE-PREFIX-FUNCTION $(P, m)$

```
 0   π[1] ← 0
 1   q ← 2   j ← 0
 2   enquanto q ≤ m faça
 3         se P[j+1] = P[q] então ▷ caso 1
 4               π[q] ← j + 1
 5               q ← q + 1
 6               j ← j + 1
 7         senão se P[j+1] ≠ P[q] e j > 0 então ▷ caso 2
 8               j ← π[j]
 9         senão se P[j+1] ≠ P[q] e j = 0 então ▷ caso 3
10               π[q] ← 0
11               q ← q + 1
12   devolva π
```

# Correção

Relações invariantes: na linha 2 vale que

(i0) $P[1\,..\,j] = P[\,..\,q{-}1]$

(i1) $P[1\,..\,k]$ não é sulfixo de $P[1\,..\,q]$, para $k > j+1$.

# Consumo de tempo

Cada linha do algoritmo consome tempo $\Theta(1)$, exceto a linha 12 que pode consumir tempo $\mathrm{O}(m)$.

Caso 1 e caso 3 ocorrem $m$ vezes (total).

O número de ocorrências do caso 2 é $\leq$ número de ocorrências do caso 1, pois

- $j$ nunca é negativo;
- $j$ é incrementado no caso 1; e
- $j$ é decrementado no caso 2.

Logo, o número total de iterações é $\leq m + m = 2m$.

O consumo de tempo do algoritmo é $\Theta(m)$.

# Conclusões

O consumo de tempo do algoritmo COMPUTE-PREFIX-FUNCTION é $\Theta(m)$.

O consumo de tempo do algoritmo KMP-MATCHER é $\Theta(n+m)$.