

Análise de Algoritmos

**Parte destes slides são adaptações de slides
do Prof. Paulo Feofiloff e do Prof. José Coelho de Pina.**

Busca de padrão

Dados

- uma palavra $P[1..m]$ e
- um texto $T[1..n]$,

uma **ocorrência** de P em T é um índice s tal que $T[s + j] = P[j]$ para $j = 1, \dots, m$.

Busca de padrão

Dados

- uma palavra $P[1..m]$ e
- um texto $T[1..n]$,

uma ocorrência de P em T é um índice s tal que $T[s + j] = P[j]$ para $j = 1, \dots, m$.

Exemplo:

	1	2	3		1	2	3	4	5	6	7	8	9	10	11
P	B	R	A		A	B	R	A	C	A	D	A	B	R	A

Busca de padrão

Dados

- uma palavra $P[1..m]$ e
- um texto $T[1..n]$,

uma **ocorrência** de P em T é um índice s tal que $T[s + j] = P[j]$ para $j = 1, \dots, m$.

Exemplo:

	1	2	3		1	2	3	4	5	6	7	8	9	10	11	
P	B	R	A		T	A	B	R	A	C	A	D	A	B	R	A

Problema: Dada uma palavra $P[1..m]$ e um texto $T[1..n]$, encontrar todas as ocorrências de P em T .

Busca de padrão

Dados

- uma palavra $P[1..m]$ e
- um texto $T[1..n]$,

uma **ocorrência** de P em T é um índice s tal que $T[s + j] = P[j]$ para $j = 1, \dots, m$.

Exemplo:

	1	2	3		1	2	3	4	5	6	7	8	9	10	11	
P	B	R	A		T	A	B	R	A	C	A	D	A	B	R	A

Problema: Dada uma palavra $P[1..m]$ e um texto $T[1..n]$, encontrar todas as ocorrências de P em T .

No exemplo, P ocorre duas vezes em T : em 1 e em 8.

Algoritmo Boyer-Moore 1

Supondo que o alfabeto é o conjunto $\Sigma = \{0, \dots, 255\}$.

BM1 (P, m, T, n)

```
1  para  $i \leftarrow 0$  até 255 faça  $v_1[i] \leftarrow m$ 
2  para  $i \leftarrow 1$  até  $m$  faça  $v_1[P[i]] \leftarrow m - i$ 
3   $c \leftarrow 0$        $k \leftarrow m$ 
4  enquanto  $k \leq n$  faça
5       $r \leftarrow 0$ 
6      enquanto  $m - r \geq 1$  e  $P[m - r] = T[k - r]$  faça
7           $r \leftarrow r + 1$ 
8      se  $m - r < 1$  então  $c \leftarrow c + 1$ 
9      se  $k = n$ 
10         então  $k \leftarrow k + 1$ 
11         senão  $k \leftarrow k + 1 + v_1[T[k + 1]]$ 
12  devolva  $c$ 
```

Algoritmo Boyer-Moore 1

Supondo que o alfabeto é o conjunto $\Sigma = \{0, \dots, 255\}$.

BM1 (P, m, T, n)

```
1  para  $i \leftarrow 0$  até 255 faça  $v_1[i] \leftarrow m$ 
2  para  $i \leftarrow 1$  até  $m$  faça  $v_1[P[i]] \leftarrow m - i$ 
3   $c \leftarrow 0$        $k \leftarrow m$ 
4  enquanto  $k \leq n$  faça
5       $r \leftarrow 0$ 
6      enquanto  $m - r \geq 1$  e  $P[m - r] = T[k - r]$  faça
7           $r \leftarrow r + 1$ 
8      se  $m - r < 1$  então  $c \leftarrow c + 1$ 
9      se  $k = n$ 
10         então  $k \leftarrow k + 1$ 
11         senão  $k \leftarrow k + 1 + v_1[T[k + 1]]$ 
12  devolva  $c$ 
```

Consumo de tempo: $\Theta(|\Sigma| + mn)$

Algoritmo Boyer-Moore 2

BM2 (P, m, T, n)

1 **para** $i \leftarrow m$ **decrecendo até** 1 **faça**

2 $j \leftarrow m - 1$ $r \leftarrow 0$

3 **enquanto** $m - r \geq i$ **e** $j - r \geq 1$ **faça**

4 **se** $P[m - r] = P[j - r]$ **então** $r \leftarrow r + 1$

5 **senão** $j \leftarrow j - 1$ $r \leftarrow 0$

6 $v_2[i] \leftarrow m - j$

Algoritmo Boyer-Moore 2

BM2 (P, m, T, n)

```
1  para  $i \leftarrow m$  decrescendo até 1 faça
    ...
6       $v_2[i] \leftarrow m - j$ 
7   $c \leftarrow 0$        $k \leftarrow m$ 
8  enquanto  $k \leq n$  faça
9       $r \leftarrow 0$ 
10     enquanto  $m - r \geq 1$  e  $P[m - r] = T[k - r]$  faça
11          $r \leftarrow r + 1$ 
12     se  $m - r < 1$  então  $c \leftarrow c + 1$ 
13     se  $r = 0$  então  $k \leftarrow k + 1$ 
14     senão  $k \leftarrow k + v_2[m - r + 1]$ 
15 devolva  $c$ 
```

Algoritmo Boyer-Moore 2

BM2 (P, m, T, n)

```
1  para  $i \leftarrow m$  decrescendo até 1 faça
    ...
6       $v_2[i] \leftarrow m - j$ 
7   $c \leftarrow 0$        $k \leftarrow m$ 
8  enquanto  $k \leq n$  faça
9       $r \leftarrow 0$ 
10     enquanto  $m - r \geq 1$  e  $P[m - r] = T[k - r]$  faça
11          $r \leftarrow r + 1$ 
12     se  $m - r < 1$  então  $c \leftarrow c + 1$ 
13     se  $r = 0$  então  $k \leftarrow k + 1$ 
14         senão  $k \leftarrow k + v_2[m - r + 1]$ 
15  devolva  $c$ 
```

Consumo de tempo: $\Theta(mn)$

Algoritmo Boyer-Moore

Calcula v_1 e v_2 , e
atualiza fazendo sempre o maior dos dois deslocamentos.

Algoritmo Boyer-Moore

Calcula v_1 e v_2 , e
atualiza fazendo sempre o maior dos dois deslocamentos.

Consumo de tempo: $\Theta(mn)$
(fazendo melhora para inicializar v_1)

Algoritmo Boyer-Moore

Calcula v_1 e v_2 , e
atualiza fazendo sempre o maior dos dois deslocamentos.

Consumo de tempo: $\Theta(mn)$
(fazendo melhora para inicializar v_1)

No caso médio entretanto, ele consome apenas $O(n)$.

É o mais rápido algoritmo de busca de padrão na prática.

Algoritmo Boyer-Moore

Calcula v_1 e v_2 , e atualiza fazendo sempre o maior dos dois deslocamentos.

Consumo de tempo: $\Theta(mn)$
(fazendo melhora para inicializar v_1)

No caso médio entretanto, ele consome apenas $O(n)$.

É o mais rápido algoritmo de busca de padrão na prática.

É possível aperfeiçoar a tabela v_2 para que este algoritmo consuma apenas $O(m + n)$ no pior caso.

Algoritmo Boyer-Moore

Calcula v_1 e v_2 , e atualiza fazendo sempre o maior dos dois deslocamentos.

Consumo de tempo: $\Theta(mn)$
(fazendo melhora para inicializar v_1)

No caso médio entretanto, ele consome apenas $O(n)$.

É o mais rápido algoritmo de busca de padrão na prática.

É possível aperfeiçoar a tabela v_2 para que este algoritmo consuma apenas $O(m + n)$ no pior caso.

Veja os exercícios nas notas de aula indicadas para leitura.

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_1

A	B	C	D	R
0	2	4	4	1

$k = 4$

$c = 0$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$$k = 4$$

$$c = 1$$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$$k = 4$$

$$c = 2$$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_1

A	B	C	D	R
0	2	4	4	1

$k = 4$

$c = 3$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_1

A	B	C	D	R
0	2	4	4	1

$k = 4$

$c = 4$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$$k = 4 + 4 + 1 = 9$$

$$c = 4$$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_1

A	B	C	D	R
0	2	4	4	1

$k = 9$

$c = 4$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_1

A	B	C	D	R
0	2	4	4	1

$k = 9$

$c = 5$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_1

A	B	C	D	R
0	2	4	4	1

$$k = 9 + 1 + 1 = 11$$

$$c = 5$$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$k = 11$

$c = 5$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$k = 11$

$c = 6$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$k = 11$

$c = 7$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$k = 11$

$c = 8$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$k = 11$

$c = 9$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$$k = 11 + 1 = 12 > 11$$

c = 9 comparações!

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_1

A	B	C	D	R
0	2	4	5	1

$k = 5$

$c = 0$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	5	1

$k = 5$

$c = 1$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	4	1

$$k = 5 + 0 + 1 = 6$$

$$c = 1$$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_1

A	B	C	D	R
0	2	4	5	1

$k = 6$

$c = 1$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_1

A	B	C	D	R
0	2	4	5	1

$k = 6$

$c = 2$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_1

A	B	C	D	R
0	2	4	5	1

$k = 6$

$c = 3$

Boyer-Moore: simulação v1

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	A	B	C	D	R
<i>v</i> ₁	0	2	4	5	1

$$k = 6 + 5 + 1 = 12 > 11$$

c = 3 comparações!!

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$k = 4$

$c = 0$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$k = 4$

$c = 1$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
T	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
P	A	B	R	A

	1	2	3	4
v_2	3	3	3	3

$$k = 4$$

$$c = 2$$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$k = 4$

$c = 3$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$k = 4$

$c = 4$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$$k = 4 + 3 = 7$$

$$c = 4$$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	1	2	3	4
<i>v₂</i>	3	3	3	3

$k = 7$

$c = 4$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	1	2	3	4
<i>v₂</i>	3	3	3	3

$$k = 7 + 1 = 8$$

$$c = 5$$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$k = 8$

$c = 5$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
T	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
P	A	B	R	A

	1	2	3	4
v_2	3	3	3	3

$k = 8$

$c = 6$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
T	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
P	A	B	R	A

	1	2	3	4
v_2	3	3	3	3

$k = 8$

$c = 7$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$$k = 8 + 3 = 11$$

$$c = 7$$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$k = 11$

$c = 7$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$k = 11$

$c = 8$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	1	2	3	4
<i>v</i> ₂	3	3	3	3

$k = 11$

$c = 9$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4
<i>P</i>	A	B	R	A

	1	2	3	4
<i>v</i> ₂	3	3	3	3

$k = 11$

$c = 10$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$k = 11$

$c = 11$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 4$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

v_2

1	2	3	4
3	3	3	3

$$k = 11 + 3 = 14 > 11$$

$c = 11$ comparações!

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_2

1	2	3	4	5
5	5	5	5	3

$k = 5$

$c = 0$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_2

1	2	3	4	5
5	5	5	5	3

$$k = 5 + 1 = 6$$

$$c = 1$$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_2

1	2	3	4	5
5	5	5	5	3

$k = 6$

$c = 1$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_2

1	2	3	4	5
5	5	5	5	3

$k = 6$

$c = 2$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_2

1	2	3	4	5
5	5	5	5	3

$k = 6$

$c = 3$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_2

1	2	3	4	5
5	5	5	5	3

$$k = 6 + 3 = 9$$

$$c = 3$$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	1	2	3	4	5
<i>v₂</i>	5	5	5	5	3

$k = 9$

$c = 3$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	1	2	3	4	5
<i>v₂</i>	5	5	5	5	3

$$k = 9 + 1 = 10$$

$$c = 4$$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_2

1	2	3	4	5
5	5	5	5	3

$k = 10$

$c = 4$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	1	2	3	4	5
<i>v₂</i>	5	5	5	5	3

$$k = 10 + 1 = 11$$

$$c = 5$$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_2

1	2	3	4	5
5	5	5	5	3

$k = 11$

$c = 5$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	1	2	3	4	5
<i>v</i> ₂	5	5	5	5	3

$k = 11$

$c = 6$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	1	2	3	4	5
<i>v</i> ₂	5	5	5	5	3

$k = 11$

$c = 7$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	1	2	3	4	5
<i>v</i> ₂	5	5	5	5	3

$k = 11$

$c = 8$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

v_2

1	2	3	4	5
5	5	5	5	3

$k = 11$

$c = 9$

Boyer-Moore: simulação v2

Exemplo: $n = 11$ $m = 5$

	1	2	3	4	5	6	7	8	9	10	11
<i>T</i>	A	B	R	A	C	A	D	A	B	R	A

	1	2	3	4	5
<i>P</i>	C	A	B	R	A

	1	2	3	4	5
<i>v</i> ₂	5	5	5	5	3

$$k = 11 + 1 = 12 > 11$$

$c = 10$ comparações!

Boyer-Moore: simulação v3

Exemplo: $n = 11$ $m = 5$

T

1	2	3	4	5	6	7	8	9	10	11
A	B	R	A	C	A	D	A	B	R	A

P

1	2	3	4
A	B	R	A

P

1	2	3	4	5
C	A	B	R	A

Exercício: Simule a versão 3 do algoritmo, que usa v_1 e v_2 , para as duas palavras acima. Quantas comparações são feitas em cada caso?

Algoritmo KMP

KMP: Knuth, Morris e Pratt.

Algoritmo KMP

KMP: Knuth, Morris e Pratt.

P_q : prefixo de P de comprimento q

Algoritmo KMP

KMP: Knuth, Morris e Pratt.

P_q : prefixo de P de comprimento q

$P_k \sqsupseteq P_q$: P_k é sufixo de P_q

Algoritmo KMP

KMP: Knuth, Morris e Pratt.

P_q : prefixo de P de comprimento q

$P_k \sqsupseteq P_q$: P_k é sufixo de P_q

Função prefixo: $\Pi[q] = \max\{k : k < q \text{ e } P_k \sqsupseteq P_q\}$

Algoritmo KMP

KMP: Knuth, Morris e Pratt.

P_q : prefixo de P de comprimento q

$P_k \sqsupseteq P_q$: P_k é sufixo de P_q

Função prefixo: $\Pi[q] = \max\{k : k < q \text{ e } P_k \sqsupseteq P_q\}$

Em palavras, $\Pi[q]$ é o comprimento do maior prefixo próprio de P_q que é sufixo de P_q .

Algoritmo KMP

KMP: Knuth, Morris e Pratt.

P_q : prefixo de P de comprimento q

$P_k \sqsupseteq P_q$: P_k é sufixo de P_q

Função prefixo: $\Pi[q] = \max\{k : k < q \text{ e } P_k \sqsupseteq P_q\}$

Em palavras, $\Pi[q]$ é o comprimento do maior prefixo próprio de P_q que é sufixo de P_q .

Exemplo:

P	a	b	a	b	a	a	b	a	b	c	a
Π	0	0	1	2	3	1	2	3	4	0	1

Algoritmo KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$

2 $k \leftarrow 0$

3 **para** $q \leftarrow 2$ **até** m **faça**

4 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

5 $k \leftarrow \Pi[k]$

6 **se** $P[k + 1] = P[q]$

7 **então** $k \leftarrow k + 1$

8 $\Pi[q] \leftarrow k$

9 **devolva** Π

Algoritmo KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$

2 $k \leftarrow 0$

3 **para** $q \leftarrow 2$ **até** m **faça**

4 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

5 $k \leftarrow \Pi[k]$

6 **se** $P[k + 1] = P[q]$

7 **então** $k \leftarrow k + 1$

8 $\Pi[q] \leftarrow k$

9 **devolva** Π

Consumo de tempo: $O(m^2)$

Algoritmo KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$

2 $k \leftarrow 0$

3 **para** $q \leftarrow 2$ **até** m **faça**

4 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

5 $k \leftarrow \Pi[k]$

6 **se** $P[k + 1] = P[q]$

7 **então** $k \leftarrow k + 1$

8 $\Pi[q] \leftarrow k$

9 **devolva** Π

Consumo de tempo: $\Theta(m^2)$??

Invariantes do Algoritmo KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

Invariantes do Algoritmo KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

(1) $k = \Pi[q - 1]$

(2) $\Pi[1 .. q - 1]$ está calculado corretamente

Consumo de tempo do KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

Método do potencial: Tome $\Phi_q = k$ no fim da iteração q .

Consumo de tempo do KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

Método do potencial: Tome $\Phi_q = k$ no fim da iteração q .

Valor inicial: $\Phi_1 = 0$

Consumo de tempo do KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

Método do potencial: Tome $\Phi_q = k$ no fim da iteração q .

$\Phi_1 = 0$ e $\Phi_q \geq 0$ para $q = 1, \dots, m$.

Consumo de tempo do KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

Método do potencial: Tome $\Phi_q = k$ no fim da iteração q .

$\Phi_1 = 0$ e $\Phi_q \geq 0$ para $q = 1, \dots, m$.

c_q : $1 +$ número de execuções da linha 4 na iteração q .

Consumo de tempo do KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

Método do potencial: Tome $\Phi_q = k$ no fim da iteração q .

$\Phi_1 = 0$ e $\Phi_q \geq 0$ para $q = 1, \dots, m$.

c_q : 1 + número de execuções da linha 4 na iteração q .

$$\hat{c}_q = c_q + \Phi_q - \Phi_{q-1}$$

Consumo de tempo do KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

$\Phi_q = k$ no fim da iteração q e $\Phi_1 = 0$.

c_q : 1 + número de execuções da linha 4 na iteração q .

$$\hat{c}_q = c_q + \Phi_q - \Phi_{q-1}$$

Consumo de tempo do KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

$\Phi_q = k$ no fim da iteração q e $\Phi_1 = 0$.

c_q : 1 + número de execuções da linha 4 na iteração q .

$$\hat{c}_q = c_q + \Phi_q - \Phi_{q-1}$$

Cada execução da linha 4 faz k diminuir de pelo menos 1.

Logo $\Phi_q \leq \Phi_{q-1} - (c_q - 1) + 1$, e $\hat{c}_q \leq c_q - (c_q - 1) + 1 = 2$.

Consumo de tempo do KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

$\Phi_q = k$ no fim da iteração q e $\Phi_1 = 0$.

c_q : 1 + número de execuções da linha 4 na iteração q .

$$\hat{c}_q = c_q + \Phi_q - \Phi_{q-1}$$

Cada execução da linha 4 faz k diminuir de pelo menos 1.

Logo $\Phi_q \leq \Phi_{q-1} - (c_q - 1) + 1$, e $\hat{c}_q \leq c_q - (c_q - 1) + 1 = 2$.

Custo amortizado por iteração: 2

Consumo de tempo do KMP

CALCULA-PREFIXO (P, m)

1 $\Pi[1] \leftarrow 0$ $k \leftarrow 0$

2 **para** $q \leftarrow 2$ **até** m **faça**

3 **enquanto** $k > 0$ **e** $P[k + 1] \neq P[q]$ **faça**

4 $k \leftarrow \Pi[k]$

5 **se** $P[k + 1] = P[q]$

6 **então** $k \leftarrow k + 1$

7 $\Pi[q] \leftarrow k$

8 **devolva** Π

$\Phi_q = k$ no fim da iteração q e $\Phi_1 = 0$.

c_q : 1 + número de execuções da linha 4 na iteração q .

$$\hat{c}_q = c_q + \Phi_q - \Phi_{q-1}$$

Cada execução da linha 4 faz k diminuir de pelo menos 1.

Logo $\Phi_q \leq \Phi_{q-1} - (c_q - 1) + 1$, e $\hat{c}_q \leq c_q - (c_q - 1) + 1 = 2$.

Consumo de tempo total: $\sum_q c_q \leq \sum_q \hat{c}_q \leq 2m = \Theta(m)$

Algoritmo KMP

```
KMP ( $T, n, P, m$ )
1   $\Pi \leftarrow \text{CALCULA-PREFIXO}(P, m)$ 
2   $q \leftarrow 0$ 
3  para  $i \leftarrow 1$  até  $n$  faça
4      enquanto  $q > 0$  e  $P[q + 1] \neq T[i]$  faça
5           $q \leftarrow \Pi[q]$ 
6      se  $P[q + 1] = T[i]$ 
7          então  $q \leftarrow q + 1$ 
8      se  $q = m$ 
9          então imprima  $i - q$ 
10          $q \leftarrow \Pi[q]$ 
```

Algoritmo KMP

Invariantes:

(1) $k = \Pi[q - 1]$

(2) $\Pi[1 .. q - 1]$ está calculado corretamente

Algoritmo KMP

Invariantes:

(1) $k = \Pi[q - 1]$

(2) $\Pi[1 .. q - 1]$ está calculado corretamente

Exercício: Complete a análise de correção, ou seja, prove os invariantes acima.

Algoritmo KMP

Invariantes:

(1) $k = \Pi[q - 1]$

(2) $\Pi[1 .. q - 1]$ está calculado corretamente

Exercício: Complete a análise de correção, ou seja, prove os invariantes acima.

Exercício: Simule o KMP com o texto e as duas palavras usadas nas simulações com o BM e calcule o número de comparações feitas para cada uma das duas palavras.