

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		T	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 $T[1..n]$, calcular o número de 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 $T[1..n]$, calcular o número de ocorrências de P em T .

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

Algoritmo ingênuo

BUSCA-TRIVIAL (T, n, P, m)

```
1   $c \leftarrow 0$ 
2  para  $s \leftarrow 0$  até  $n - m$  faça
3       $j \leftarrow 1$ 
4      enquanto  $j \leq m$  e  $P[j] = T[s + j]$  faça
5           $j \leftarrow j + 1$ 
6      se  $j > m$ 
7          então  $c \leftarrow c + 1$ 
8  return  $c$ 
```

Algoritmo ingênuo

BUSCA-TRIVIAL (T, n, P, m)

```
1    $c \leftarrow 0$ 
2   para  $s \leftarrow 0$  até  $n - m$  faça
3        $j \leftarrow 1$ 
4       enquanto  $j \leq m$  e  $P[j] = T[s + j]$  faça
5            $j \leftarrow j + 1$ 
6       se  $j > m$ 
7           então  $c \leftarrow c + 1$ 
8   return  $c$ 
```

Consumo de tempo: $O(mn)$

Algoritmo ingênuo

BUSCA-TRIVIAL (T, n, P, m)

```
1    $c \leftarrow 0$ 
2   para  $s \leftarrow 0$  até  $n - m$  faça
3        $j \leftarrow 1$ 
4       enquanto  $j \leq m$  e  $P[j] = T[s + j]$  faça
5            $j \leftarrow j + 1$ 
6       se  $j > m$ 
7           então  $c \leftarrow c + 1$ 
8   return  $c$ 
```

Consumo de tempo: $\Theta(mn)$

Exemplo: $T = a^n$ e $P = a^m$, ou $P = a^{m-1}b$.

Algoritmo ingênuo

BUSCA-TRIVIAL (T, n, P, m)

```
1   $c \leftarrow 0$ 
2  para  $s \leftarrow 0$  até  $n - m$  faça
3       $j \leftarrow 1$ 
4      enquanto  $j \leq m$  e  $P[j] = T[s + j]$  faça
5           $j \leftarrow j + 1$ 
6      se  $j > m$ 
7          então  $c \leftarrow c + 1$ 
8  return  $c$ 
```

Consumo de tempo: $\Theta(mn)$

Exemplo: $T = a^n$ e $P = a^m$, ou $P = a^{m-1}b$.

Nesta aula, vamos ver um algoritmo cujo consumo de tempo no pior caso é $\Theta(mn)$, porém na prática é $O(n)$, podendo inclusive ter um comportamento sublinear.

Algoritmo BM – primeira versão

BM: Boyer-Moore.

Algoritmo BM – primeira versão

BM: Boyer-Moore.

$v_1[c]$: menor t em $0 \dots m - 1$ tal que $P[m - t] = c$
(se c não aparece em P , então $v_1[c] = m$)

Algoritmo BM – primeira versão

BM: Boyer-Moore.

$v_1[c]$: menor t em $0 \dots m - 1$ tal que $P[m - t] = c$
(se c não aparece em P , então $v_1[c] = m$)

$v_1[c]$ corresponde à última ocorrência de c em P .

Algoritmo BM – primeira versão

BM: Boyer-Moore.

$v_1[c]$: menor t em $0 \dots m - 1$ tal que $P[m - t] = c$
(se c não aparece em P , então $v_1[c] = m$)

$v_1[c]$ corresponde à última ocorrência de c em P .

Exemplo:

	1	2	3	4	5	6	7	8	9	10	11
P	a	b	b	c	a	b	d	a	c	a	d
				a	b	c	d				
v_1	1	5	2	0							

Algoritmo BM – primeira versão

BM: Boyer-Moore.

$v_1[c]$: menor t em $0 \dots m - 1$ tal que $P[m - t] = c$
(se c não aparece em P , então $v_1[c] = m$)

$v_1[c]$ corresponde à última ocorrência de c em P .

Exemplo:

	1	2	3	4	5	6	7	8	9	10	11
P	a	b	b	c	a	b	d	a	c	a	d
				a	b	c	d				
v_1	1	5	2	0							

Seja $k \geq m$. Suponha que $c = T[k + 1]$ e que $v_1[c] = 4$. Então $P[1 \dots m]$ não é um sufixo de $T[1 \dots k + 1], \dots, T[1 \dots k + 4]$.
(Supondo que $k + 4 \leq n$.)

Algoritmo BM – primeira versão

BM: Boyer-Moore.

$v_1[c]$: menor t em $0 \dots m - 1$ tal que $P[m - t] = c$
(se c não aparece em P , então $v_1[c] = m$)

$v_1[c]$ corresponde à última ocorrência de c em P .

Exemplo:

	1	2	3	4	5	6	7	8	9	10	11
P	a	b	b	c	a	b	d	a	c	a	d
				a	b	c	d				
v_1	1	5	2	0							

Seja $k \geq m$. Suponha que $c = T[k + 1]$. Então

$P[1 \dots m]$ não é um sufixo de $T[1 \dots k+1], \dots, T[1 \dots k+\mu]$,

onde $\mu = \min\{n - k, v_1[c]\}$.

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  $c$ 
```

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 BM – segunda versão

j em $1..m-1$ é bom para i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

Algoritmo BM – segunda versão

j em $1..m-1$ é **bom para** i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$
é sufixo do mais longo dos dois.

Algoritmo BM – segunda versão

j em $1..m-1$ é bom para i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$
é sufixo do mais longo dos dois.

Exemplos:

	1	2	3	4	5	6
P	c	a	a	b	a	a

Algoritmo BM – segunda versão

j em $1..m-1$ é bom para i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$
é sufixo do mais longo dos dois.

Exemplos:

	1	2	3	4	5	6
P	c	a	a	b	a	a

$j = 3$ é bom para $i = 5$, pois aa é sufixo de caa .

Algoritmo BM – segunda versão

j em $1..m-1$ é bom para i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$
é sufixo do mais longo dos dois.

Exemplos:

	1	2	3	4	5	6	7	8	9
P	c	a	a	b	a	a	b	a	a

Algoritmo BM – segunda versão

j em $1..m-1$ é bom para i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$
é sufixo do mais longo dos dois.

Exemplos:

	1	2	3	4	5	6	7	8	9
P	c	a	a	b	a	a	b	a	a

$j = 6$ é bom para $i = 8$, pois aa é sufixo de $caabaa$.

Algoritmo BM – segunda versão

j em $1..m-1$ é bom para i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$
é sufixo do mais longo dos dois.

Exemplos:

	1	2	3	4	5	6	7	8	9
P	c	a	a	b	a	a	b	a	a

$j = 6$ é bom para $i = 8$, pois aa é sufixo de $caabaa$.

$j = 6$ é bom para $i = 5$, pois $aabaa$ é sufixo de $caabaa$.

Algoritmo BM – segunda versão

j em $1..m-1$ é bom para i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$
é sufixo do mais longo dos dois.

Exemplos:

	1	2	3	4	5	6	7	8	9
P	c	a	a	b	a	a	b	a	a

$j = 6$ é bom para $i = 8$, pois aa é sufixo de $caabaa$.

$j = 6$ é bom para $i = 5$, pois $aabaa$ é sufixo de $caabaa$.

$j = 3$ é bom para $i = 8$, pois aa é sufixo de caa .

Algoritmo BM – segunda versão

j em $1..m-1$ é bom para i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$
é sufixo do mais longo dos dois.

Exemplos:

	1	2	3	4	5	6	7	8	9
P	c	a	a	b	a	a	c	a	a

$j = 3$ é bom para $i = 6$, pois caa é sufixo de $acaa$.

Algoritmo BM – segunda versão

j em $1..m-1$ é bom para i se $P[m-\mu..m] = P[j-\mu..j]$,
onde $\mu = \min\{m-i, j-1\}$.

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$
é sufixo do mais longo dos dois.

Exemplos:

	1	2	3	4	5	6	7	8	9
P	c	a	a	b	a	a	c	a	a

$j = 3$ é bom para $i = 6$, pois caa é sufixo de $acaa$.

$j = 3$ é bom para $i = 1, \dots, 7$, pois caa é sufixo de $caabaacaa$.

Algoritmo BM – segunda versão

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$ é sufixo do mais longo dos dois.

Algoritmo BM – segunda versão

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$ é sufixo do mais longo dos dois.

$v_2[i]$: menor t em $0..m-1$ tal que $m-t$ é bom para i .
(se não há j bom para i , então $v_2[i] = m$)

Algoritmo BM – segunda versão

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$ é sufixo do mais longo dos dois.

$v_2[i]$: menor t em $0..m-1$ tal que $m-t$ é bom para i .
(se não há j bom para i , então $v_2[i] = m$)

Exemplos:

1	2	3	4	5	6
c	a	a	b	a	a

i	6	5	4	3	2	1
v_2	1	3	6	6	6	6

Algoritmo BM – segunda versão

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$ é sufixo do mais longo dos dois.

$v_2[i]$: menor t em $0..m-1$ tal que $m-t$ é bom para i .
(se não há j bom para i , então $v_2[i] = m$)

Exemplos:

1	2	3	4	5	6
c	a	a	b	a	a

1	2	3	4	5	6	7	8
b	a	-	b	a	.	b	a

i	6	5	4	3	2	1
v_2	1	3	6	6	6	6

i	8	7	6	5	4	3	2	1
v_2	3	3	6	6	6	6	6	6

Algoritmo BM – segunda versão

j é bom para i se o mais curto entre $P[i..m]$ e $P[1..j]$ é sufixo do mais longo dos dois.

$v_2[i]$: menor t em $0..m-1$ tal que $m-t$ é bom para i .
(se não há j bom para i , então $v_2[i] = m$)

Exemplos:

1	2	3	4	5	6
c	a	a	b	a	a

i	6	5	4	3	2	1
v_2	1	3	6	6	6	6

1	2	3	4	5	6	7	8
b	a	-	b	a	.	b	a

i	8	7	6	5	4	3	2	1
v_2	3	3	6	6	6	6	6	6

1	2	3	4	5	6	7	8	9	10	11
b	a	-	b	a	*	b	a	*	b	a

i	11	10	9	8	7	6	5	4	3	2	1
v_2	3	3	3	3	3	9	9	9	9	9	9

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: $n = 11$ $m = 4$

	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

$k = 4$

$c = 0$

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: $n = 11$ $m = 4$

	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

$$k = 4$$

$$c = 3$$

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

$$k = 4$$

$$c = 4$$

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: $n = 11$ $m = 4$

	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

$k = 9$

$c = 4$

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

$k = 9$

$c = 5$

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

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

$$c = 5$$

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
v_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: $n = 11$ $m = 5$

	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

$k = 5$

$c = 0$

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
<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

$k = 4$

$c = 2$

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

$k = 4$

$c = 3$

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

$$k = 4$$

$$c = 4$$

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

$$k = 4 + 3 = 7$$

$$c = 4$$

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

$k = 7$

$c = 4$

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>v</i> ₂	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: $n = 11$ $m = 4$

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

$k = 8$

$c = 6$

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
v_2	3	3	3	3

$$k = 8$$

$$c = 7$$

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
v_2	3	3	3	3

$$k = 8 + 3 = 11$$

$$c = 7$$

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

$k = 11$

$c = 7$

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

$k = 11$

$c = 8$

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>v₂</i>	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>v₂</i>	3	3	3	3

$k = 11$

$c = 10$

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

$k = 11$

$c = 11$

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

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

$c = 11$ 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: $n = 11$ $m = 5$

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

$k = 9$

$c = 3$

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>v₂</i>	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: $n = 11$ $m = 5$

	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

$k = 11$

$c = 9$

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>v₂</i>	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?