Algoritmos e heurísticas para comparações exata e aproximada de seqüências

Guilherme P. Telles[1], Nalvo F. de Almeida Jr.[2] e Fábio H. Viduani Martinez[2]

`gpt@icmc.usp.br, nalvo@dct.ufms.br, fhvm@dct.ufms.br`

[1]ICMC–USP e [2]DCT–CCET–UFMS

Roteiro

- Definições básicas e notação

Roteiro

- Definições básicas e notação
- Algoritmos, complexidades e aplicações

Roteiro

- Definições básicas e notação
- Algoritmos, complexidades e aplicações
  - Algoritmos para comparação exata

Roteiro

- Definições básicas e notação
- Algoritmos, complexidades e aplicações
  - Algoritmos para comparação exata
  - Algoritmos para comparação exata baseados em árvores digitais

Roteiro

- Definições básicas e notação
- Algoritmos, complexidades e aplicações
  - Algoritmos para comparação exata
  - Algoritmos para comparação exata baseados em árvores digitais
  - Algoritmos para comparação exata e aproximada baseados em programação dinâmica

Roteiro

- Definições básicas e notação
- Algoritmos, complexidades e aplicações
  - Algoritmos para comparação exata
  - Algoritmos para comparação exata baseados em árvores digitais
  - Algoritmos para comparação exata e aproximada baseados em programação dinâmica
  - Heurísticas para comparação aproximada

Roteiro

- Definições básicas e notação
- Algoritmos, complexidades e aplicações
    - Algoritmos para comparação exata
    - Algoritmos para comparação exata baseados em árvores digitais
    - Algoritmos para comparação exata e aproximada baseados em programação dinâmica
    - Heurísticas para comparação aproximada
    - Outros métodos de comparação

Roteiro

- Definições básicas e notação
- Algoritmos, complexidades e aplicações
  - Algoritmos para comparação exata
  - Algoritmos para comparação exata baseados em árvores digitais
  - Algoritmos para comparação exata e aproximada baseados em programação dinâmica
  - Heurísticas para comparação aproximada
  - Outros métodos de comparação
- Métodos para comparação de textos

Roteiro

- Definições básicas e notação
- Algoritmos, complexidades e aplicações
  - Algoritmos para comparação exata
  - Algoritmos para comparação exata baseados em árvores digitais
  - Algoritmos para comparação exata e aproximada baseados em programação dinâmica
  - Heurísticas para comparação aproximada
  - Outros métodos de comparação
- Métodos para comparação de textos
- Considerações finais

Comparação de Duas Seqüências

- Sejam s uma seqüência de n símbolos chamada **texto** e p uma seqüência de m símbolos chamada **padrão**.

Comparação de Duas Seqüências

- Sejam s uma seqüência de n símbolos chamada **texto** e p uma seqüência de m símbolos chamada **padrão**.
- p **ocorre em** s **na posição** $d+1$ se

$$s[d+1..d+m] = p[1..m]$$

para algum d, com $0 \leqslant d \leqslant n-m$, e d é chamado um **deslocamento** de p sobre s.

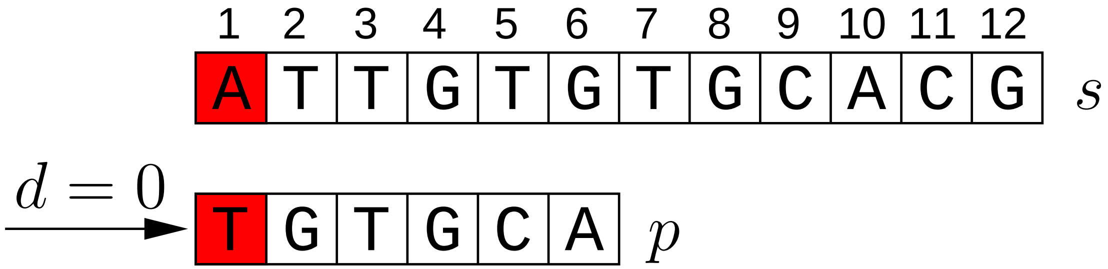
Comparação de Duas Seqüências

- Sejam s uma seqüência de n símbolos chamada **texto** e p uma seqüência de m símbolos chamada **padrão**.
- p **ocorre em** s **na posição** $d+1$ se

$$s[d+1..d+m] = p[1..m]$$

  para algum d, com $0 \leqslant d \leqslant n-m$, e d é chamado um **deslocamento** de p sobre s.
- Problema SM(s, p): dada uma seqüência s de n símbolos e uma seqüência p de m símbolos, com $m \leqslant n$, encontrar todas as ocorrências de p em s.

Força Bruta ou Ingênuo



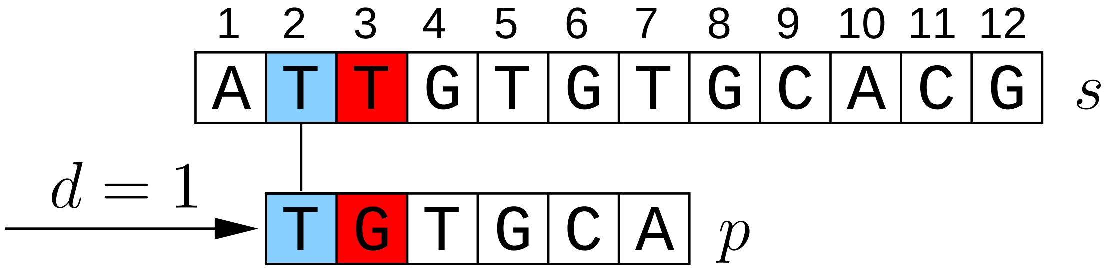
1 2 3 4 5 6 7 8 9 10 11 12

A T T G T G T G C A C G s

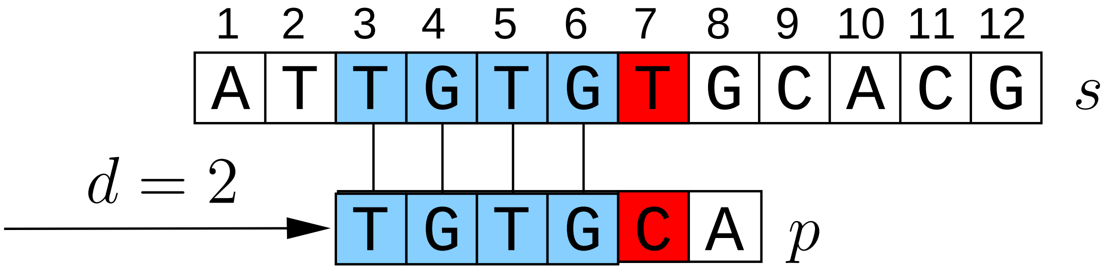
$d = 0$

T G T G C A p

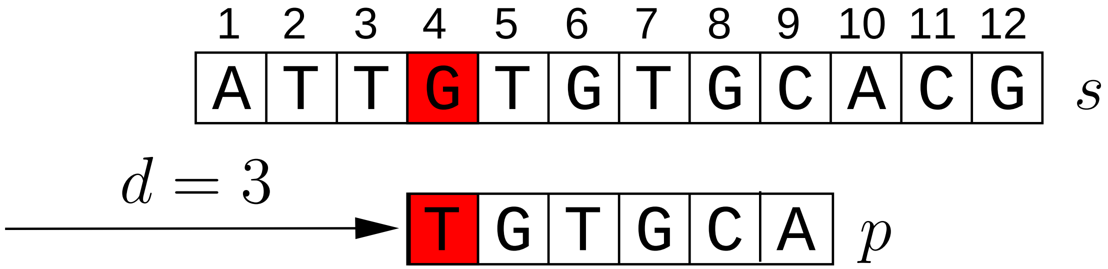
Força Bruta ou Ingênuo



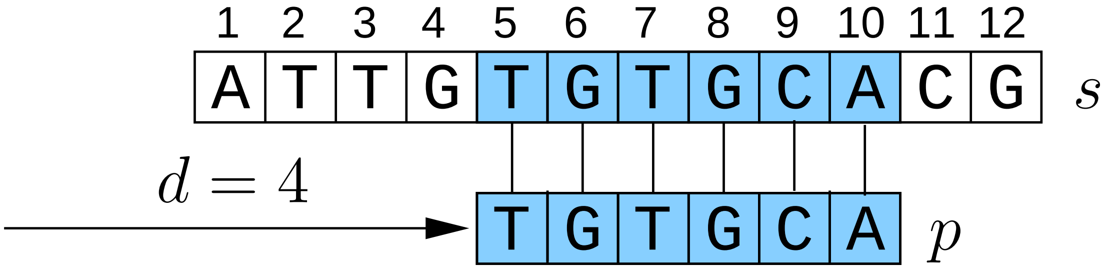
Força Bruta ou Ingênuo



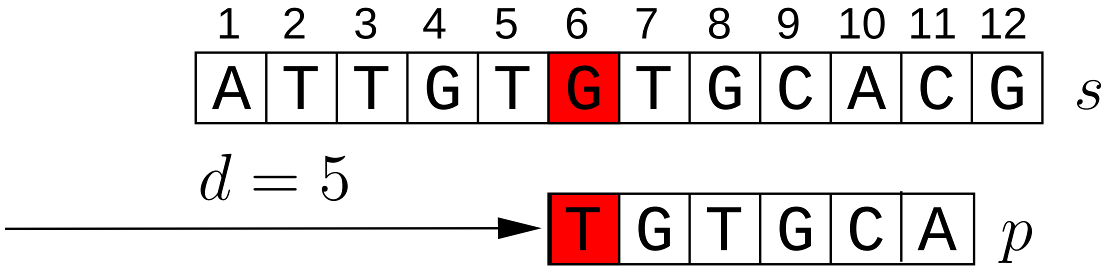
Força Bruta ou Ingênuo



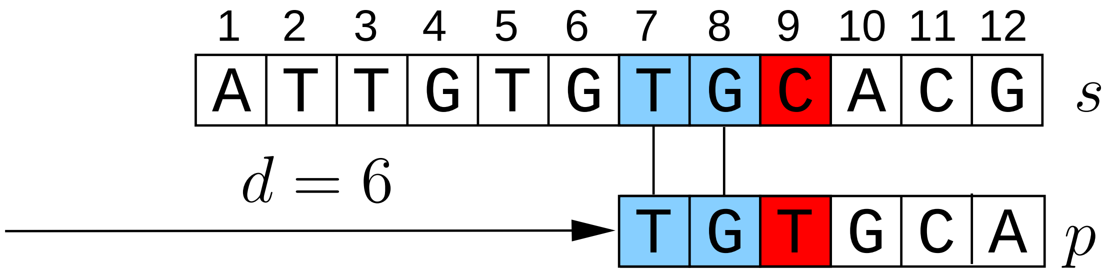
Força Bruta ou Ingênuo



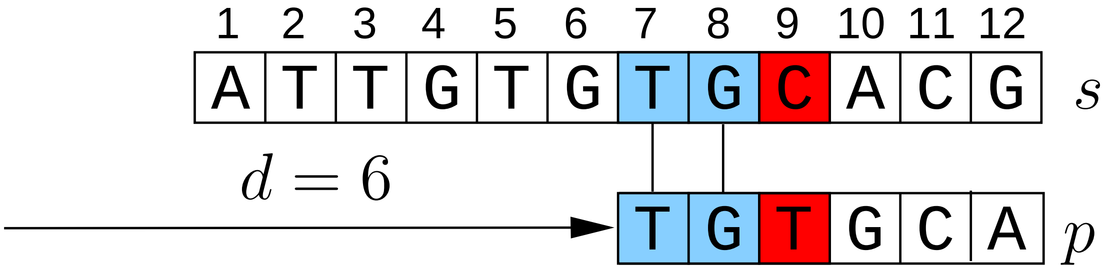
Força Bruta ou Ingênuo



Força Bruta ou Ingênuo

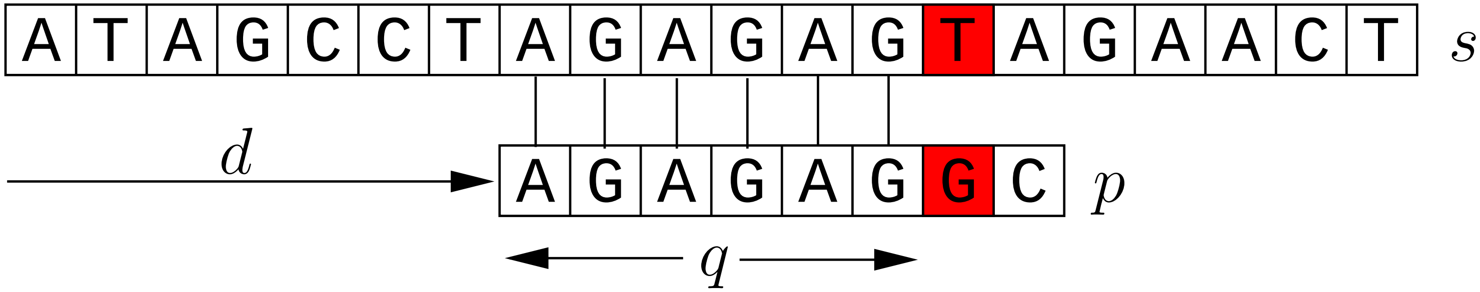


Força Bruta ou Ingênuo

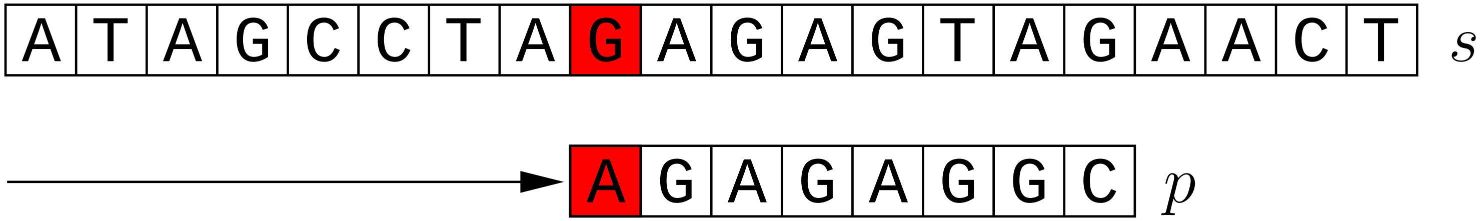


$$O(mn)$$

Knuth, Morris e Pratt



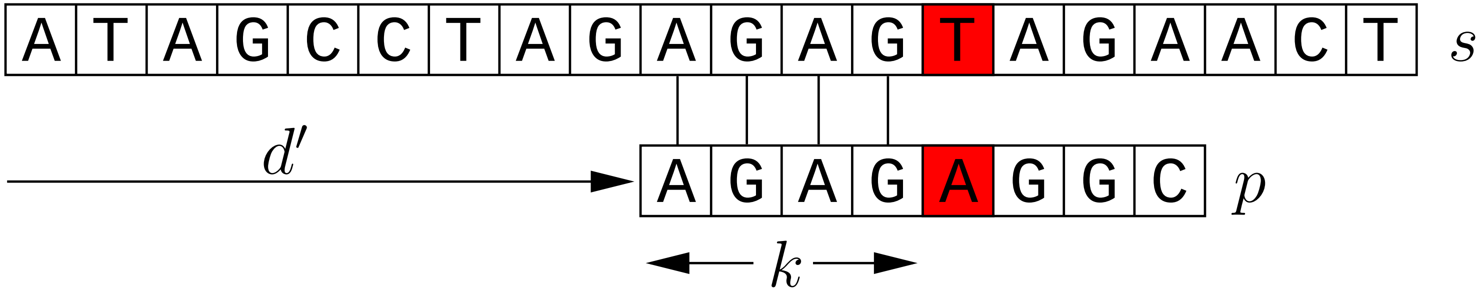
Knuth, Morris e Pratt



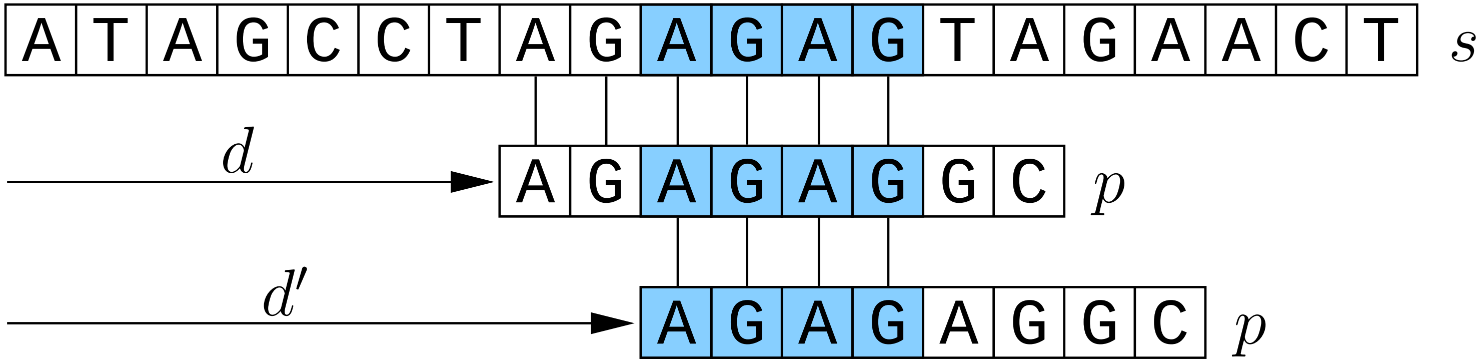
A T A G C C T A G A G A G T A G A A C T s

A G A G A G G C p

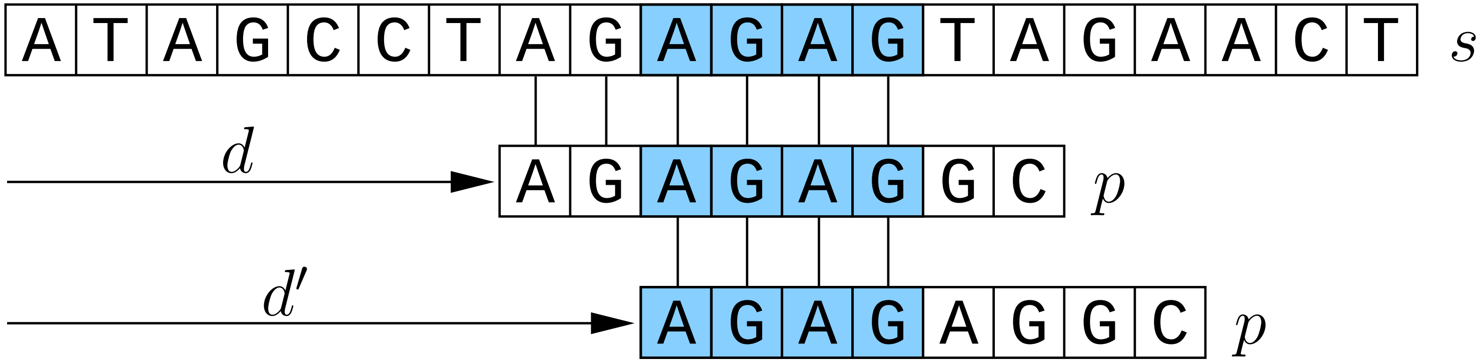
Knuth, Morris e Pratt



Knuth, Morris e Pratt



Knuth, Morris e Pratt

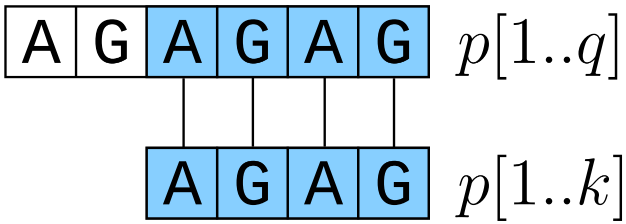


Dado que os símbolos $p[1..q]$ do padrão coincidem com os símbolos $s[d+1..d+q]$ do texto, qual o menor deslocamento $d' > d$ tal que

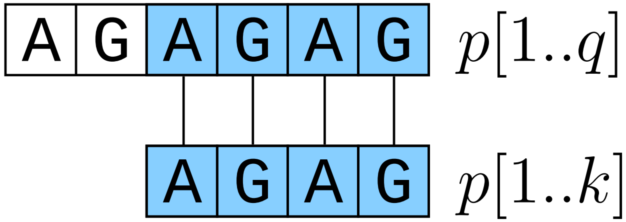
$$p[1..k] = s[d'+1..d'+k],$$

onde $d'+k = d+q$?

Knuth, Morris e Pratt



Knuth, Morris e Pratt



Queremos encontrar o maior $k < q$ tal que $p[1..k]$ é um sufixo de $p[1..q]$. Assim, o próximo deslocamento a ser verificado é $d' = d + (q - k)$.

Knuth, Morris e Pratt

Dado um padrão p de comprimento m, a **função prefixo** π é tal que:

$$\pi\colon \{1,\ldots,m\} \to \{0,\ldots,m-1\} \text{ e}$$

Knuth, Morris e Pratt

Dado um padrão p de comprimento m, a **função prefixo** π é tal que:

$$\pi\colon \{1,\ldots,m\} \to \{0,\ldots,m-1\} \text{ e}$$

$$\pi(q) = \max\{k\colon k < q \text{ e } p[1..k] \text{ é sufixo de } p[1..q]\}.$$

Knuth, Morris e Pratt

FUNÇÃO-PREFIXO(p): recebe uma seqüência p de m símbolos e devolve a função prefixo π para p.

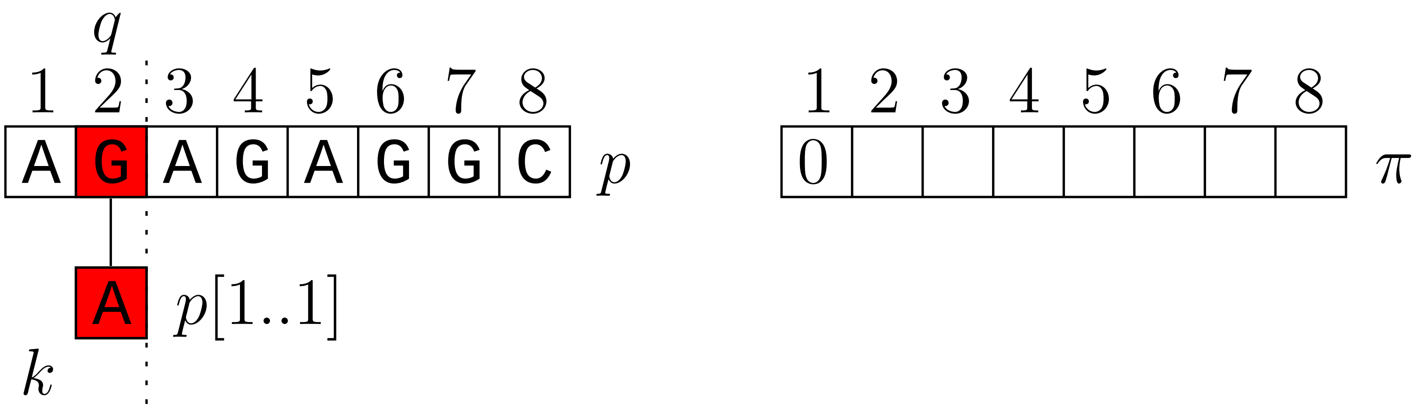
```
$\pi[1] \leftarrow 0$
$k \leftarrow 0$
para $q \leftarrow 2$ até $m$ faça
    enquanto $k > 0$ e $p[k+1] \neq p[q]$ faça
        $k \leftarrow \pi[k]$
    se $p[k+1] = p[q]$ então
        $k \leftarrow k+1$
    $\pi[q] \leftarrow k$
devolva $\pi$
```

Knuth, Morris e Pratt

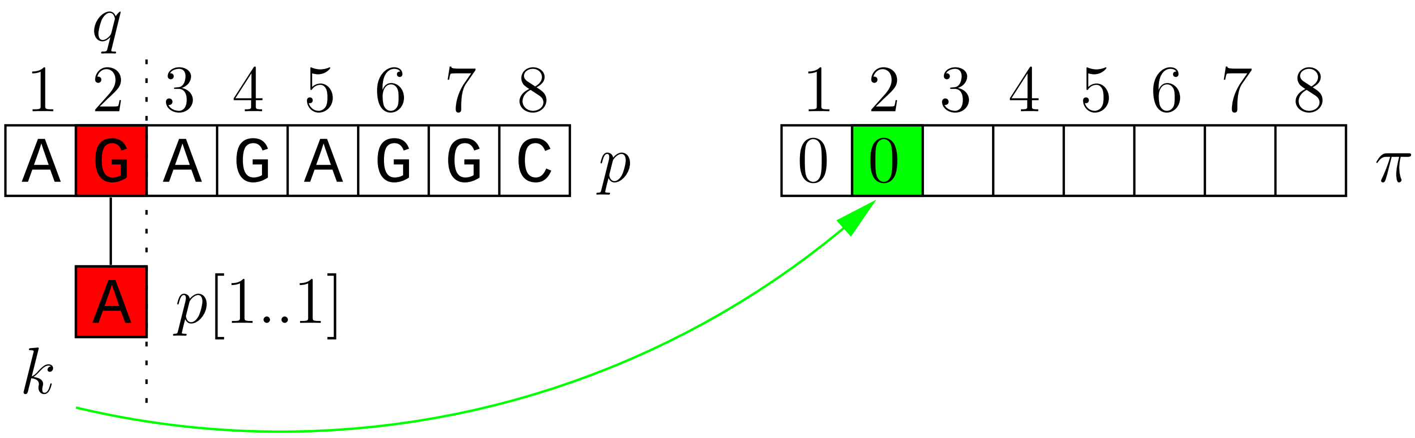
1	2	3	4	5	6	7	8	
A	G	A	G	A	G	G	C	p

1	2	3	4	5	6	7	8	
0								π

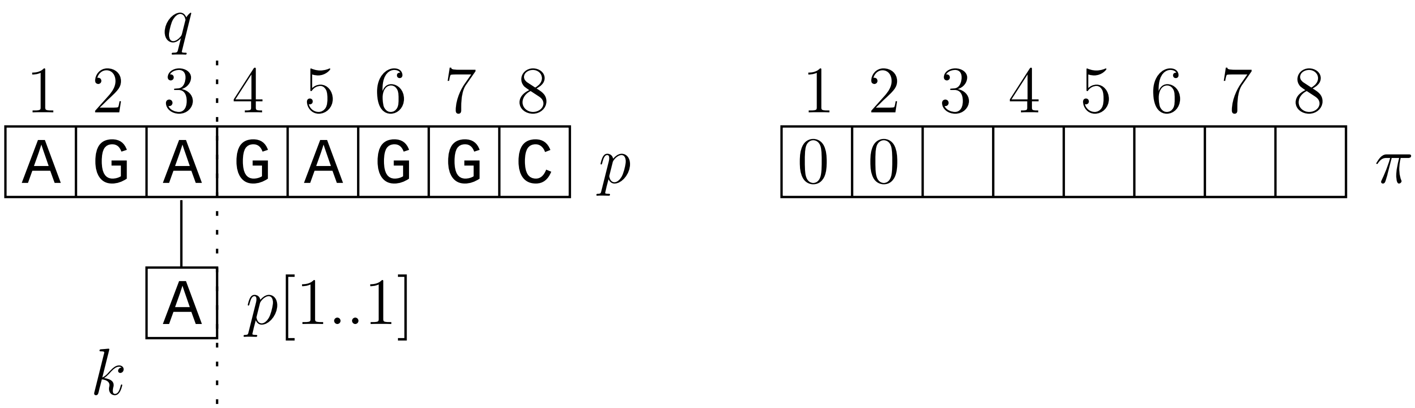
Knuth, Morris e Pratt



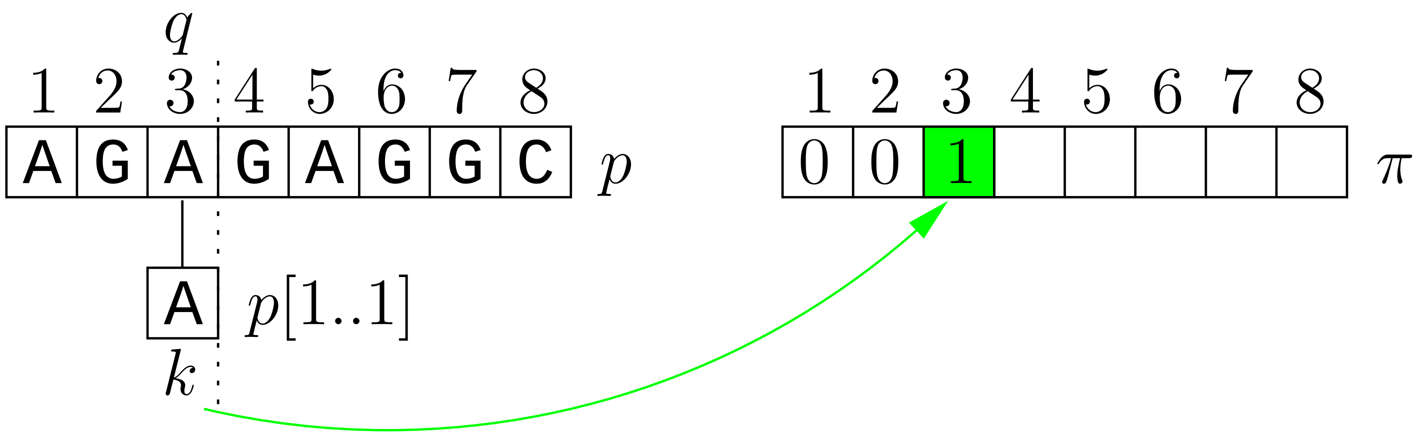
Knuth, Morris e Pratt



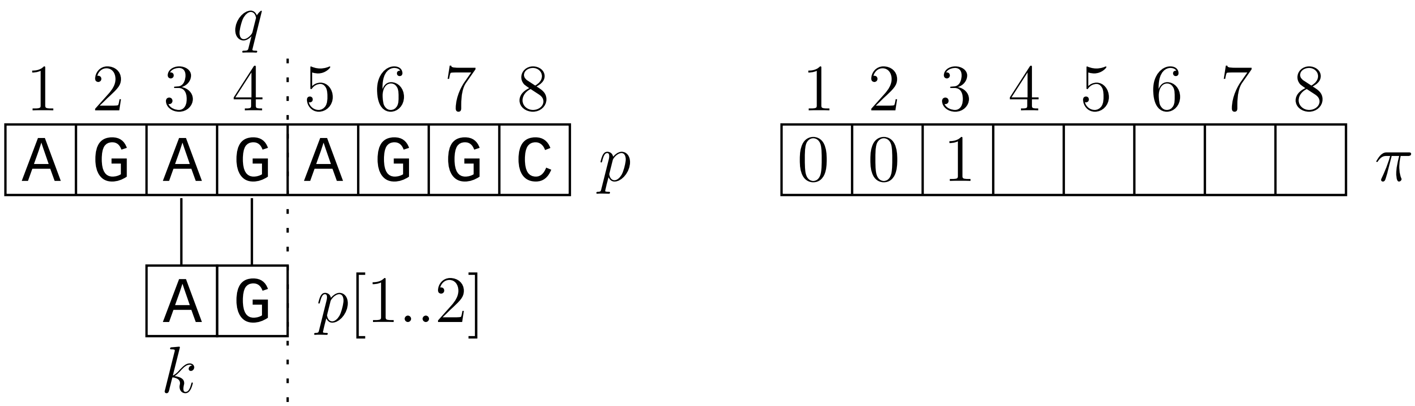
Knuth, Morris e Pratt



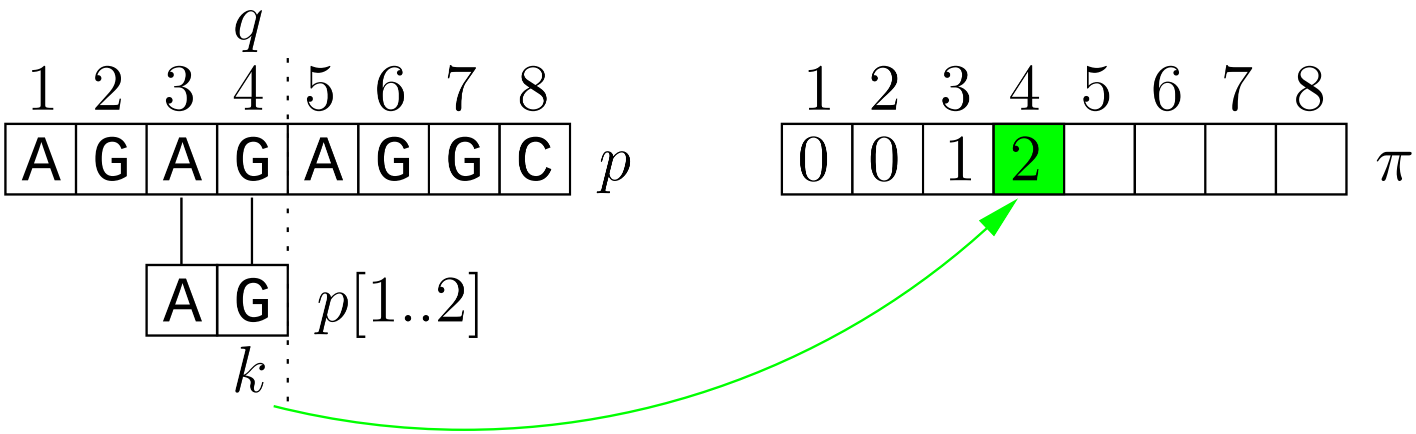
Knuth, Morris e Pratt



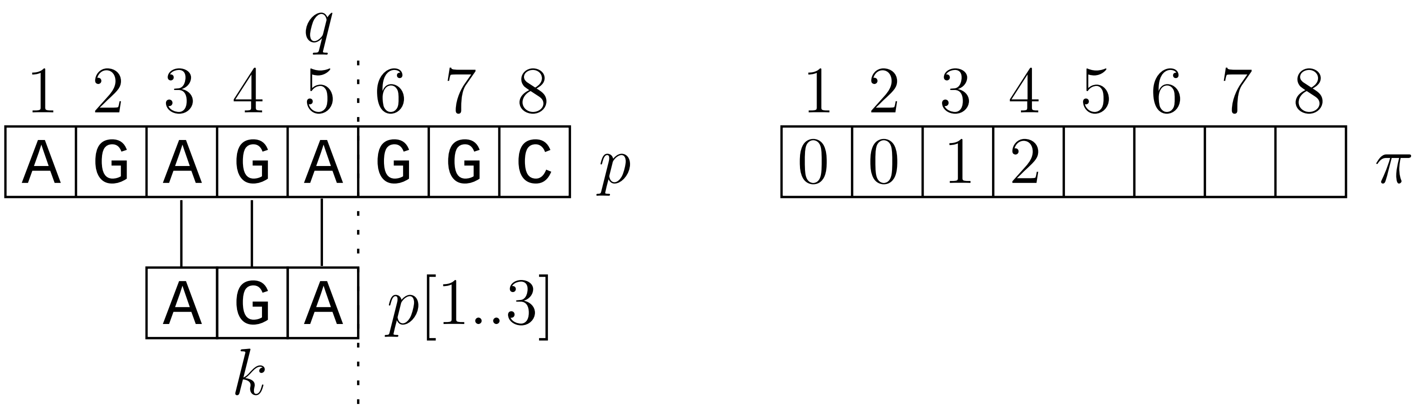
Knuth, Morris e Pratt



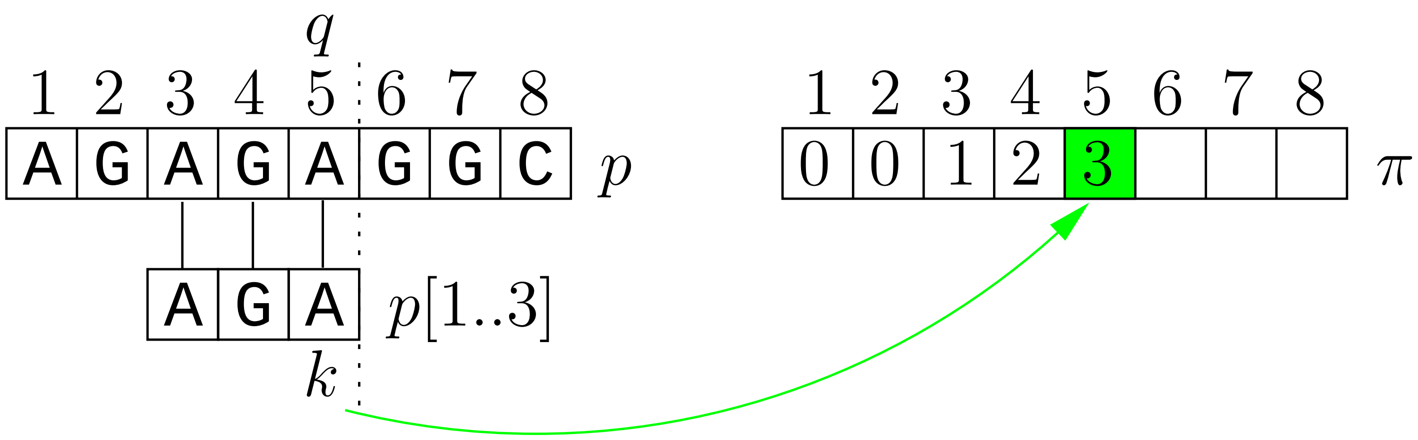
Knuth, Morris e Pratt



Knuth, Morris e Pratt



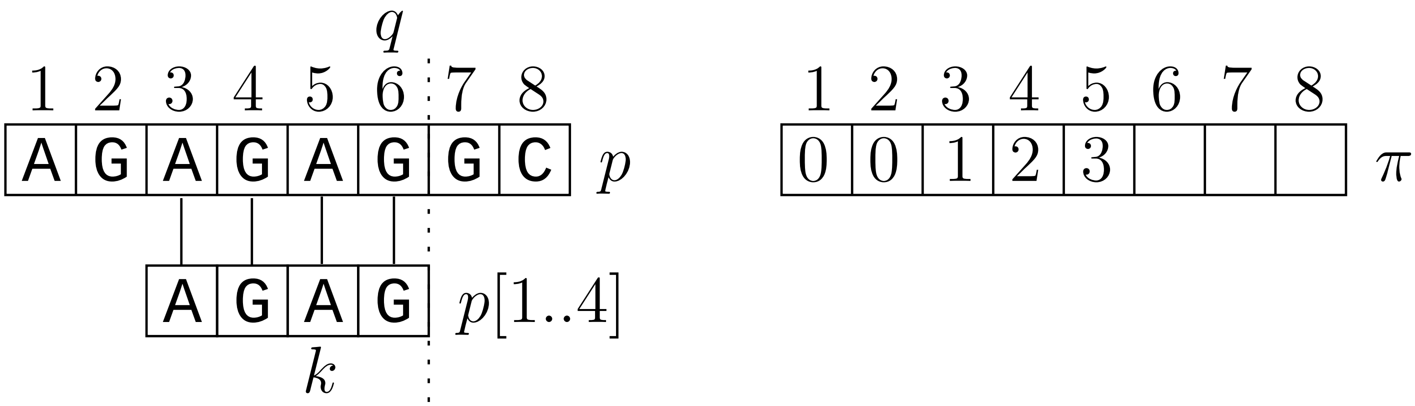
Knuth, Morris e Pratt



	1	2	3	4	5	6	7	8
p	A	G	A	G	A	G	G	C

	1	2	3	4	5	6	7	8
π	0	0	1	2	🟩 3			

Knuth, Morris e Pratt

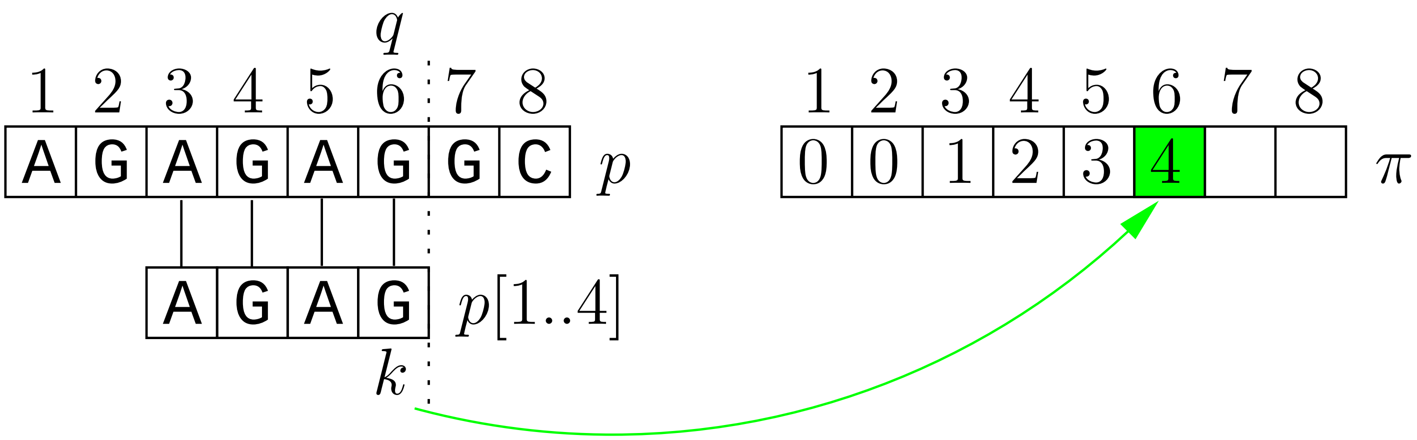


1	2	3	4	5	6	7	8
A	G	A	G	A	G	G	C

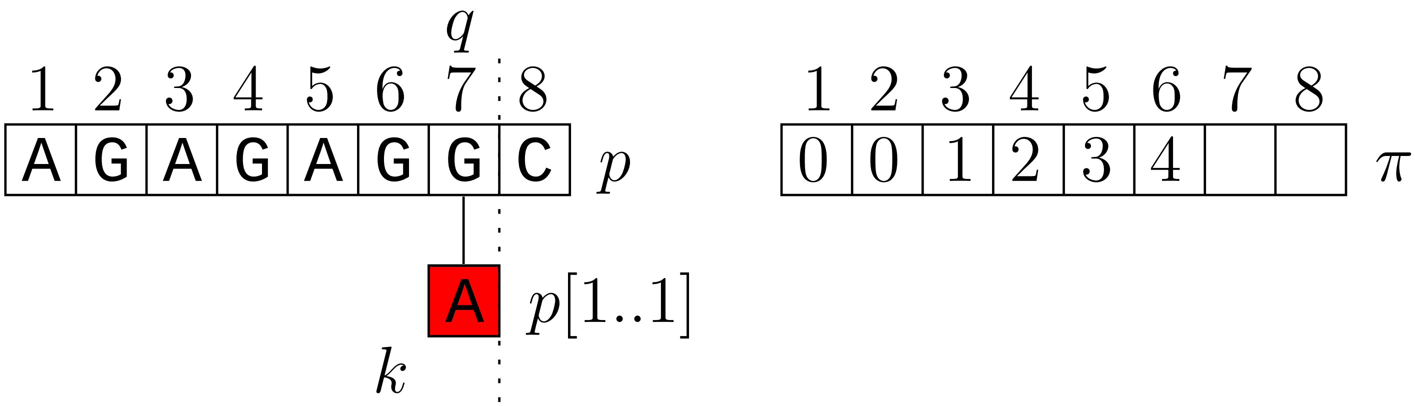
A	G	A	G

1	2	3	4	5	6	7	8
0	0	1	2	3			

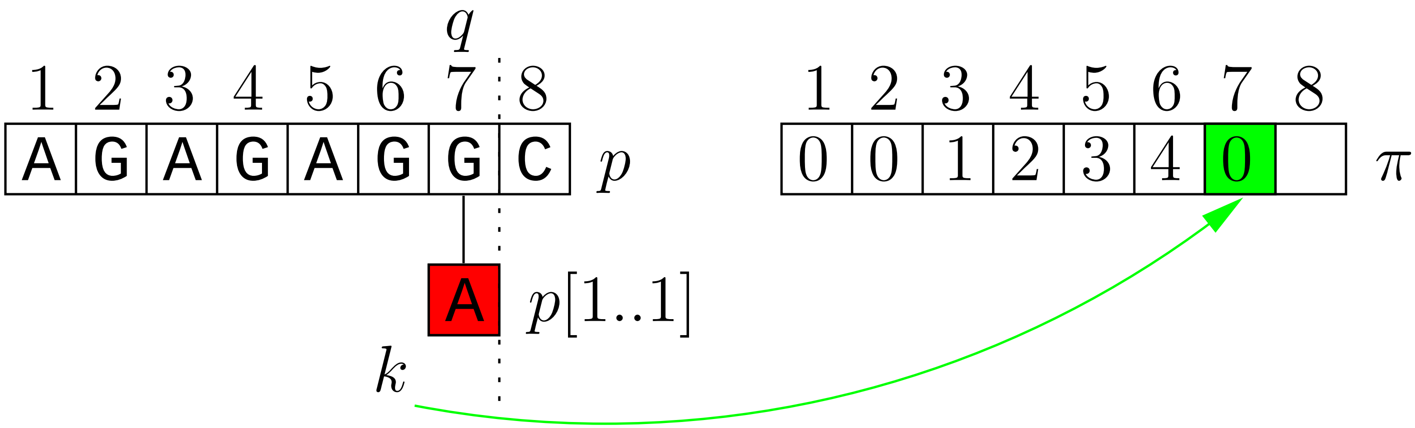
Knuth, Morris e Pratt



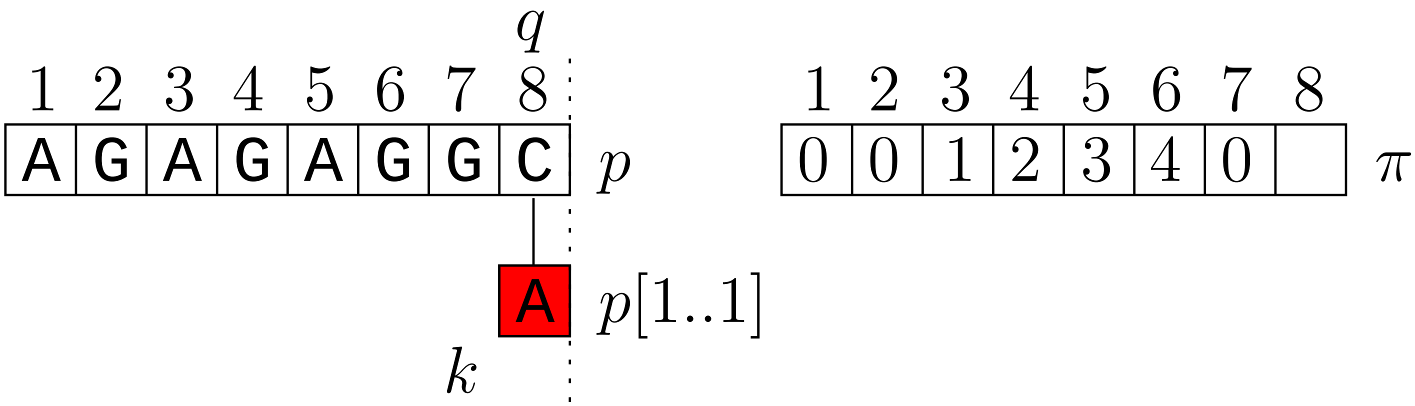
Knuth, Morris e Pratt



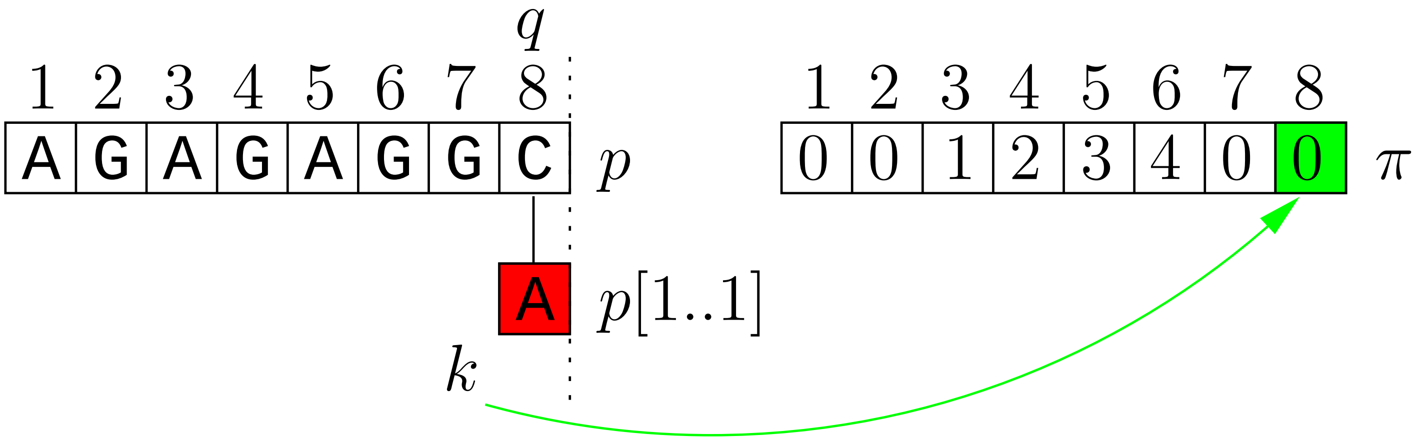
Knuth, Morris e Pratt



Knuth, Morris e Pratt



Knuth, Morris e Pratt



Knuth, Morris e Pratt

1	2	3	4	5	6	7	8	
A	G	A	G	A	G	G	C	p

1	2	3	4	5	6	7	8	
0	0	1	2	3	4	0	0	π

Tempo de execução: $O(m)$

Knuth, Morris e Pratt

KMP(s, p): recebe um texto s de n símbolos e um padrão p de m símbolos e realiza a comparação de s e p, devolvendo os índices em s onde p ocorre.

```
$\pi \leftarrow$ FUNÇÃO-PREFIXO($p$)
$q \leftarrow 0$
para $i \leftarrow 1$ até $n$ faça
    enquanto $q > 0$ e $p[q+1] \neq s[i]$ faça
        $q \leftarrow \pi[q]$
    se $p[q+1] = s[i]$ então
        $q \leftarrow q + 1$
    se $q = m$ então
        escreva “Padrão ocorre no texto com deslocamento” $i - m$
        $q \leftarrow \pi[q]$
```

Knuth, Morris e Pratt

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C	s

1	2	3	4	5	6	7	8	
A	G	A	G	A	G	G	C	p

Knuth, Morris e Pratt

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C	s

1	2	3	4	5	6	7	8	
0	0	1	2	3	4	0	0	π

A	G	A	G	A	G	G	C	p
1	2	3	4	5	6	7	8	

Knuth, Morris e Pratt

i

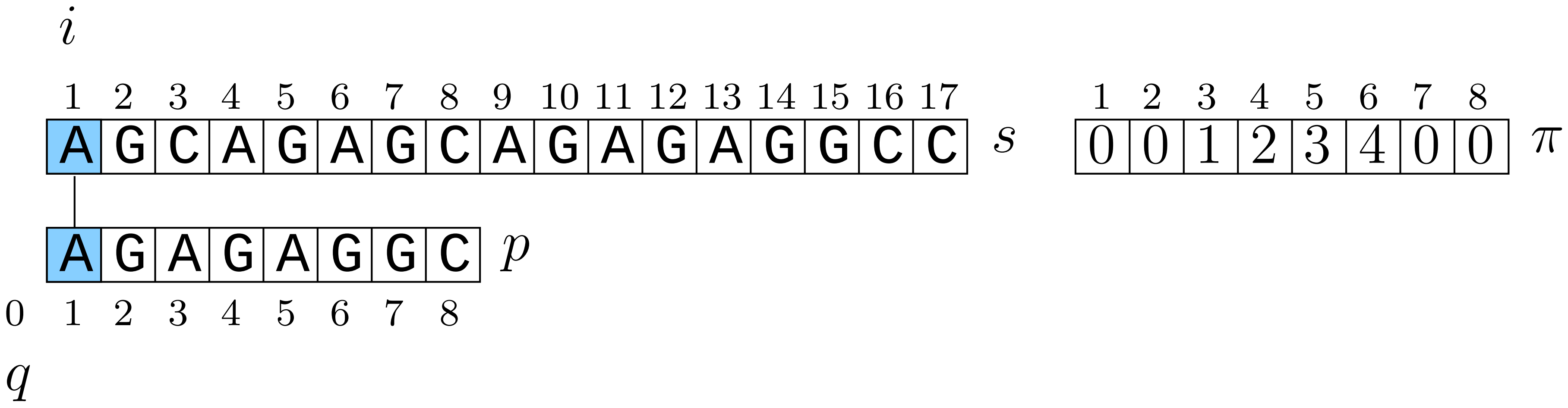
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C	s

1	2	3	4	5	6	7	8	
0	0	1	2	3	4	0	0	π

	A	G	A	G	A	G	G	C	p
0	1	2	3	4	5	6	7	8	

q

Knuth, Morris e Pratt



i

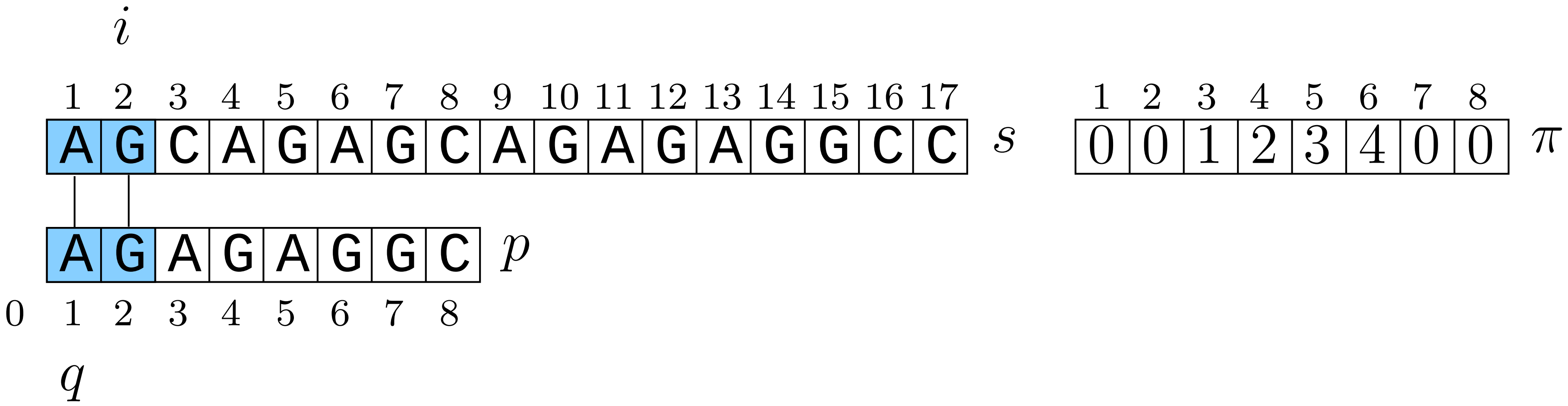
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C	s

1	2	3	4	5	6	7	8	
0	0	1	2	3	4	0	0	π

	A	G	A	G	A	G	G	C	p
0	1	2	3	4	5	6	7	8	

q

Knuth, Morris e Pratt

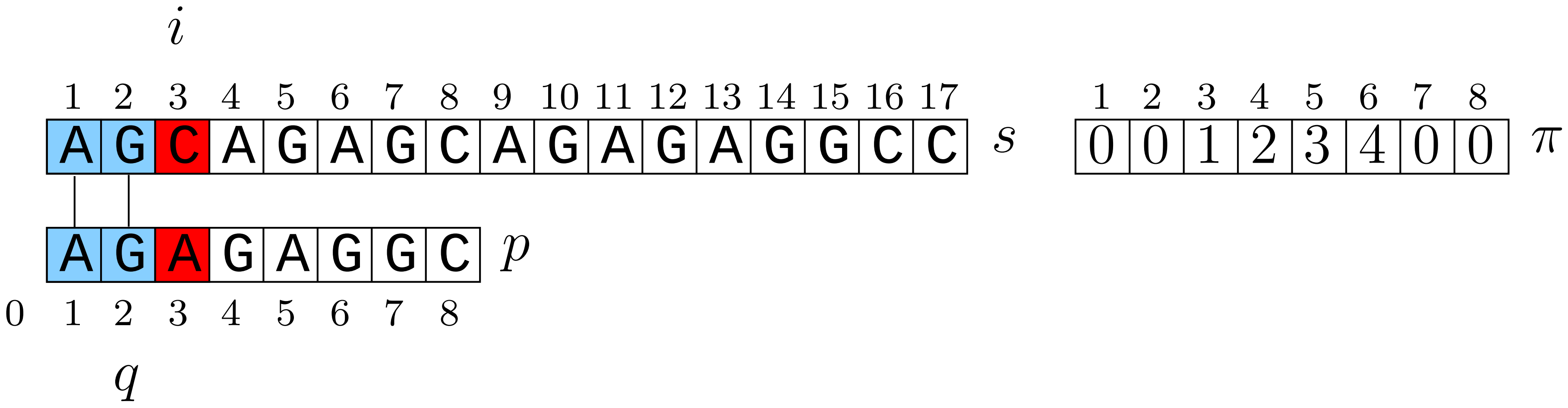


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

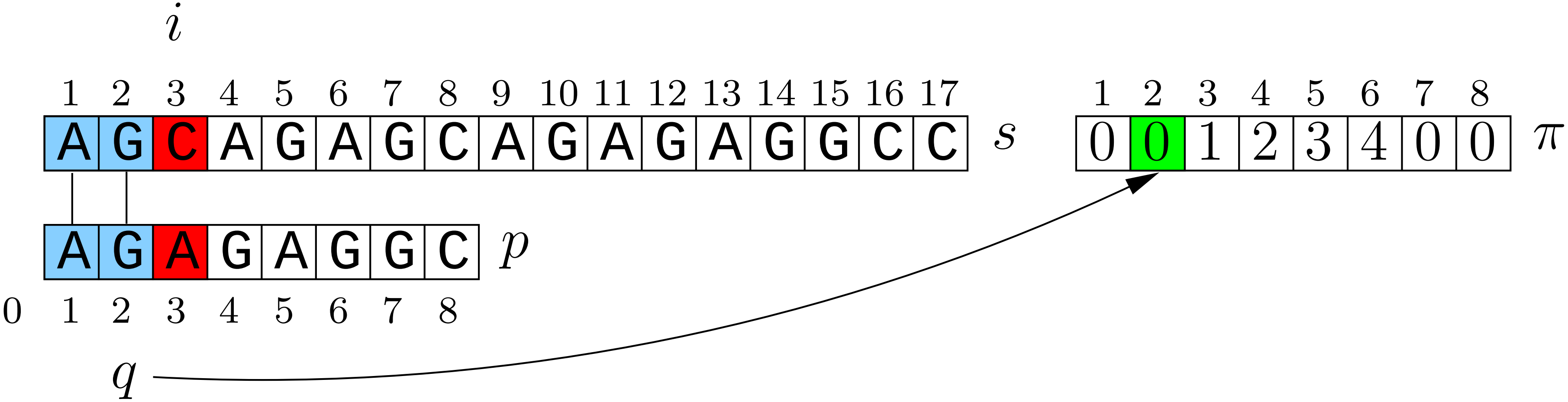
1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

0	1	2	3	4	5	6	7	8
	A	G	A	G	A	G	G	C

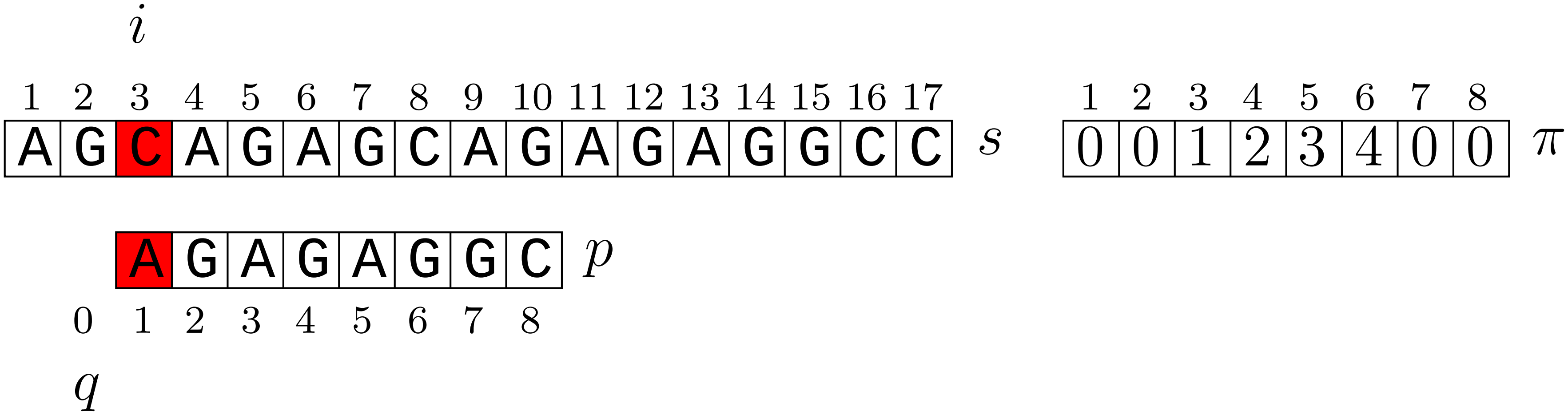
Knuth, Morris e Pratt



Knuth, Morris e Pratt



Knuth, Morris e Pratt



i

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

s

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

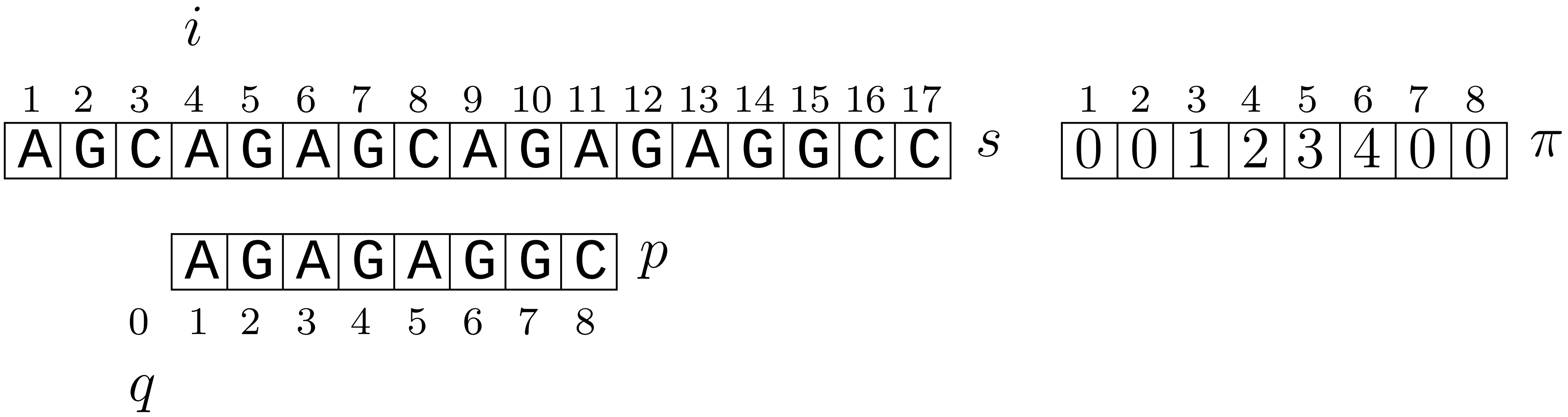
π

	A	G	A	G	A	G	G	C
0	1	2	3	4	5	6	7	8

p

q

Knuth, Morris e Pratt

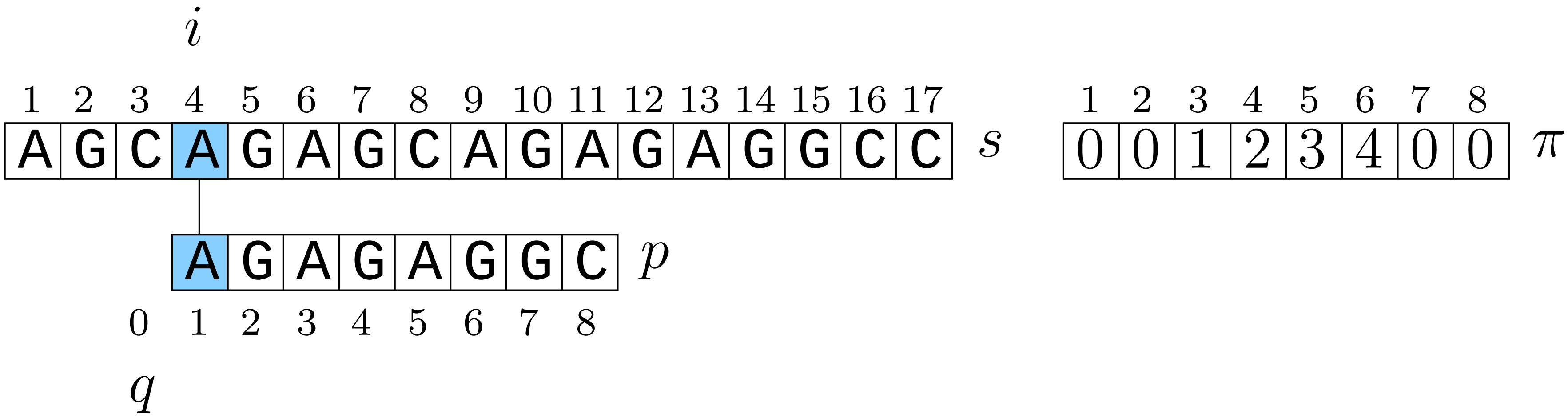


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

0	1	2	3	4	5	6	7	8
	A	G	A	G	A	G	G	C

Knuth, Morris e Pratt

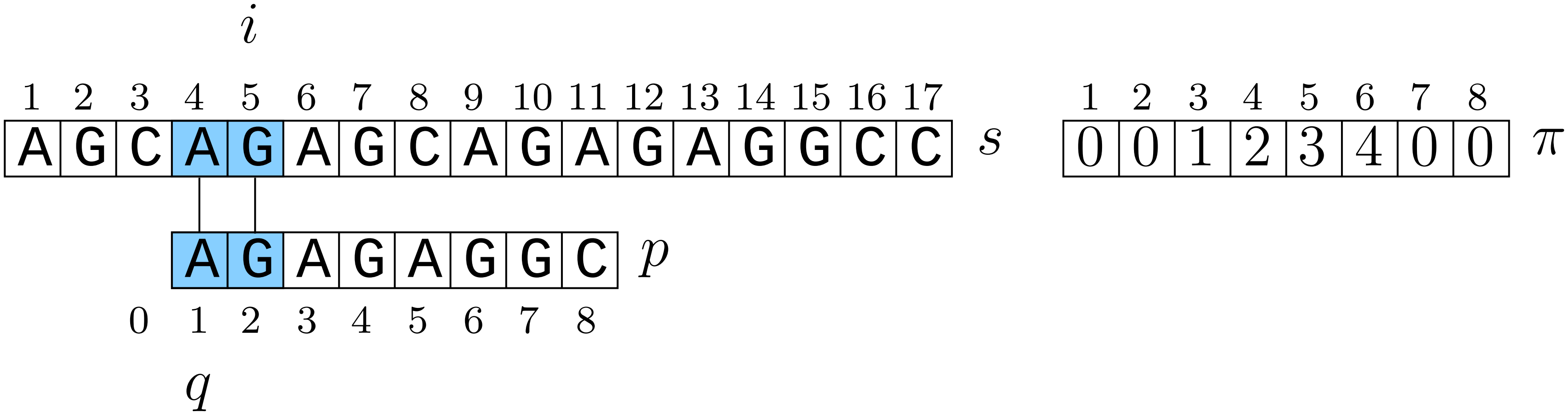


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

0	1	2	3	4	5	6	7	8
	A	G	A	G	A	G	G	C

Knuth, Morris e Pratt

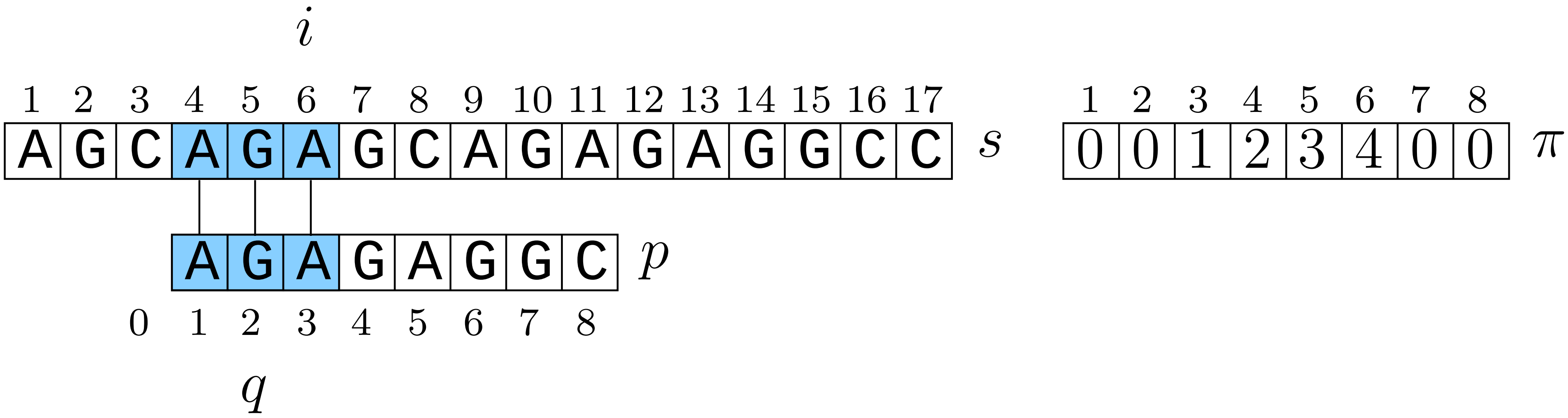


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

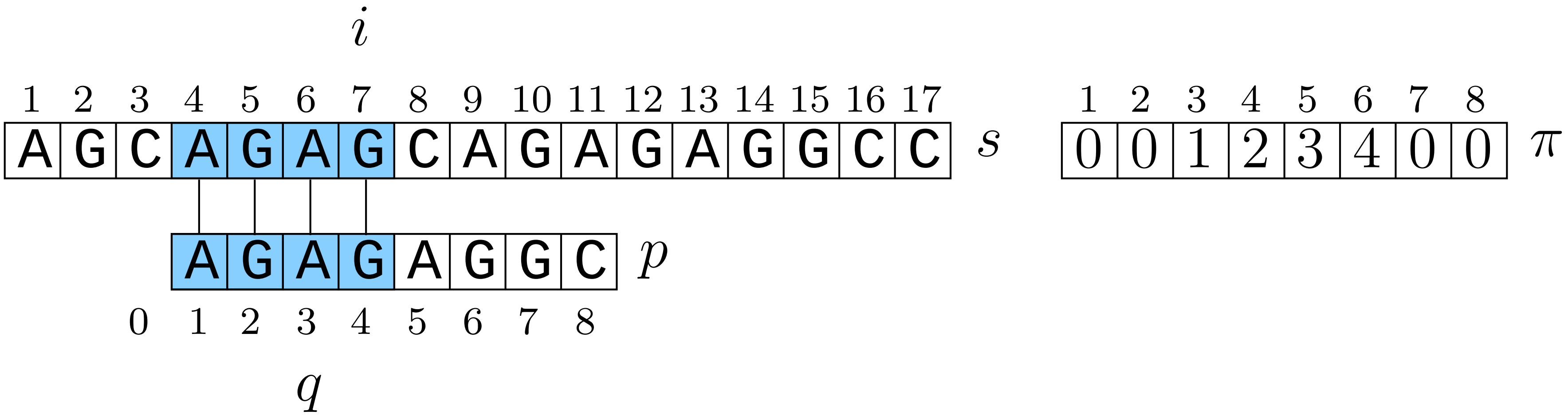
1	2	3	4	5	6	7	8
A	G	A	G	A	G	G	C

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

Knuth, Morris e Pratt



Knuth, Morris e Pratt

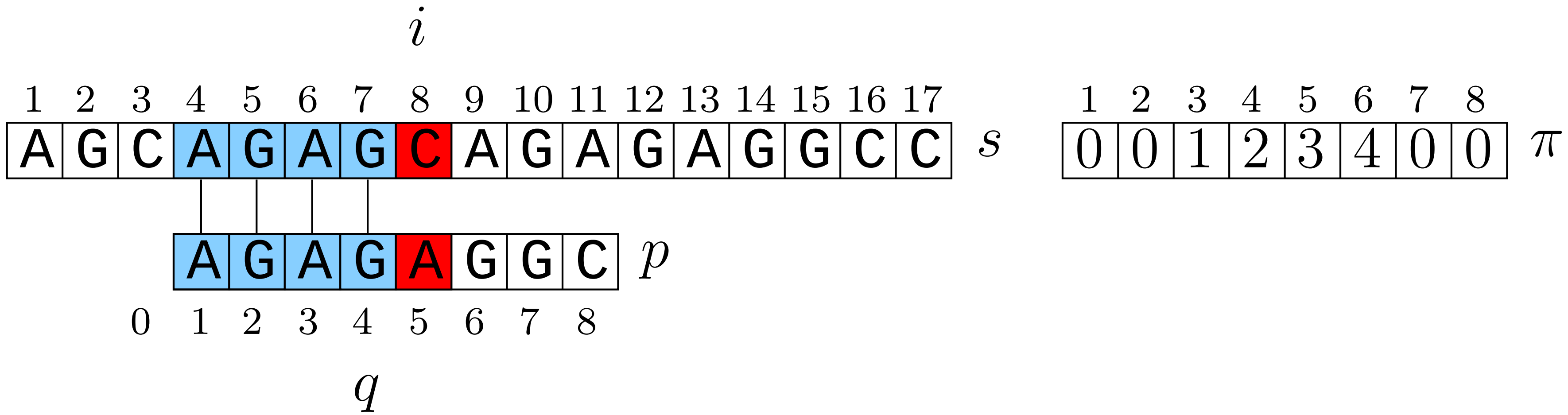


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
s	A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

	0	1	2	3	4	5	6	7	8
p		A	G	A	G	A	G	G	C

	1	2	3	4	5	6	7	8
π	0	0	1	2	3	4	0	0

Knuth, Morris e Pratt

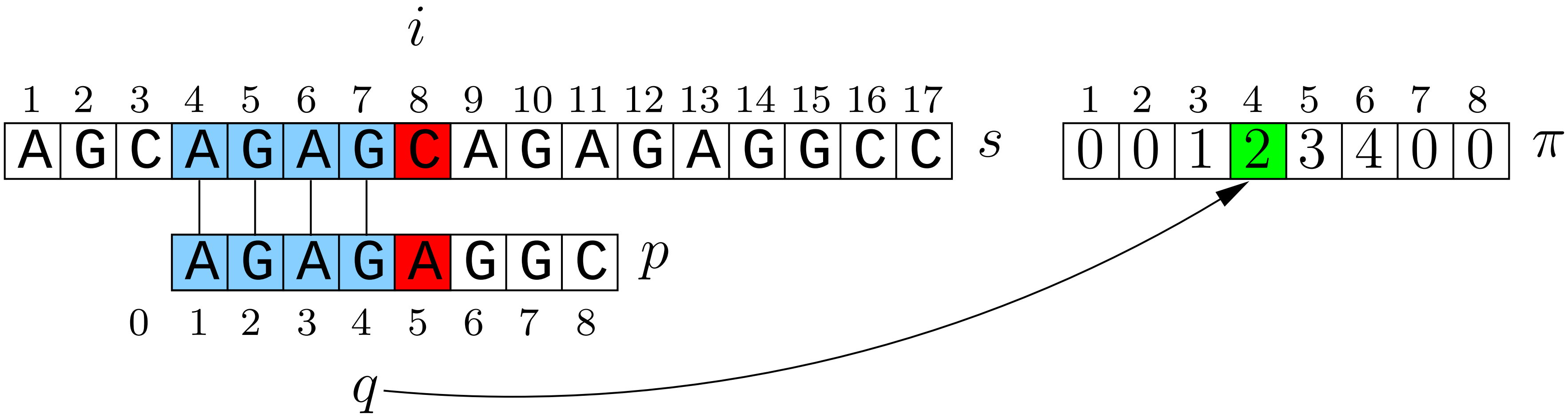


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

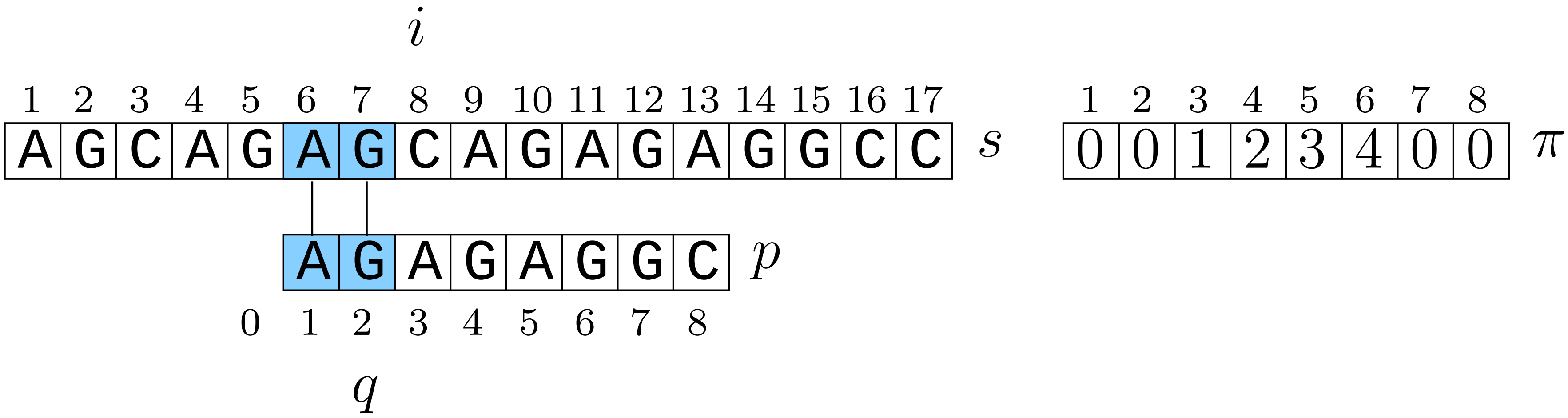
A	G	A	G	A	G	G	C

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

Knuth, Morris e Pratt



Knuth, Morris e Pratt

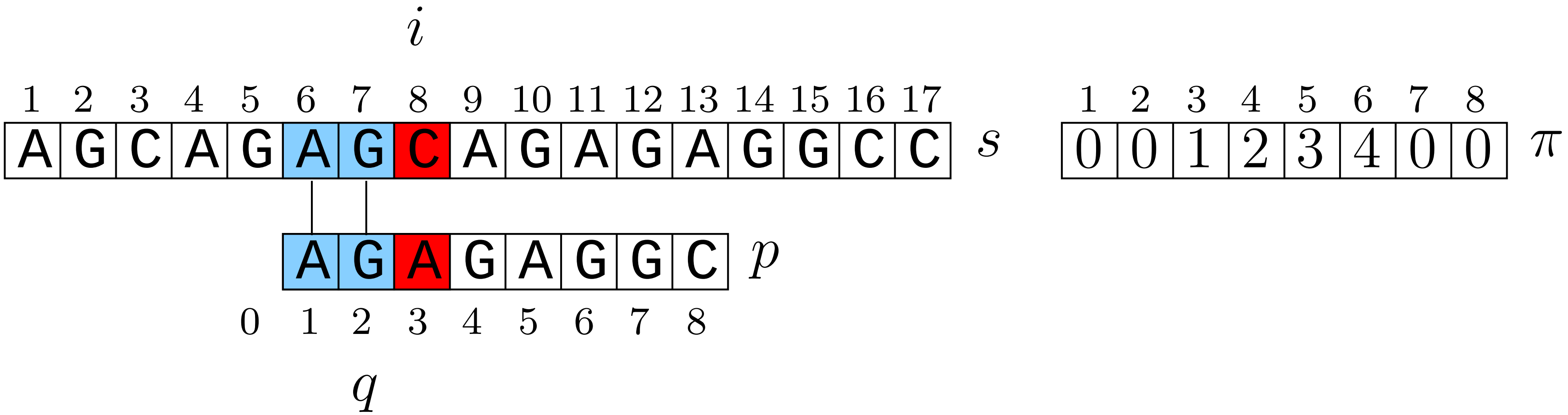


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

0	1	2	3	4	5	6	7	8
	A	G	A	G	A	G	G	C

Knuth, Morris e Pratt



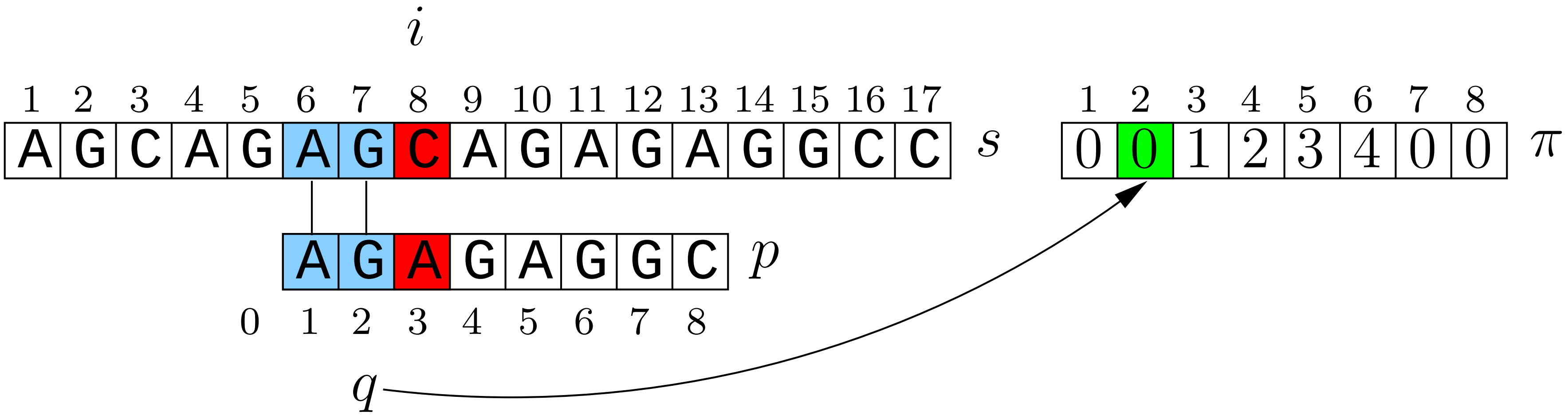
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

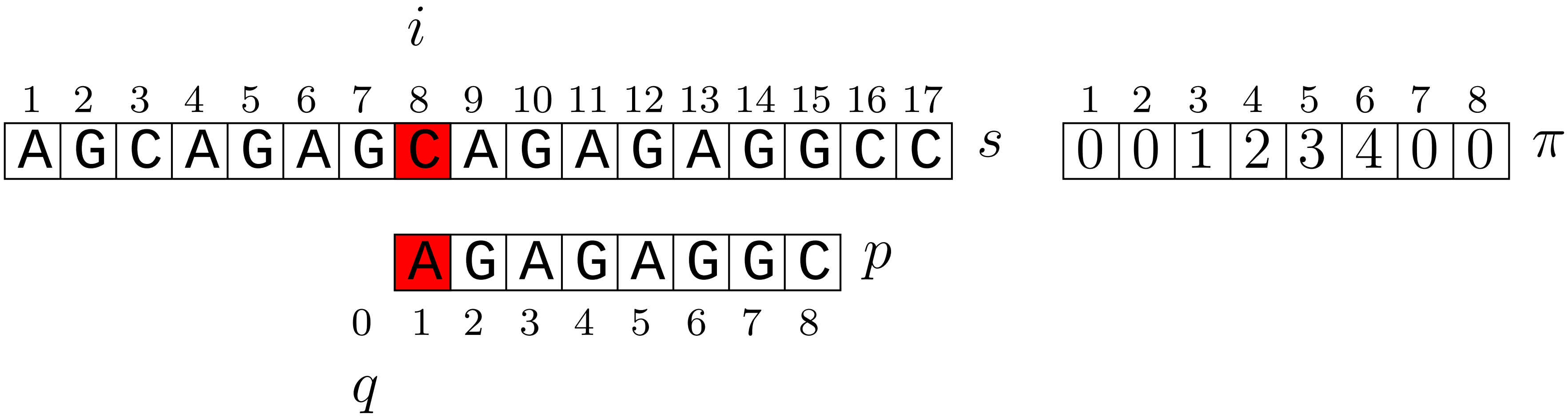
A	G	A	G	A	G	G	C

0 1 2 3 4 5 6 7 8

Knuth, Morris e Pratt



Knuth, Morris e Pratt

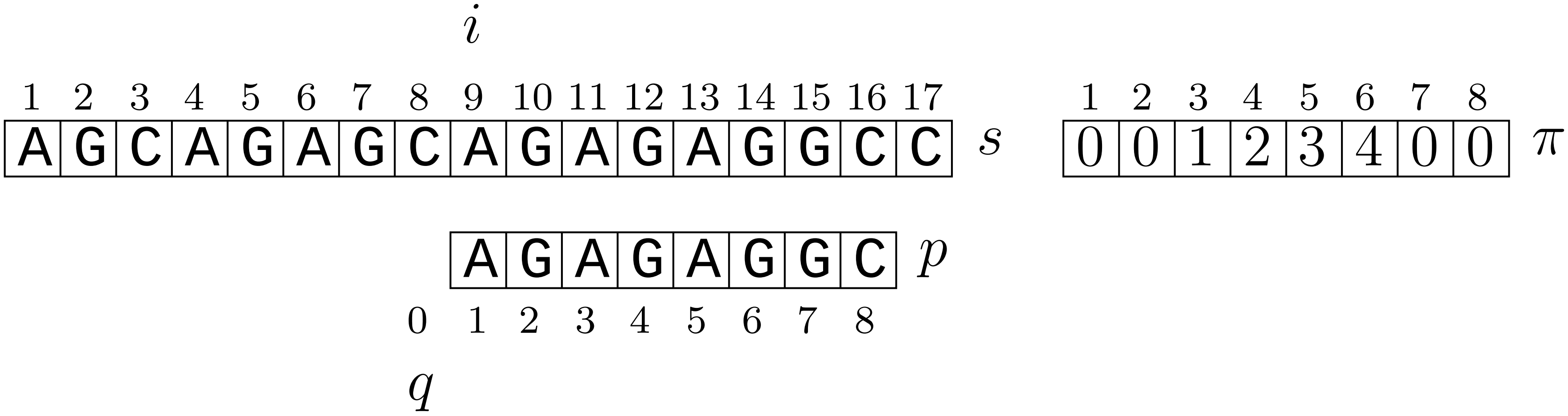


							i										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	A	G	A	G	🟥 C	A	G	A	G	A	G	G	C	C	*s*

1	2	3	4	5	6	7	8	
0	0	1	2	3	4	0	0	π

	🟥 A	G	A	G	A	G	G	C	*p*
0	1	2	3	4	5	6	7	8	
q									

Knuth, Morris e Pratt



i

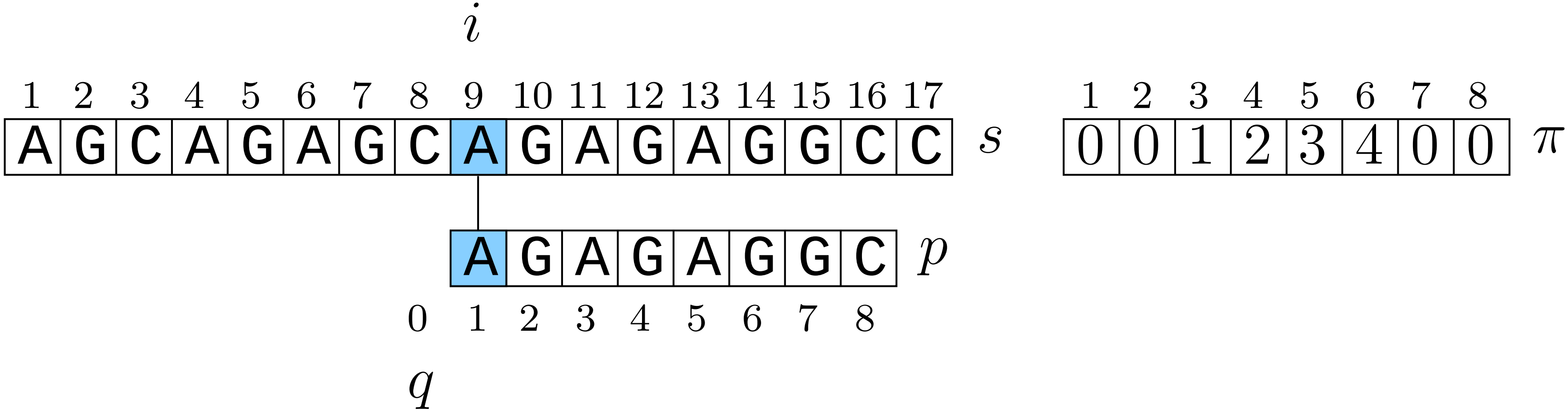
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C	s

1	2	3	4	5	6	7	8	
0	0	1	2	3	4	0	0	π

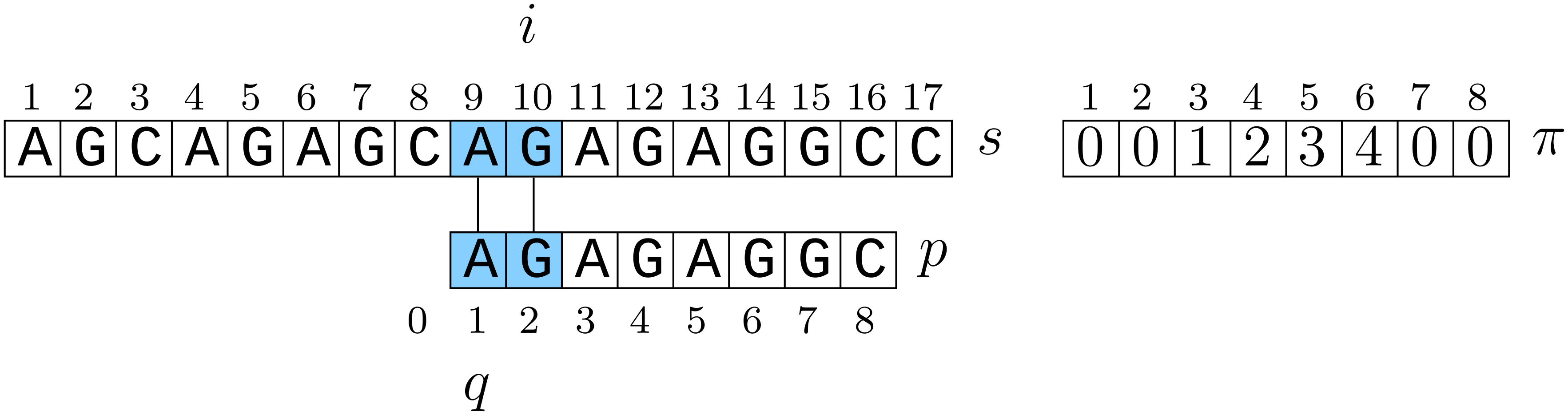
	A	G	A	G	A	G	G	C	p
0	1	2	3	4	5	6	7	8	

q

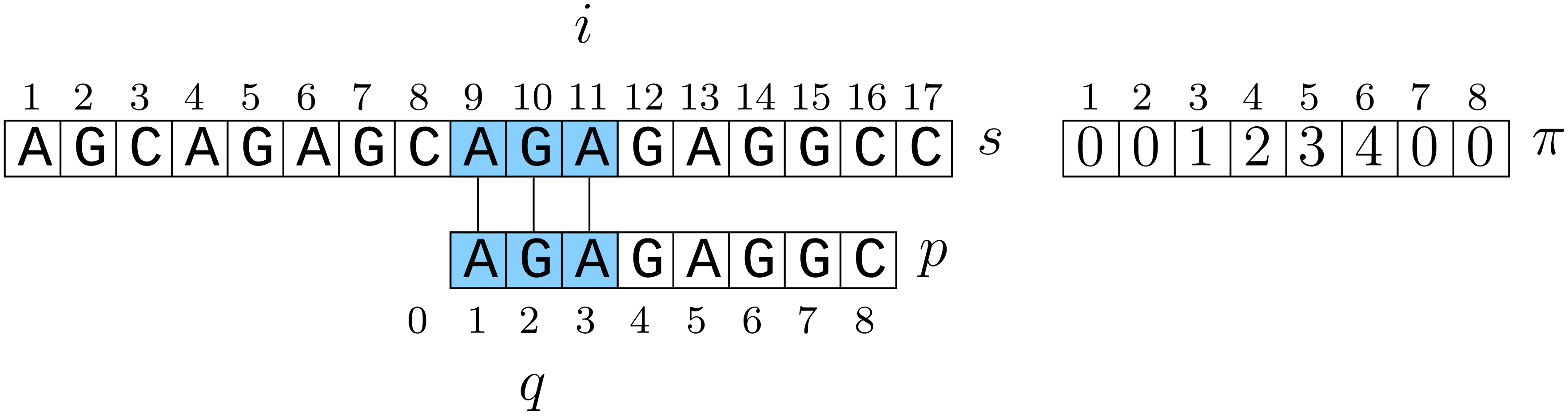
Knuth, Morris e Pratt



Knuth, Morris e Pratt



Knuth, Morris e Pratt

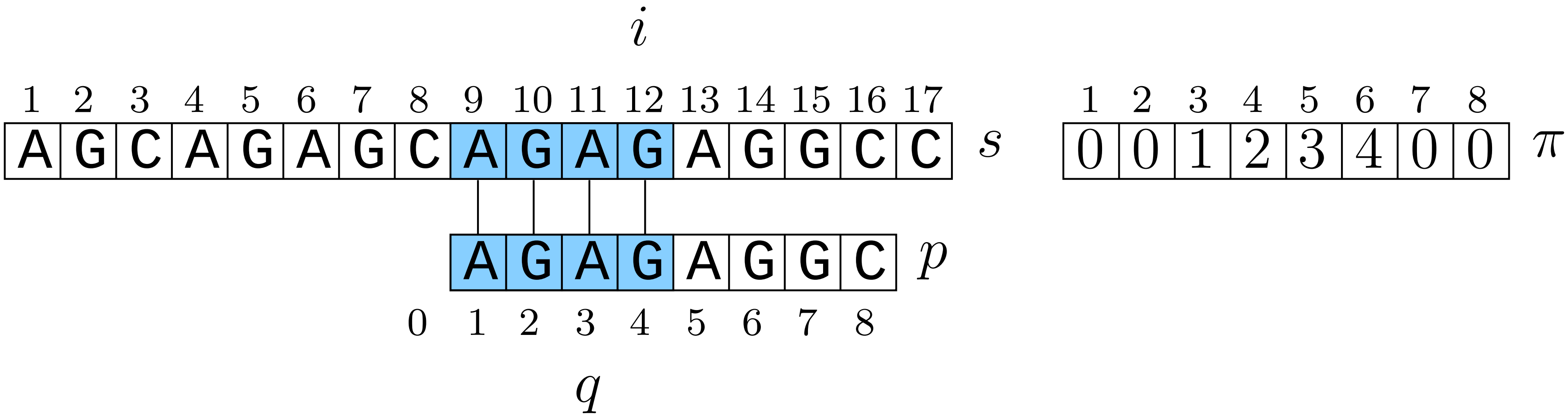


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

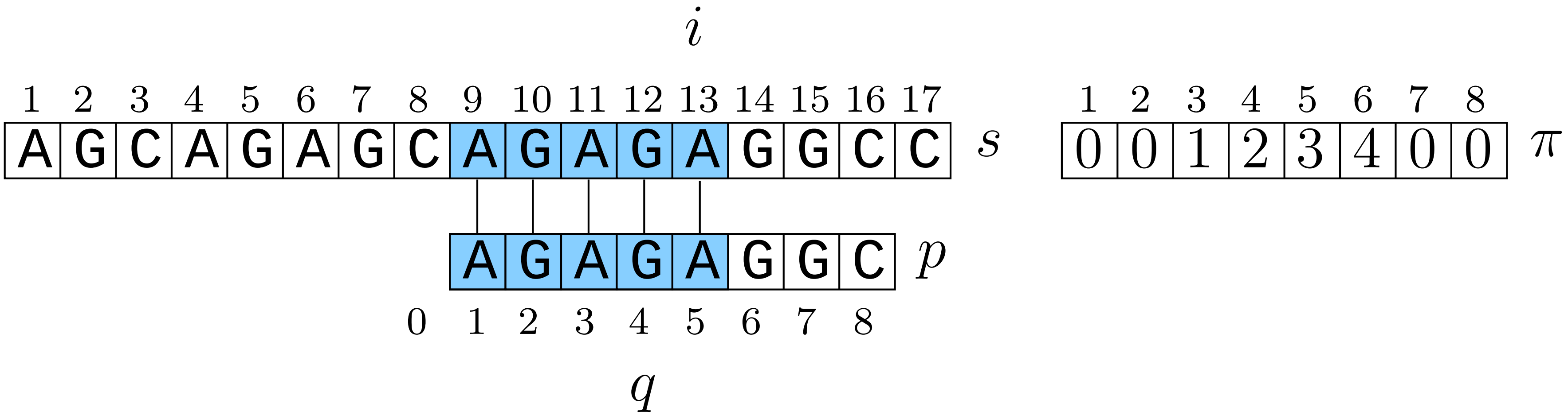
0	1	2	3	4	5	6	7	8
	A	G	A	G	A	G	G	C

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

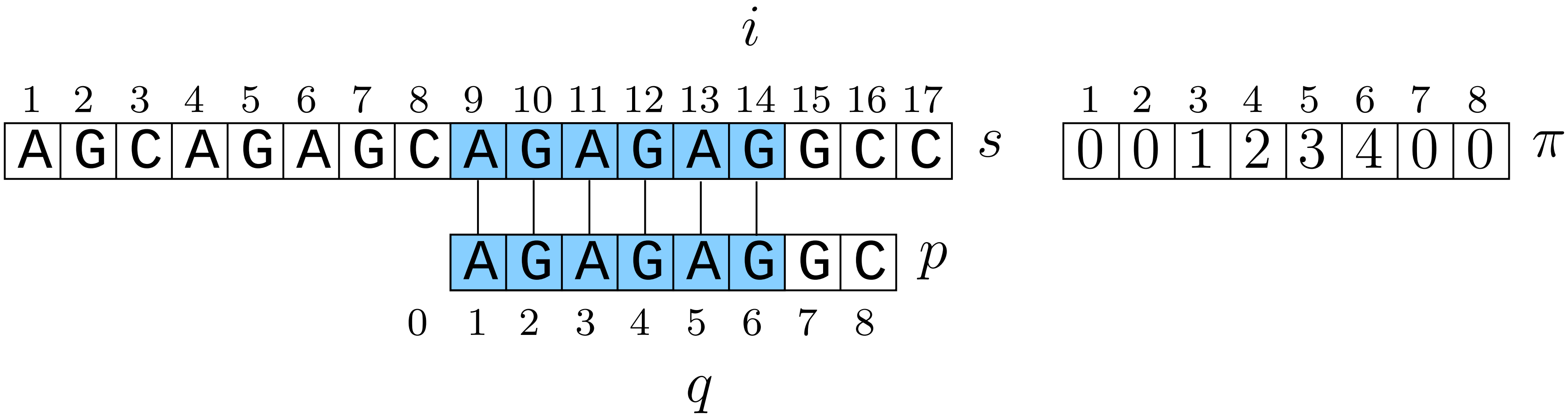
Knuth, Morris e Pratt



Knuth, Morris e Pratt



Knuth, Morris e Pratt



i

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C	s

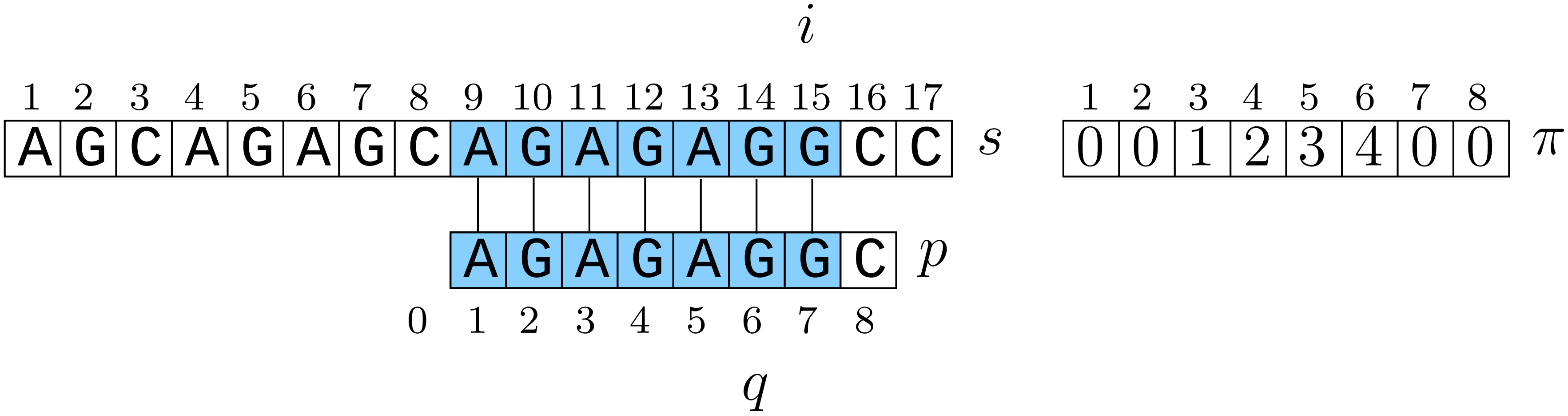
1	2	3	4	5	6	7	8	
0	0	1	2	3	4	0	0	π

A	G	A	G	A	G	G	C	p

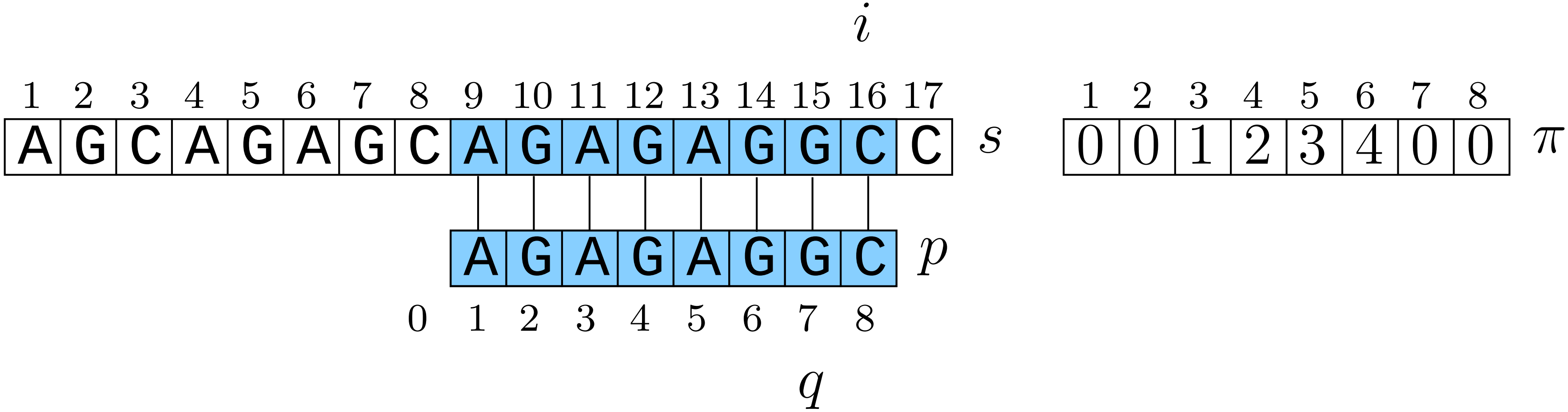
0 1 2 3 4 5 6 7 8

q

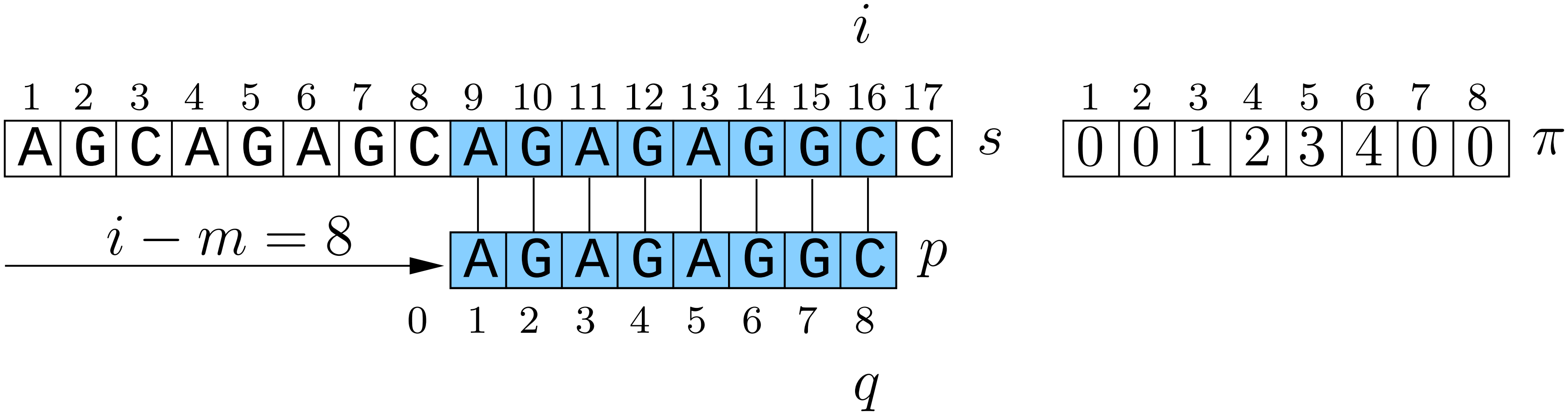
Knuth, Morris e Pratt



Knuth, Morris e Pratt



Knuth, Morris e Pratt

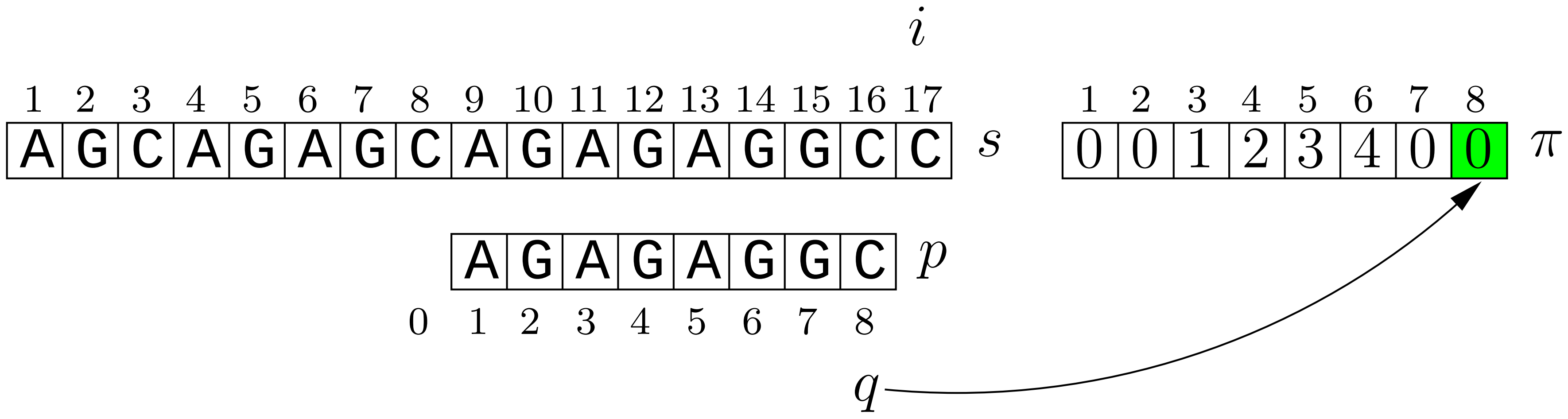


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
s	A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

	0	1	2	3	4	5	6	7	8
p		A	G	A	G	A	G	G	C

	1	2	3	4	5	6	7	8
π	0	0	1	2	3	4	0	0

Knuth, Morris e Pratt



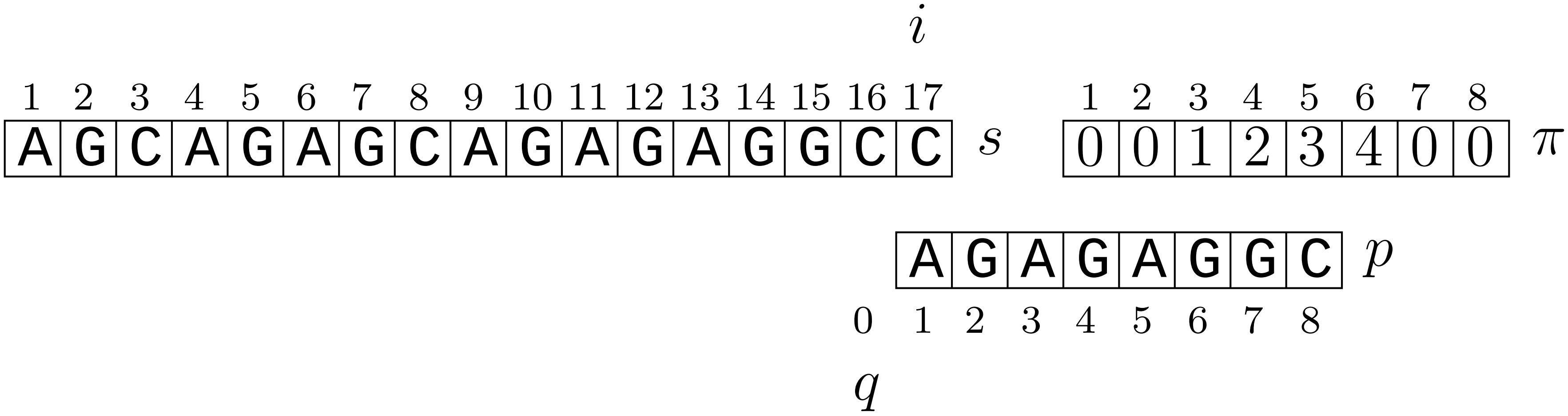
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

A	G	A	G	A	G	G	C

0 1 2 3 4 5 6 7 8

Knuth, Morris e Pratt



i

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C	s

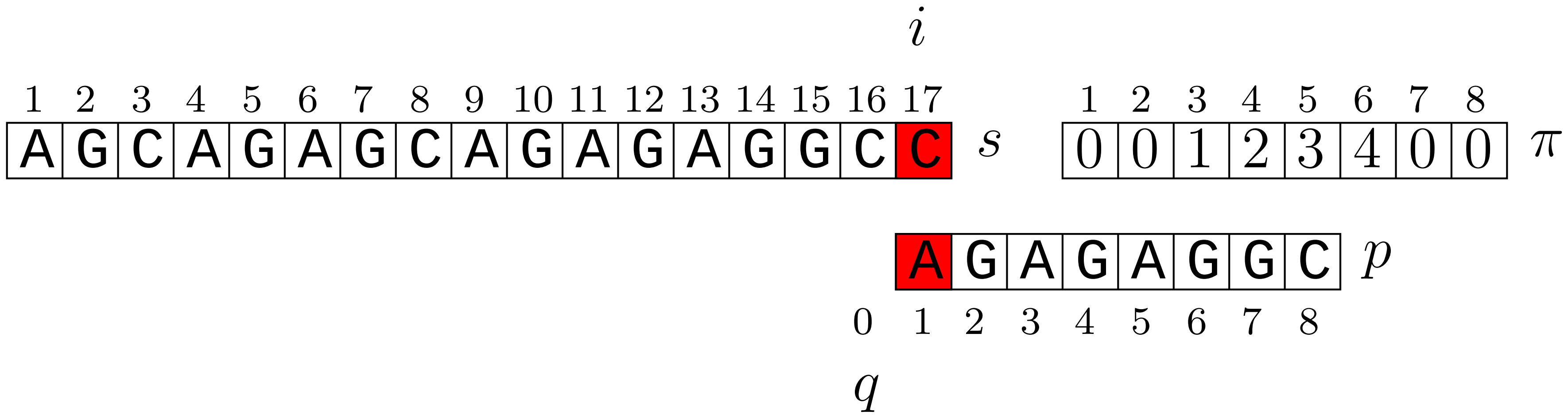
1	2	3	4	5	6	7	8	
0	0	1	2	3	4	0	0	π

A	G	A	G	A	G	G	C	p

0 1 2 3 4 5 6 7 8

q

Knuth, Morris e Pratt

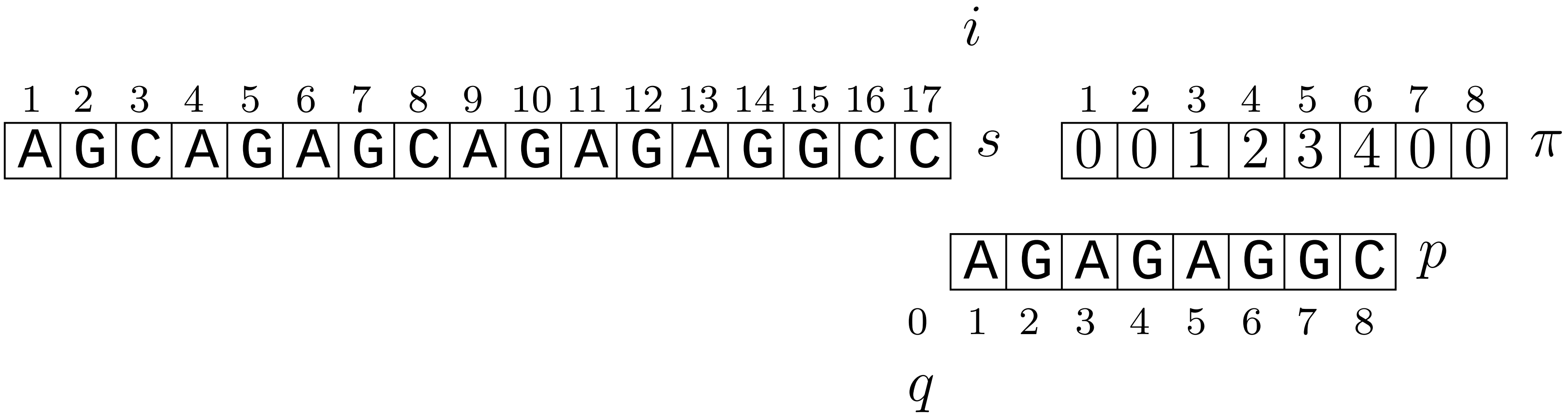


1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

1	2	3	4	5	6	7	8
0	0	1	2	3	4	0	0

0	1	2	3	4	5	6	7	8
	A	G	A	G	A	G	G	C

Knuth, Morris e Pratt



i

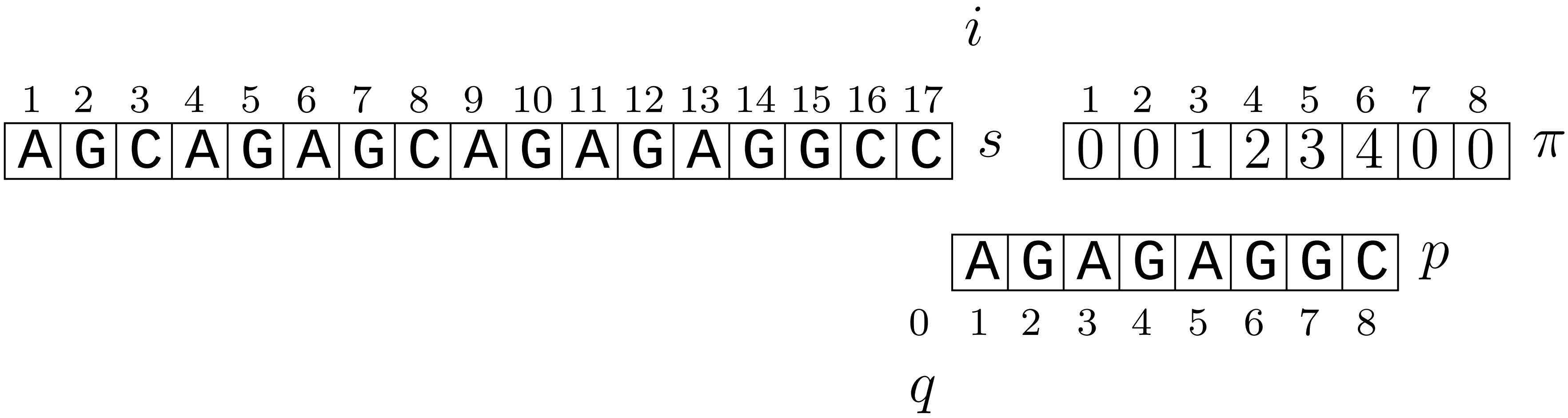
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C	s

1	2	3	4	5	6	7	8	
0	0	1	2	3	4	0	0	π

	A	G	A	G	A	G	G	C p
0	1	2	3	4	5	6	7	8

q

Knuth, Morris e Pratt



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
s	A	G	C	A	G	A	G	C	A	G	A	G	A	G	G	C	C

	1	2	3	4	5	6	7	8
π	0	0	1	2	3	4	0	0

	0	1	2	3	4	5	6	7	8
p		A	G	A	G	A	G	G	C

Tempo de execução: $O(m+n)$.

Boyer e Moore

- O algoritmo de Boyer e Moore tem a estrutura do algoritmo ingênuo com a adição de duas heurísticas:

Boyer e Moore

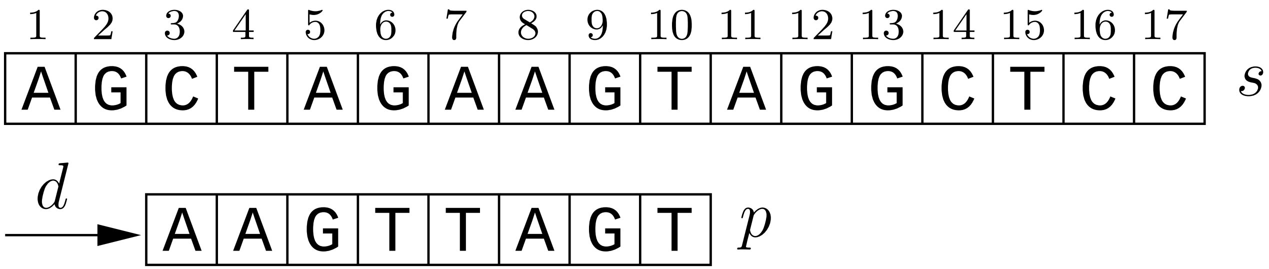
- O algoritmo de Boyer e Moore tem a estrutura do algoritmo ingênuo com a adição de duas heurísticas:
    - heurística do símbolo ruim;

Boyer e Moore

- O algoritmo de Boyer e Moore tem a estrutura do algoritmo ingênuo com a adição de duas heurísticas:
  - heurística do símbolo ruim;
  - heurística do sufixo bom.

Boyer e Moore

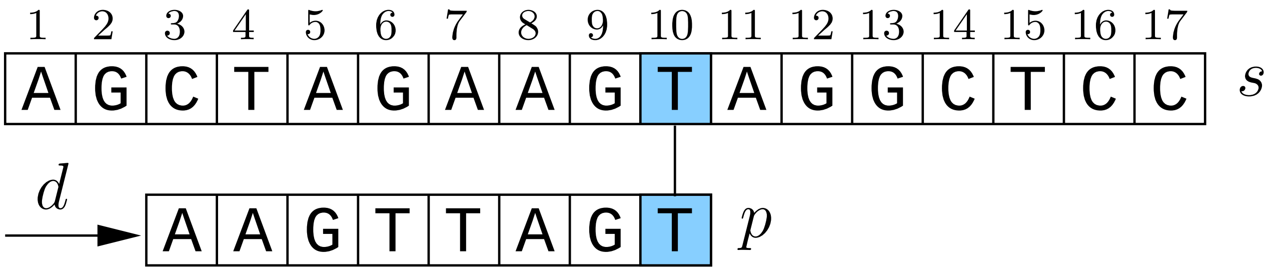
Heurística do símbolo ruim



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	T	A	G	A	A	G	T	A	G	G	C	T	C	C	s
		A	A	G	T	T	A	G	T								p

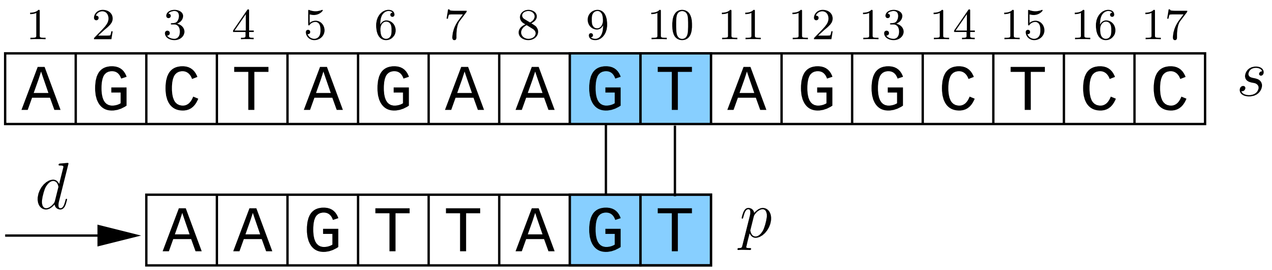
Boyer e Moore

Heurística do símbolo ruim



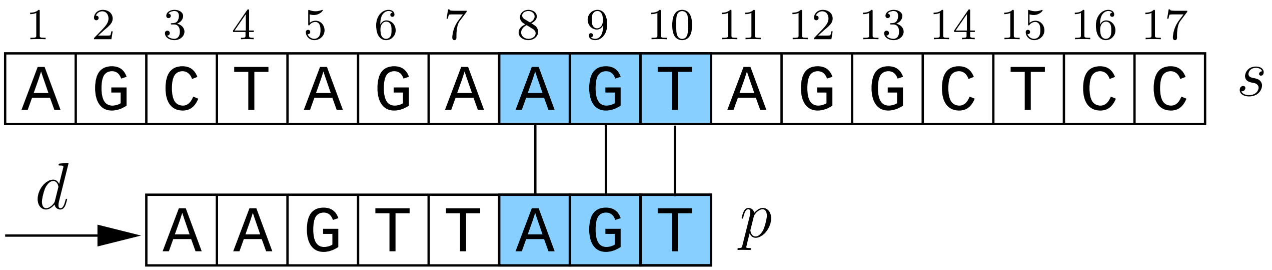
Boyer e Moore

Heurística do símbolo ruim



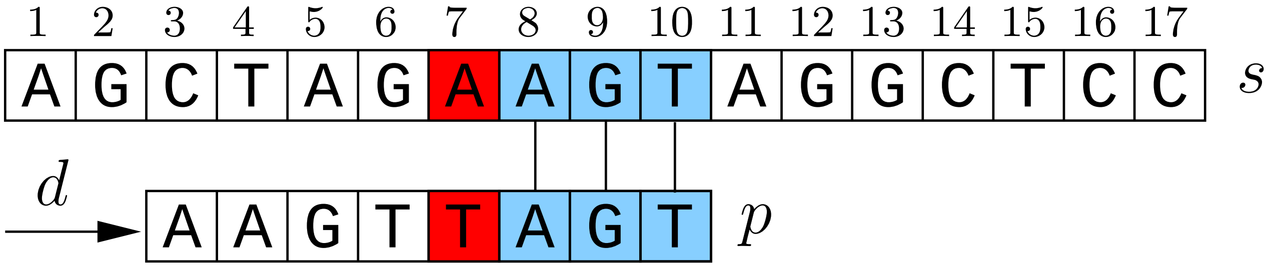
Boyer e Moore

Heurística do símbolo ruim



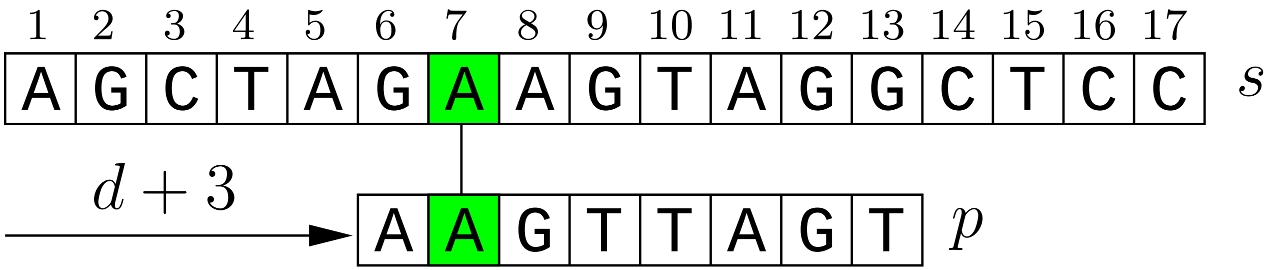
Boyer e Moore

Heurística do símbolo ruim



Boyer e Moore

Heurística do símbolo ruim



Dado um deslocamento $d \geqslant 0$, se $p[j] \neq s[d+j]$ para algum j, com $1 \leqslant j \leqslant m$, então a heurística do símbolo ruim encontra o maior índice k, com $1 \leqslant k \leqslant m$, tal que $s[d+j] = p[k]$ se um tal k existe. Caso contrário, $k = 0$. Então, o algoritmo toma o próximo deslocamento como sendo $d + j - k$.

Boyer e Moore

A **função última ocorrência** λ implementa a idéia de deslocamentos baseados em um símbolo ruim e é definida da seguinte forma: $\lambda[a]$ é o índice da posição mais à direita no padrão p em que o símbolo a ocorre, para cada símbolo $a \in \Sigma$.

Boyer e Moore

FUNÇÃO-ÚLTIMA-OCORRÊNCIA(p, Σ): recebe um padrão p de m símbolos e um alfabeto Σ e devolve a função última ocorrência λ para todo símbolo em Σ.

```
para cada símbolo $a \in \Sigma$ faça
    $\lambda[a] \leftarrow 0$
para $j \leftarrow 1$ até $m$ faça
    $\lambda[p[j]] \leftarrow j$
devolva $\lambda$
```

Boyer e Moore

FUNÇÃO-ÚLTIMA-OCORRÊNCIA(p, Σ): recebe um padrão p de m símbolos e um alfabeto Σ e devolve a função última ocorrência λ para todo símbolo em Σ.

```
para cada símbolo $a \in \Sigma$ faça
    $\lambda[a] \leftarrow 0$
para $j \leftarrow 1$ até $m$ faça
    $\lambda[p[j]] \leftarrow j$
devolva $\lambda$
```

Tempo de execução: $O(\Sigma + m)$

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
				λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
0				λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
0	0			λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
0	0	0		λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
0	0	0	0	λ

Boyer e Moore

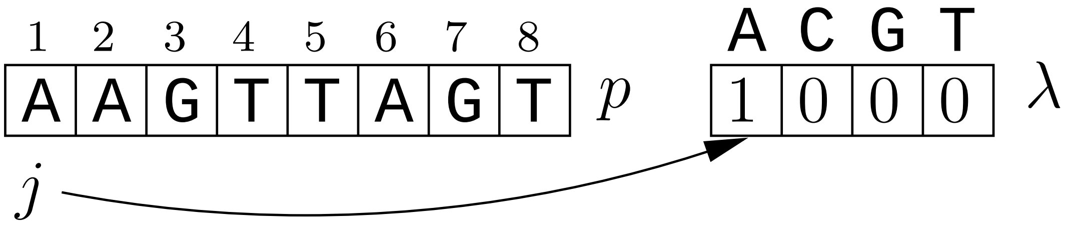
Heurística do símbolo ruim – Função última ocorrência

1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p
j								

A	C	G	T	
0	0	0	0	λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência



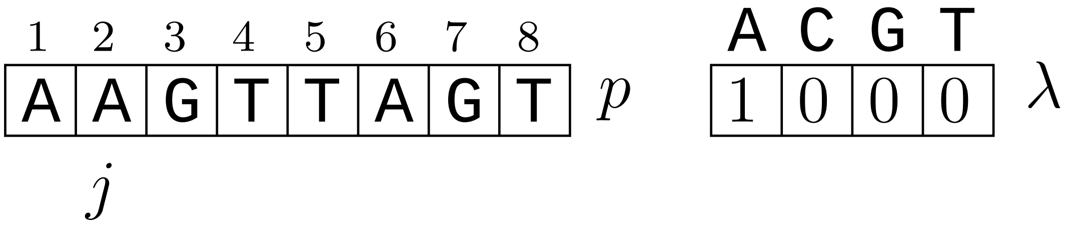
1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
1	0	0	0	λ

j

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

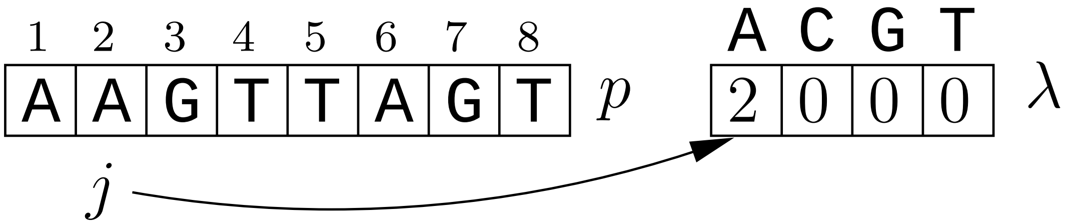


1	2	3	4	5	6	7	8
A	A	G	T	T	A	G	T

A	C	G	T
1	0	0	0

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência



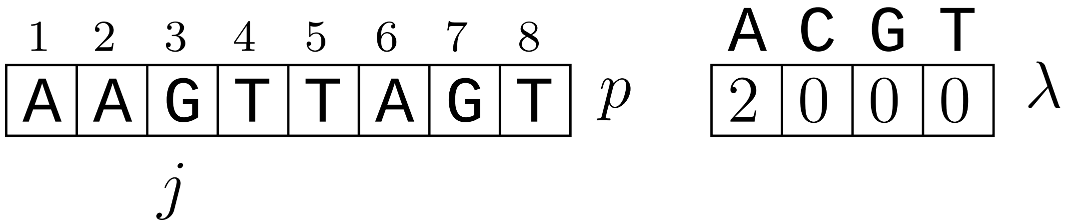
1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
2	0	0	0	λ

j

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência



1	2	3	4	5	6	7	8
A	A	G	T	T	A	G	T

p

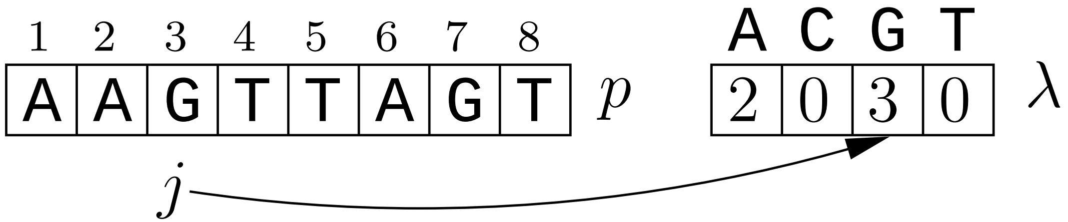
j

A	C	G	T
2	0	0	0

λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

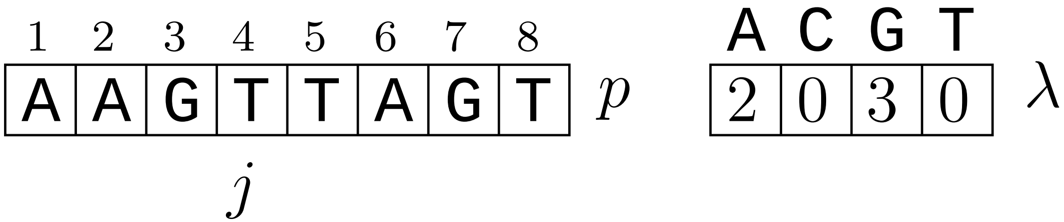


1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
2	0	3	0	λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

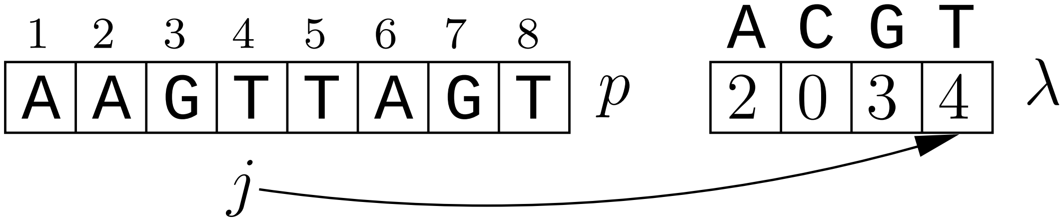


1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p
			j					

A	C	G	T	
2	0	3	0	λ

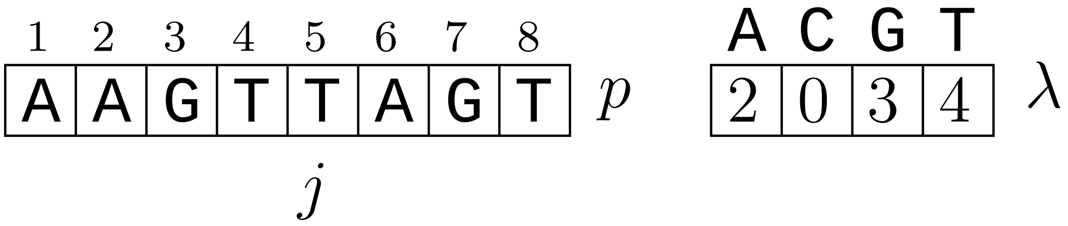
Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência



Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

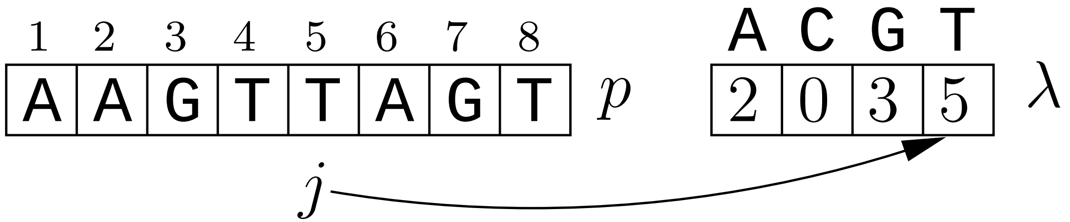


1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p
				j				

A	C	G	T	
2	0	3	4	λ

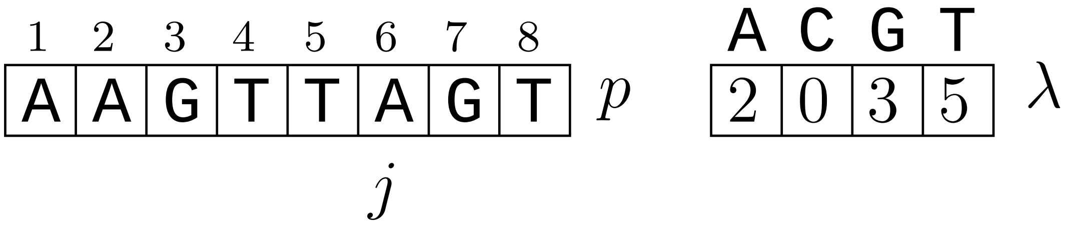
Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência



Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

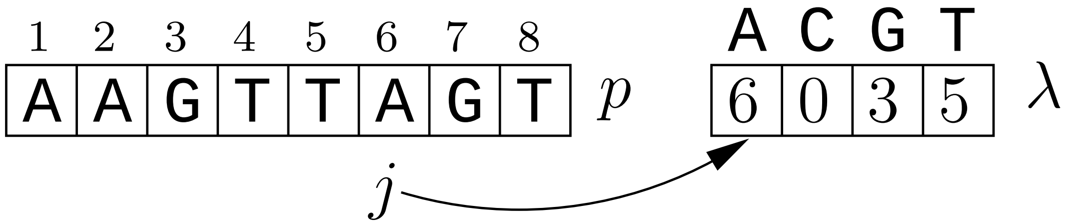


1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p
					j			

A	C	G	T	
2	0	3	5	λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

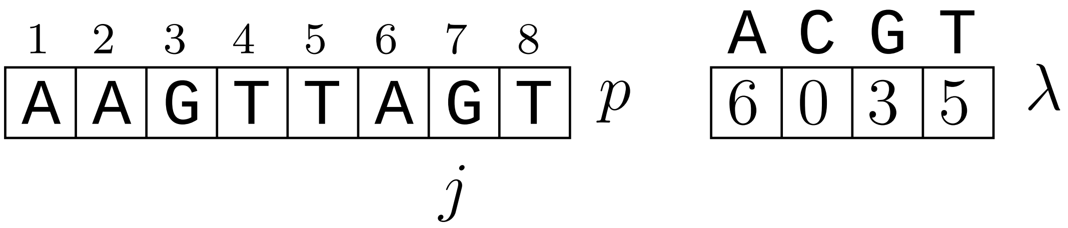


1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
6	0	3	5	λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência

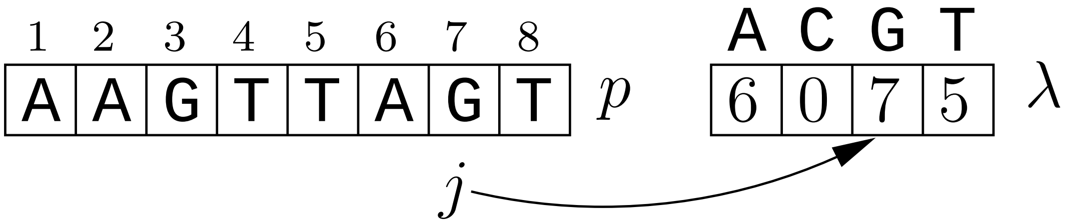


1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p
						j		

A	C	G	T	
6	0	3	5	λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência



Boyer e Moore

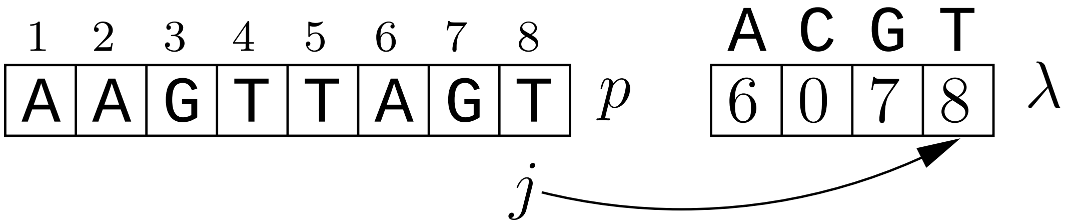
Heurística do símbolo ruim – Função última ocorrência

1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p
							j	

A	C	G	T	
6	0	7	5	λ

Boyer e Moore

Heurística do símbolo ruim – Função última ocorrência



1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
6	0	7	8	λ

Boyer e Moore

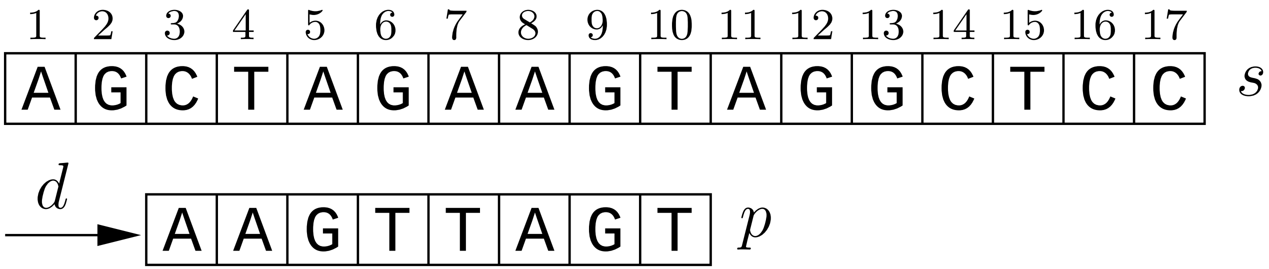
Heurística do símbolo ruim – Função última ocorrência

1	2	3	4	5	6	7	8	
A	A	G	T	T	A	G	T	p

A	C	G	T	
6	0	7	8	λ

Boyer e Moore

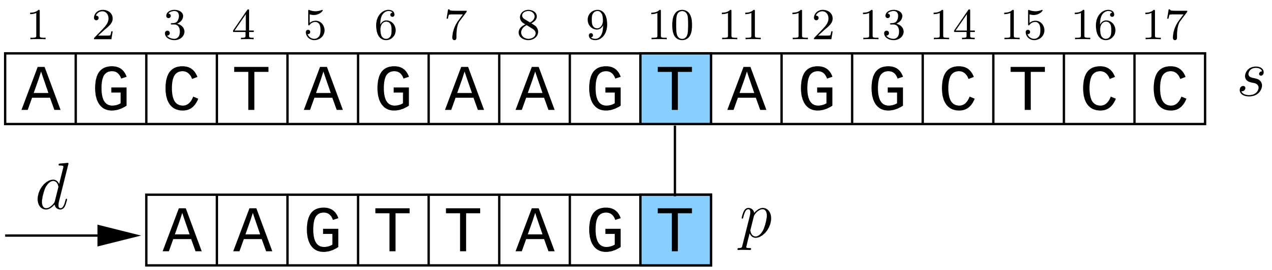
Heurística do sufixo bom



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
s	A	G	C	T	A	G	A	A	G	T	A	G	G	C	T	C	C
p			A	A	G	T	T	A	G	T							

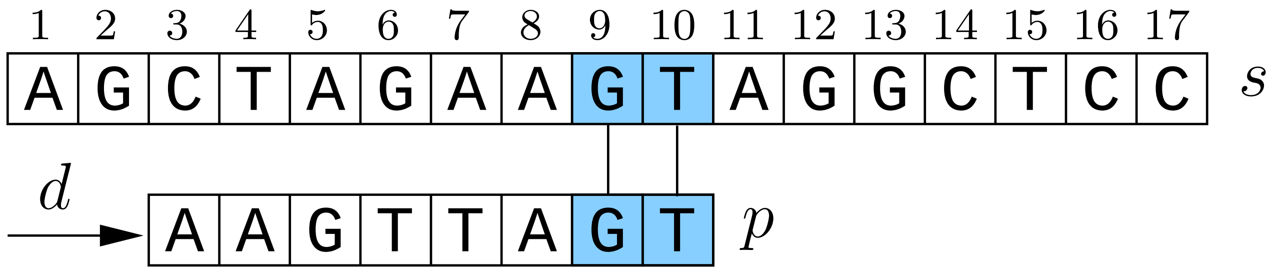
Boyer e Moore

Heurística do sufixo bom



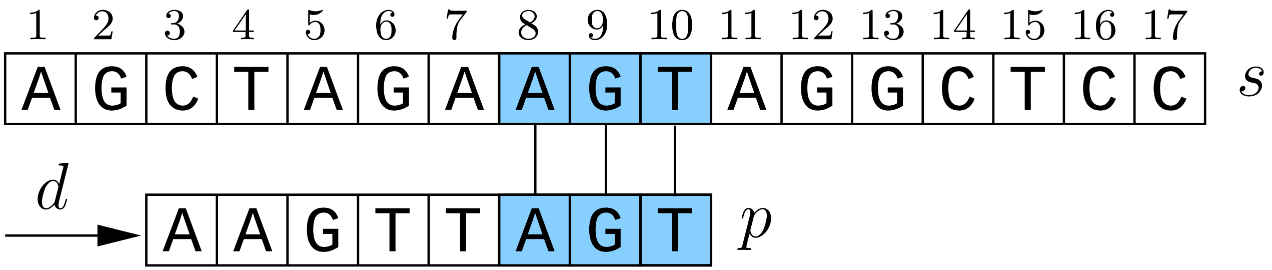
Boyer e Moore

Heurística do sufixo bom



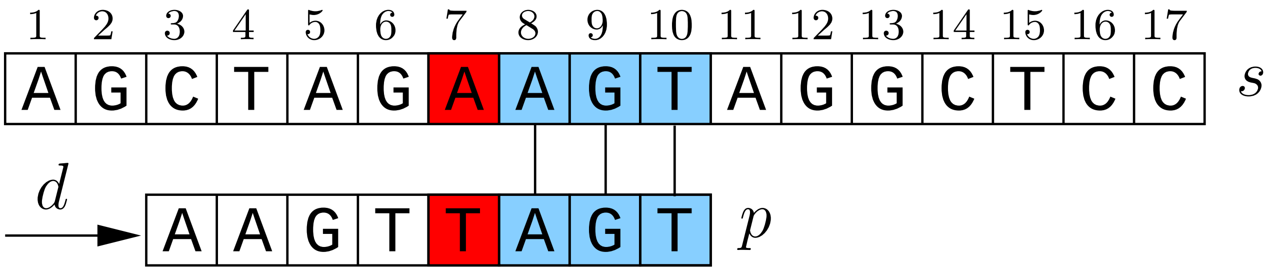
Boyer e Moore

Heurística do sufixo bom



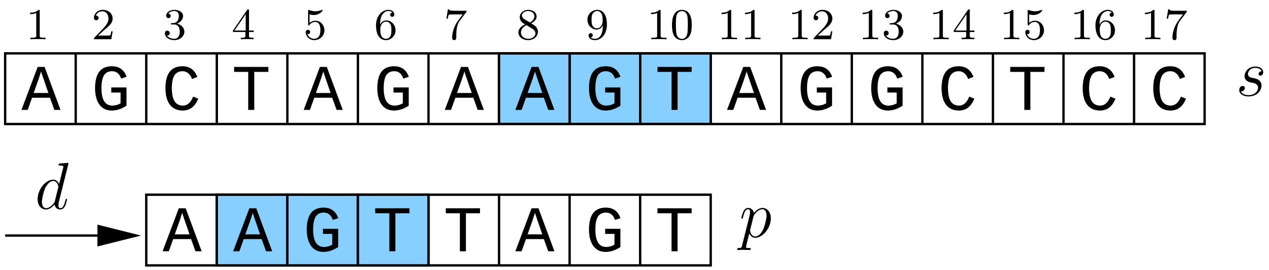
Boyer e Moore

Heurística do sufixo bom



Boyer e Moore

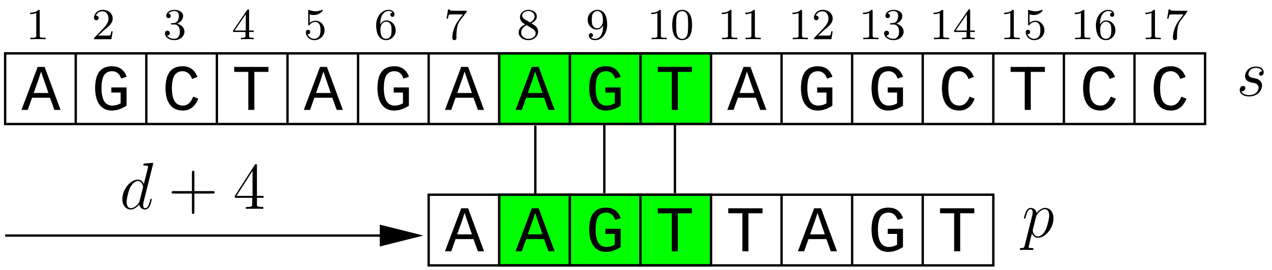
Heurística do sufixo bom



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	T	A	G	A	A	G	T	A	G	G	C	T	C	C	s
		A	A	G	T	T	A	G	T	p							

Boyer e Moore

Heurística do sufixo bom



Boyer e Moore

Heurística do sufixo bom

- Dado um deslocamento $d \geqslant 0$, se $p[j] \neq s[d+j]$, para $j < m$, então d pode ser seguramente avançado de $\gamma[j]$ posições;

Boyer e Moore

Heurística do sufixo bom

- Dado um deslocamento $d \geqslant 0$, se $p[j] \neq s[d+j]$, para $j < m$, então d pode ser seguramente avançado de $\gamma[j]$ posições;
- A função γ é chamada **função sufixo bom** para o padrão p. O valor $\gamma[j]$ é o menor avanço possível para o deslocamento d para o qual os símbolos no sufixo bom $s[d+j+1..d+m]$ do texto se casam com alguma subseqüência do padrão;

Boyer e Moore

Heurística do sufixo bom

- Dado um deslocamento $d \geqslant 0$, se $p[j] \neq s[d+j]$, para $j < m$, então d pode ser seguramente avançado de $\gamma[j]$ posições;
- A função γ é chamada **função sufixo bom** para o padrão p. O valor $\gamma[j]$ é o menor avanço possível para o deslocamento d para o qual os símbolos no sufixo bom $s[d+j+1..d+m]$ do texto se casam com alguma subseqüência do padrão;
- A função sufixo bom é dada por

$$\gamma[j] = m - \max\{k \colon 0 \leqslant k < m \text{ e } p[j+1..m] \sim p[1..k]\}.$$

Ou seja, $\gamma[j]$ é o menor valor que podemos adicionar ao deslocamento d e para fazer com que os símbolos do sufixo bom $s[d+j+1..d+m]$ de s se casem com o novo posicionamento de p.

Boyer e Moore

Heurística do sufixo bom

- Dado um deslocamento $d \geqslant 0$, se $p[j] \neq s[d+j]$, para $j < m$, então d pode ser seguramente avançado de $\gamma[j]$ posições;
- A função γ é chamada **função sufixo bom** para o padrão p. O valor $\gamma[j]$ é o menor avanço possível para o deslocamento d para o qual os símbolos no sufixo bom $s[d+j+1..d+m]$ do texto se casam com alguma subseqüência do padrão;
- A função sufixo bom é dada por

$$\gamma[j] = m - \max\{k \colon \pi[m] \leqslant k < m \text{ e } p[j+1..m] \sim p[1..k]\},$$

onde π é a função prefixo para o padrão p.

Boyer e Moore

Heurística do sufixo bom

- Dado um deslocamento $d \geqslant 0$, se $p[j] \neq s[d+j]$, para $j < m$, então d pode ser seguramente avançado de $\gamma[j]$ posições;
- A função γ é chamada **função sufixo bom** para o padrão p. O valor $\gamma[j]$ é o menor avanço possível para o deslocamento d para o qual os símbolos no sufixo bom $s[d+j+1..d+m]$ do texto se casam com alguma subseqüência do padrão;
- A função sufixo bom é dada por

$$\gamma[j] = m - \max\Big(\{\pi[m]\} \cup \{k \colon \pi[m] \leqslant k < m \text{ e } p[j+1..m] \text{ é sufixo de } p[1..k]\}\Big),$$

onde π é a função prefixo para o padrão p.

Boyer e Moore

Heurística do sufixo bom

- Dado um deslocamento $d \geqslant 0$, se $p[j] \neq s[d+j]$, para $j < m$, então d pode ser seguramente avançado de $\gamma[j]$ posições;
- A função γ é chamada **função sufixo bom** para o padrão p. O valor $\gamma[j]$ é o menor avanço possível para o deslocamento d para o qual os símbolos no sufixo bom $s[d+j+1..d+m]$ do texto se casam com alguma subseqüência do padrão;
- A função sufixo bom é dada por

$$\gamma[j] = m - \max\left(\{\pi[m]\} \cup \{m - l + \pi'[l] \colon 1 \leqslant l \leqslant m \text{ e } j = m - \pi'[l]\}\right),$$

onde π é a função prefixo para o padrão p e π' é a função prefixo para o reverso p' do padrão p.

Boyer e Moore

Heurística do sufixo bom

- Dado um deslocamento $d \geqslant 0$, se $p[j] \neq s[d+j]$, para $j < m$, então d pode ser seguramente avançado de $\gamma[j]$ posições;
- A função γ é chamada **função sufixo bom** para o padrão p. O valor $\gamma[j]$ é o menor avanço possível para o deslocamento d para o qual os símbolos no sufixo bom $s[d+j+1..d+m]$ do texto se casam com alguma subseqüência do padrão;
- A função sufixo bom é dada por

$$\gamma[j] = \min\left(\{m - \pi[m]\} \cup \{l - \pi'[l] \colon 1 \leqslant l \leqslant m \text{ e } j = m - \pi'[l]\}\right),$$

onde π é a função prefixo para o padrão p e π' é a função prefixo para o reverso p' do padrão p.

Boyer e Moore

FUNÇÃO-SUFIXO-BOM(p): recebe o padrão p de m símbolos e devolve a função sufixo bom γ para p.

```
1: $\pi \leftarrow$ FUNÇÃO-PREFIXO($p$)
2: $p' \leftarrow$ REVERSO($p$)
3: $\pi' \leftarrow$ FUNÇÃO-PREFIXO($p'$)
4: **para** $j \leftarrow 0$ **até** $m$ **faça**
5:     $\gamma[j] \leftarrow m - \pi[m]$
6: **para** $l \leftarrow 1$ **até** $m$ **faça**
7:     $j \leftarrow m - \pi'[l]$
8:     **se** $\gamma[j] > l - \pi'[l]$ **então**
9:         $\gamma[j] \leftarrow l - \pi'[l]$
10: **devolva** $\gamma$
```

Boyer e Moore

FUNÇÃO-SUFIXO-BOM(p): recebe o padrão p de m símbolos e devolve a função sufixo bom γ para p.

```
FUNÇÃO-SUFIXO-BOM(p)
 $\pi \leftarrow$ FUNÇÃO-PREFIXO($p$)
 $p' \leftarrow$ REVERSO($p$)
 $\pi' \leftarrow$ FUNÇÃO-PREFIXO($p'$)
 para $j \leftarrow 0$ até $m$ faça
     $\gamma[j] \leftarrow m - \pi[m]$
 para $l \leftarrow 1$ até $m$ faça
     $j \leftarrow m - \pi'[l]$
     se $\gamma[j] > l - \pi'[l]$ então
         $\gamma[j] \leftarrow l - \pi'[l]$
 devolva $\gamma$
```

Tempo de execução: $O(m)$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

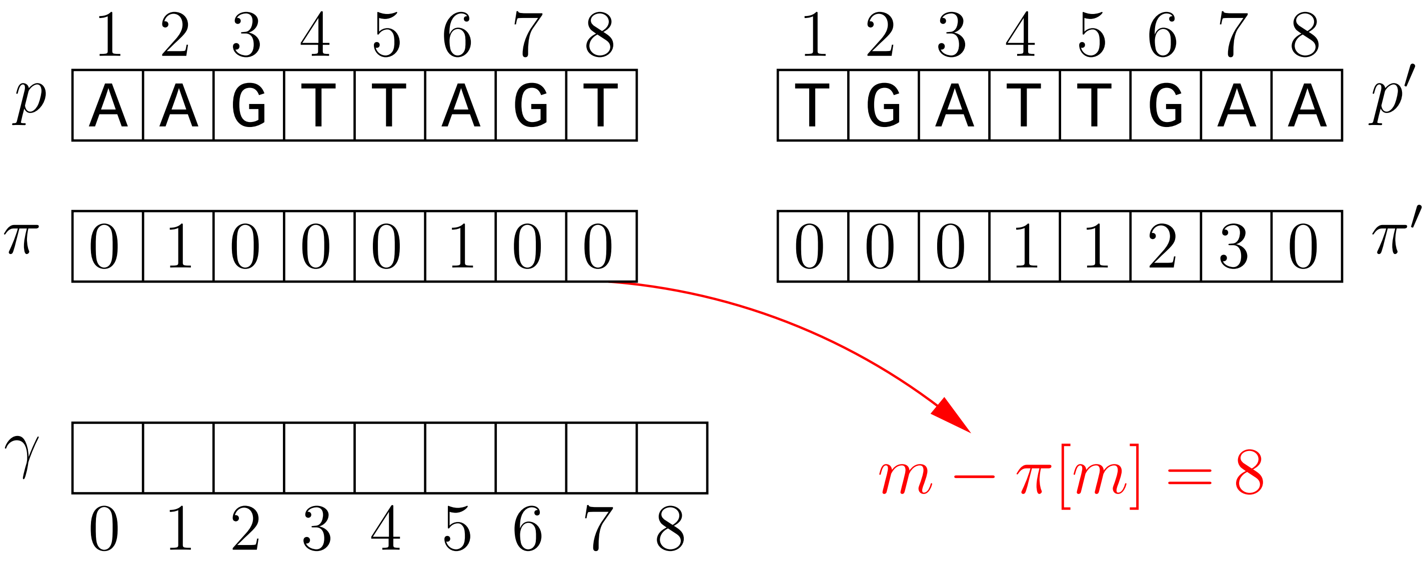
	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ									
	0	1	2	3	4	5	6	7	8

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom



	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ									
	0	1	2	3	4	5	6	7	8

$$m - \pi[m] = 8$$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	8	8	8	8
	0	1	2	3	4	5	6	7	8

$$m - \pi[m] = 8$$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

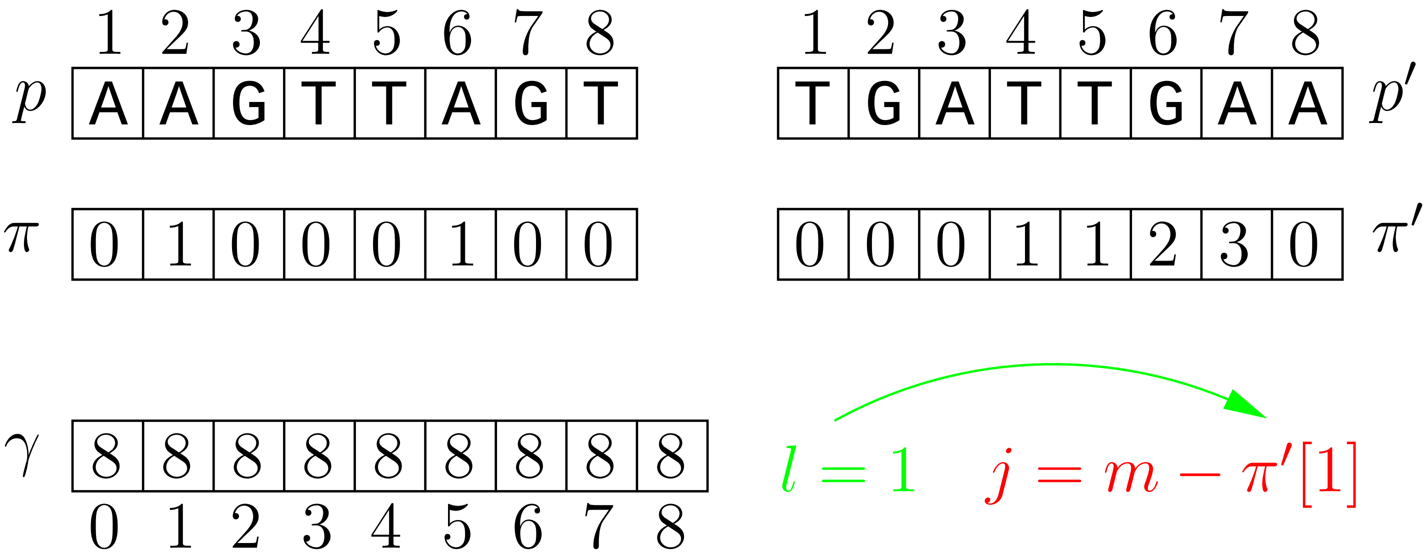
γ	8	8	8	8	8	8	8	8	8
	0	1	2	3	4	5	6	7	8

l

$j = m - \pi'[l]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom



	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

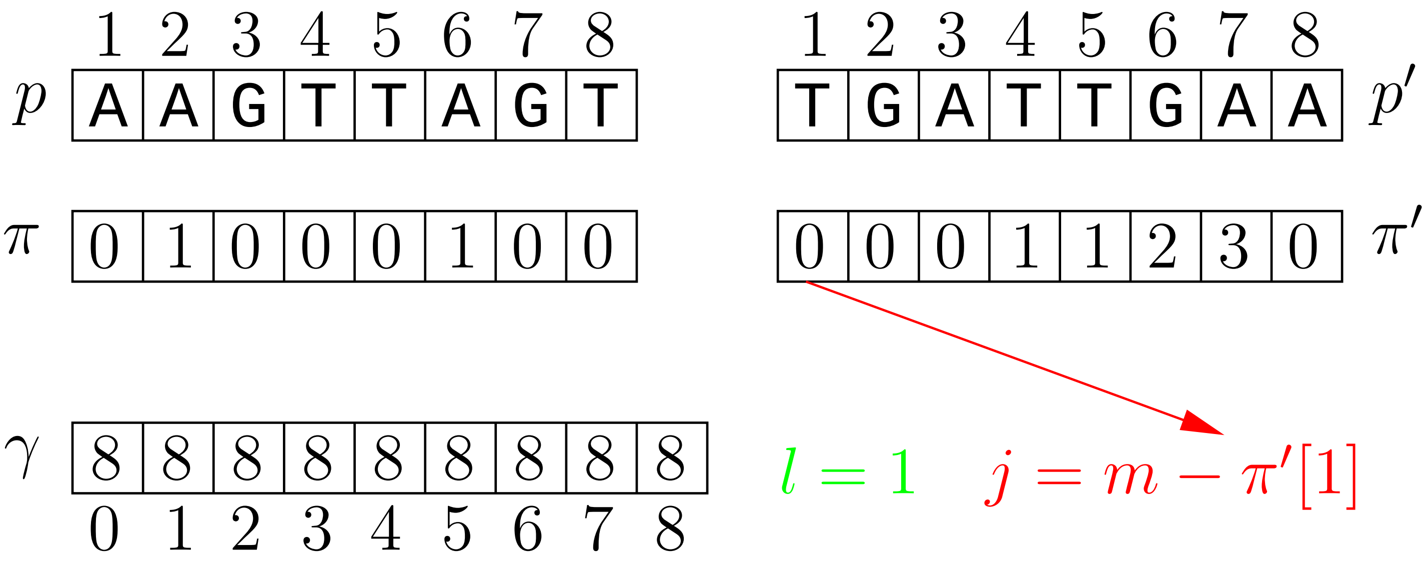
	1	2	3	4	5	6	7	8
p'	T	G	A	T	T	G	A	A
π'	0	0	0	1	1	2	3	0

	0	1	2	3	4	5	6	7	8
γ	8	8	8	8	8	8	8	8	8

$l = 1 \quad j = m - \pi'[1]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom



	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	8	8	8	8
	0	1	2	3	4	5	6	7	8

$l = 1 \quad j = m - \pi'[1]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	8	8	8	8
	0	1	2	3	4	5	6	7	8

$l = 1$ $\quad$ $j = 8$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

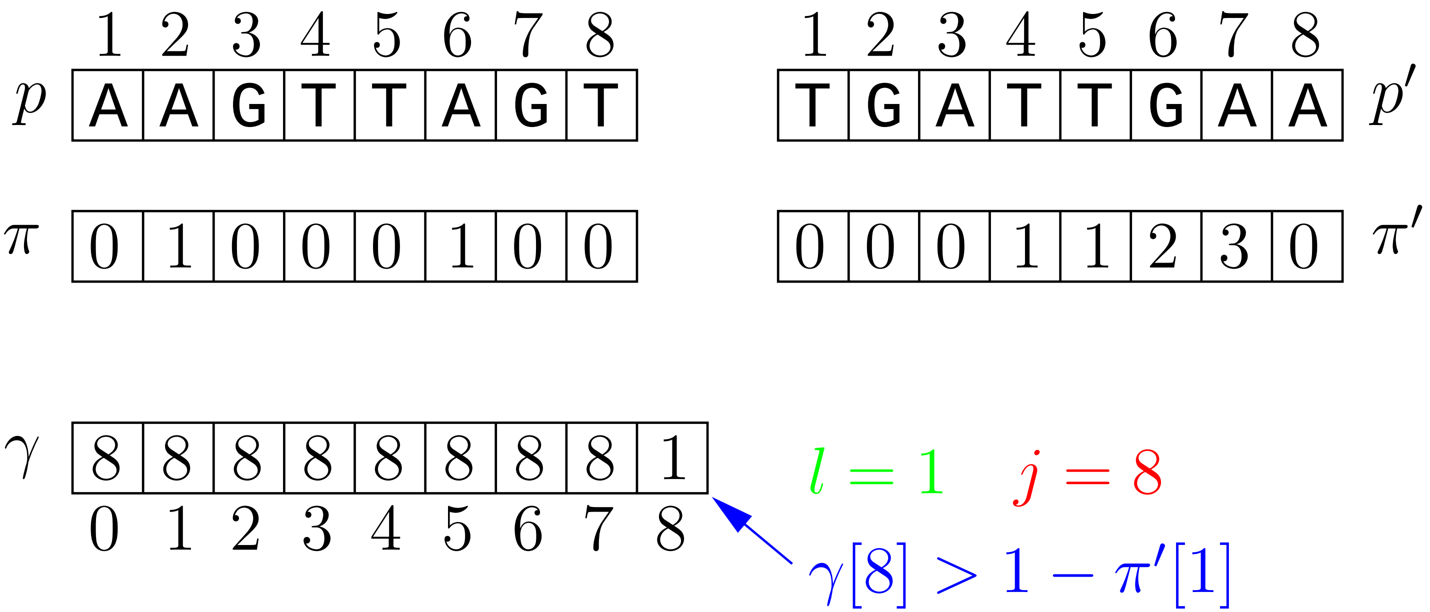
γ	8	8	8	8	8	8	8	8	8
	0	1	2	3	4	5	6	7	8

$l = 1$ $j = 8$

$\gamma[8] > 1 - \pi'[1]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom



	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

	0	1	2	3	4	5	6	7	8
γ	8	8	8	8	8	8	8	8	1

$l = 1 \quad j = 8$

$\gamma[8] > 1 - \pi'[1]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	8	8	8	1
	0	1	2	3	4	5	6	7	8

$l = 2 \quad j = 8$

$\gamma[8] \not> 2 - \pi'[2]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	8	8	8	1
	0	1	2	3	4	5	6	7	8

$l = 3 \quad j = 8$

$\gamma[8] \not> 3 - \pi'[3]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

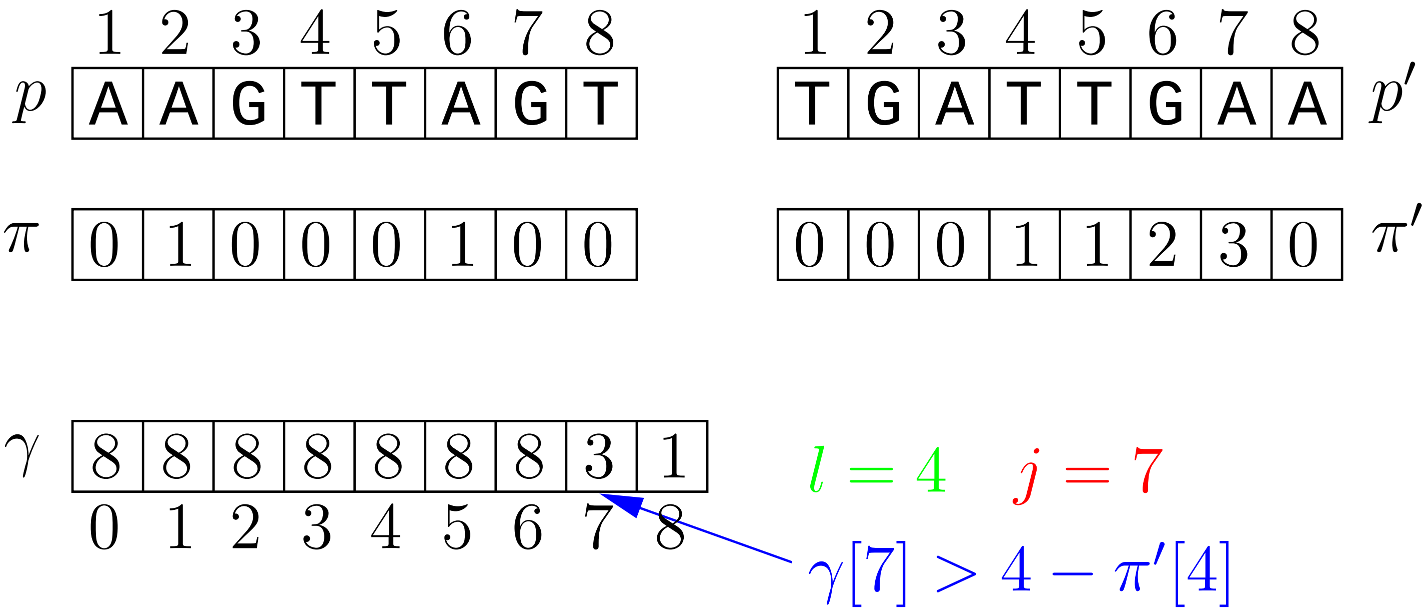
	0	1	2	3	4	5	6	7	8
γ	8	8	8	8	8	8	8	8	1

$l = 4 \quad j = 7$

$\gamma[7] > 4 - \pi'[4]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom



	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

	0	1	2	3	4	5	6	7	8
γ	8	8	8	8	8	8	8	3	1

$l = 4 \quad j = 7$

$\gamma[7] > 4 - \pi'[4]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	8	8	3	1
	0	1	2	3	4	5	6	7	8

$l = 5$ $\quad j = 7$

$\gamma[7] \not> 5 - \pi'[5]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	8	8	3	1
	0	1	2	3	4	5	6	7	8

$l = 6 \quad j = 6$

$\gamma[6] > 6 - \pi'[6]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	8	4	3	1
	0	1	2	3	4	5	6	7	8

$l = 6$ $\quad j = 6$

$\gamma[6] > 6 - \pi'[6]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	8	4	3	1
	0	1	2	3	4	5	6	7	8

$l = 7 \quad j = 5$

$\gamma[5] > 7 - \pi'[7]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	4	4	3	1
	0	1	2	3	4	5	6	7	8

$l = 7$ $\quad j = 5$

$\gamma[5] > 7 - \pi'[7]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	4	4	3	1
	0	1	2	3	4	5	6	7	8

$l = 8 \quad j = 8$

$\gamma[8] \not> 9 - \pi'[8]$

Boyer e Moore

Heurística do sufixo bom – Função sufixo bom

	1	2	3	4	5	6	7	8
p	A	A	G	T	T	A	G	T
π	0	1	0	0	0	1	0	0

1	2	3	4	5	6	7	8	
T	G	A	T	T	G	A	A	p'
0	0	0	1	1	2	3	0	π'

γ	8	8	8	8	8	4	4	3	1
	0	1	2	3	4	5	6	7	8

Boyer e Moore

BM(s, p, Σ): recebe um texto s de n símbolos, um padrão p de m símbolos de um alfabeto Σ e devolve os índices em s onde p ocorre.

```
$\lambda \leftarrow$ FUNÇÃO-ÚLTIMA-OCORRÊNCIA($p, \Sigma$)
$\gamma \leftarrow$ FUNÇÃO-SUFIXO-BOM($p$)
$d \leftarrow 0$
enquanto $d \leqslant n - m$ faça
    $j \leftarrow m$
    enquanto $j > 0$ e $p[j] = s[d + j]$ faça
        $j \leftarrow j - 1$
    se $j = 0$ então
        escreva "Padrão ocorre no texto com deslocamento" $d$
        $d \leftarrow d + \gamma[0]$
    senão
        $d \leftarrow d + \max(\gamma[j], j - \lambda[s[d + j]])$
```

Boyer e Moore

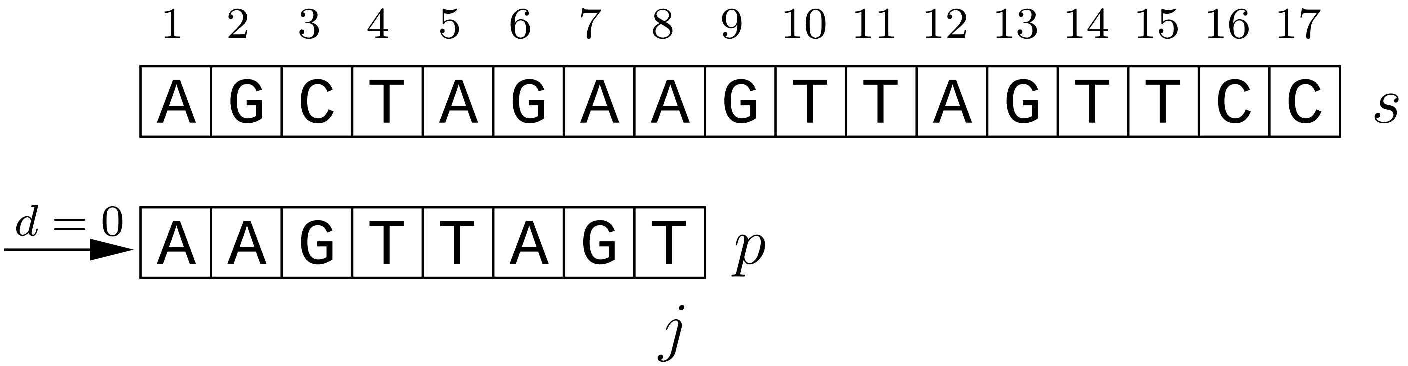
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C	s

A	A	G	T	T	A	G	T	p

A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

Boyer e Moore



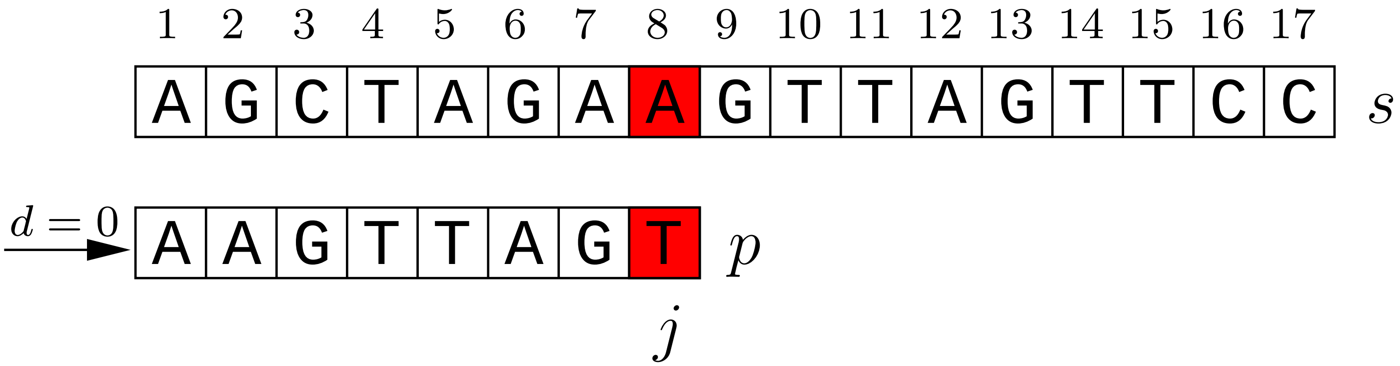
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C	s

A	A	G	T	T	A	G	T	p

A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

Boyer e Moore



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C	s

$d = 0$

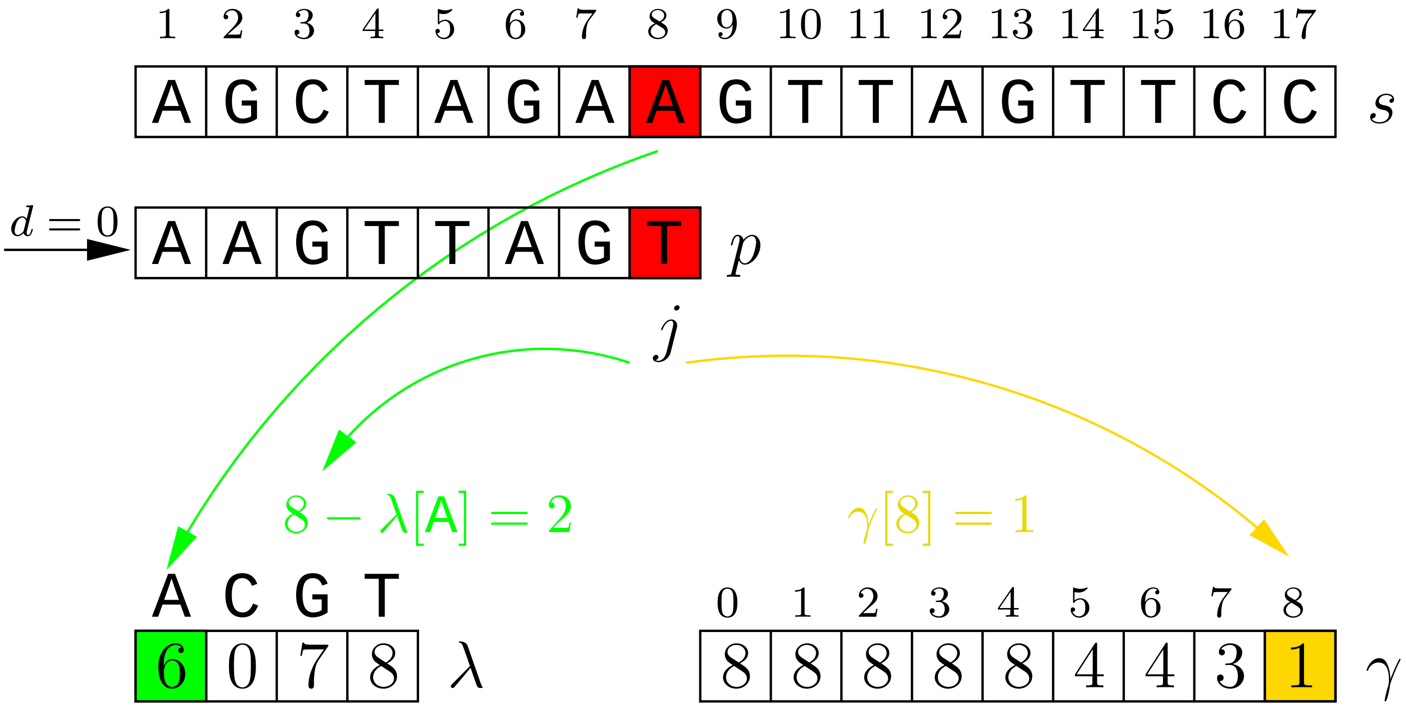
A	A	G	T	T	A	G	T	p

j

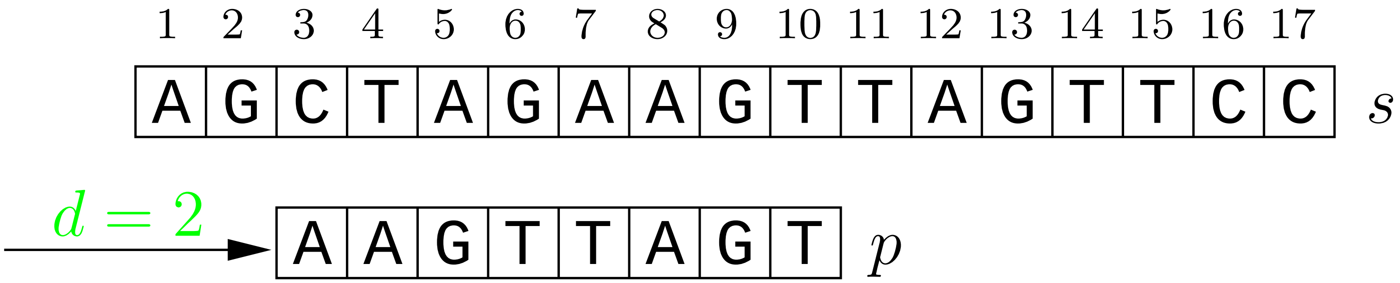
A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

Boyer e Moore



Boyer e Moore



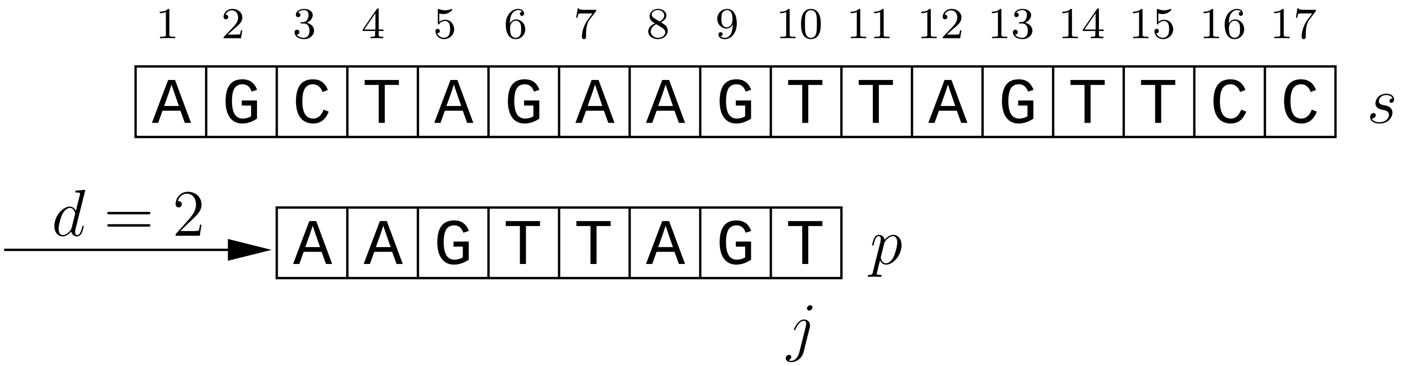
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
s	A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C
p			A	A	G	T	T	A	G	T							

$d = 2$

	A	C	G	T
λ	6	0	7	8

	0	1	2	3	4	5	6	7	8
γ	8	8	8	8	8	4	4	3	1

Boyer e Moore



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C	s

$d = 2$

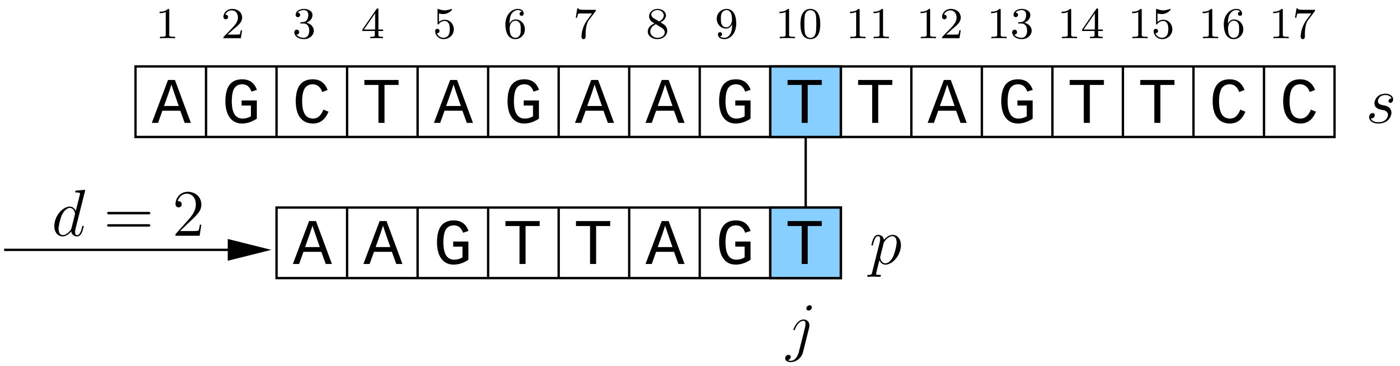
A	A	G	T	T	A	G	T	p

j

A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

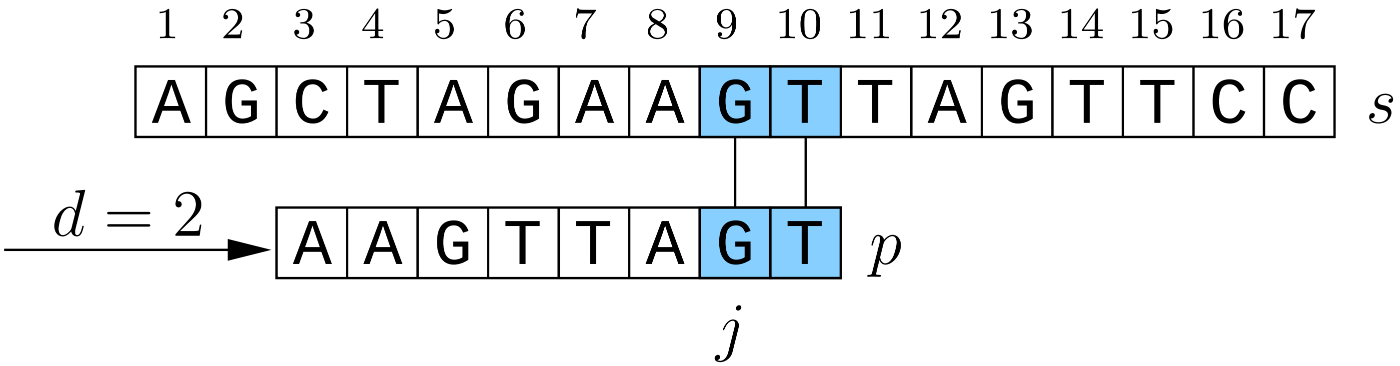
Boyer e Moore



A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

Boyer e Moore

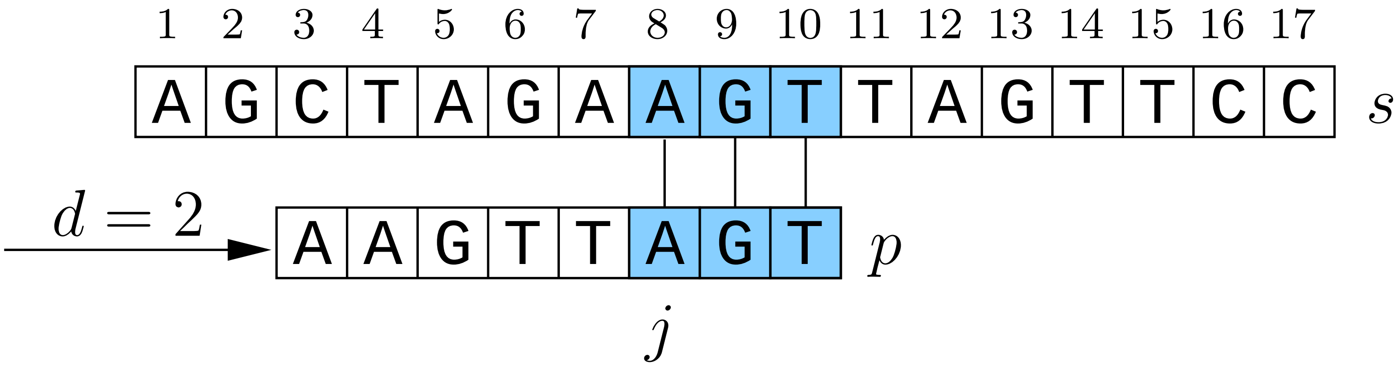


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
s	A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C
p			A	A	G	T	T	A	G	T							

A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

Boyer e Moore



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C	s

$d = 2$

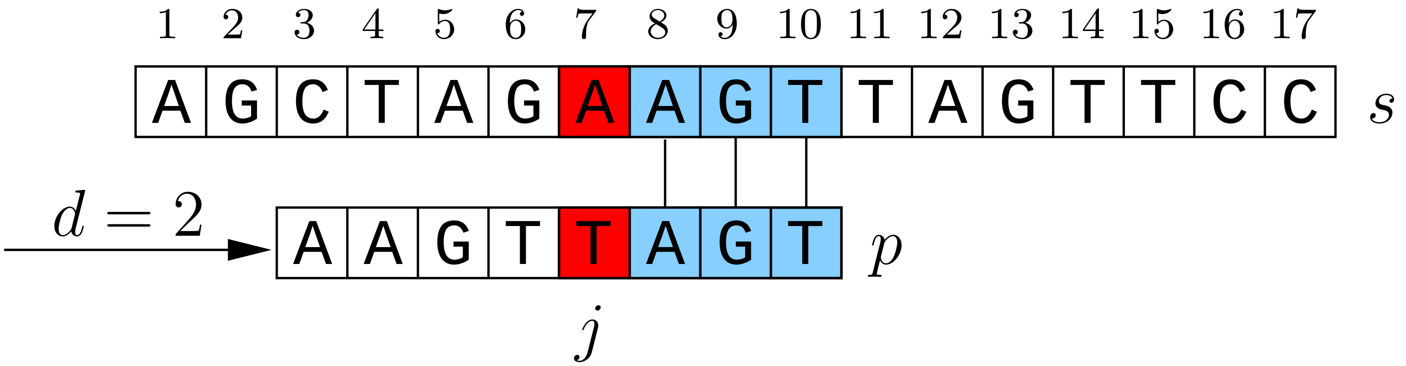
A	A	G	T	T	A	G	T	p

j

A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

Boyer e Moore



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C	s

$d = 2$

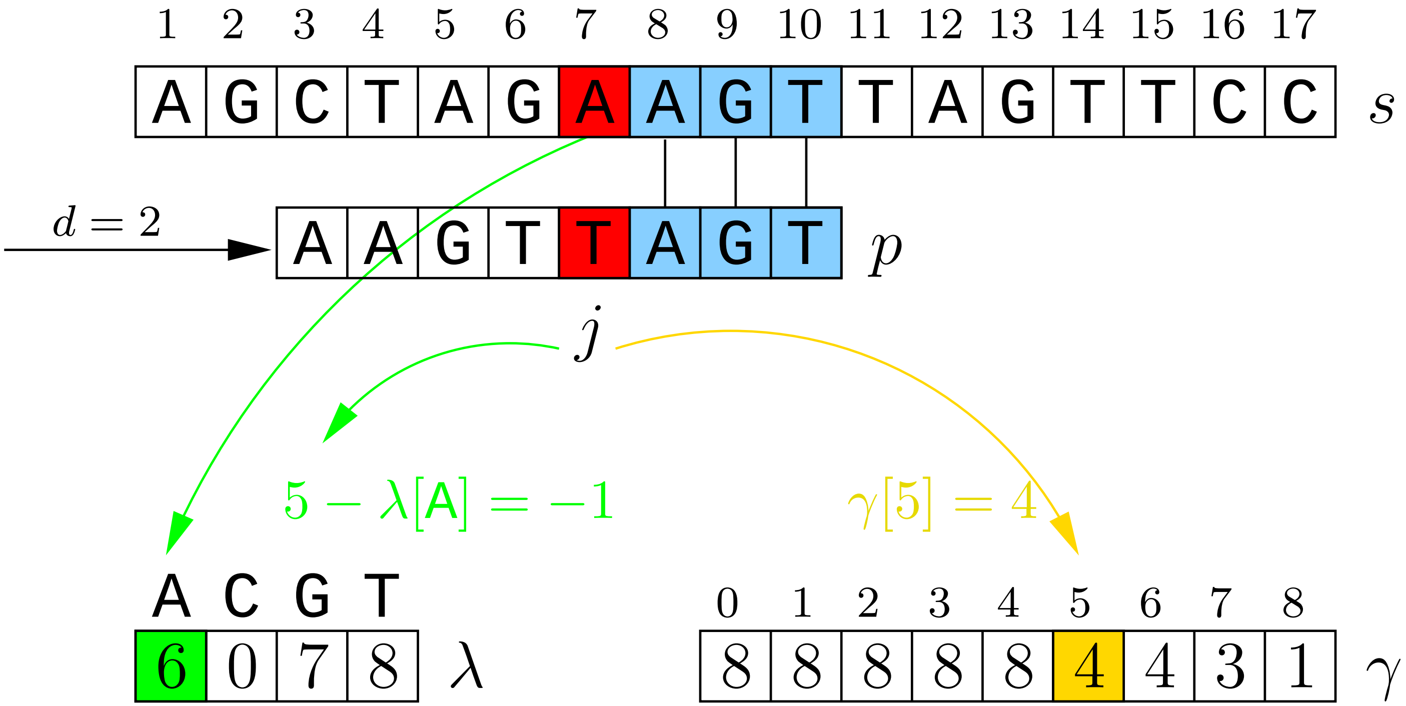
A	A	G	T	T	A	G	T	p

j

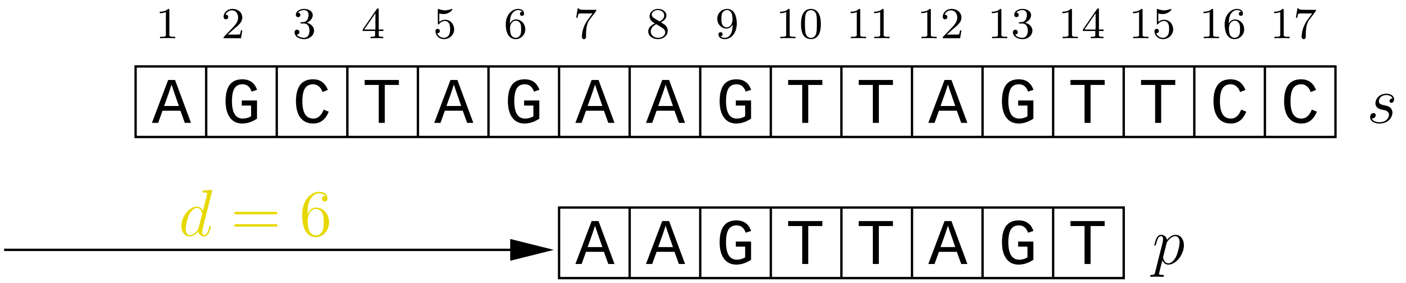
A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

Boyer e Moore



Boyer e Moore



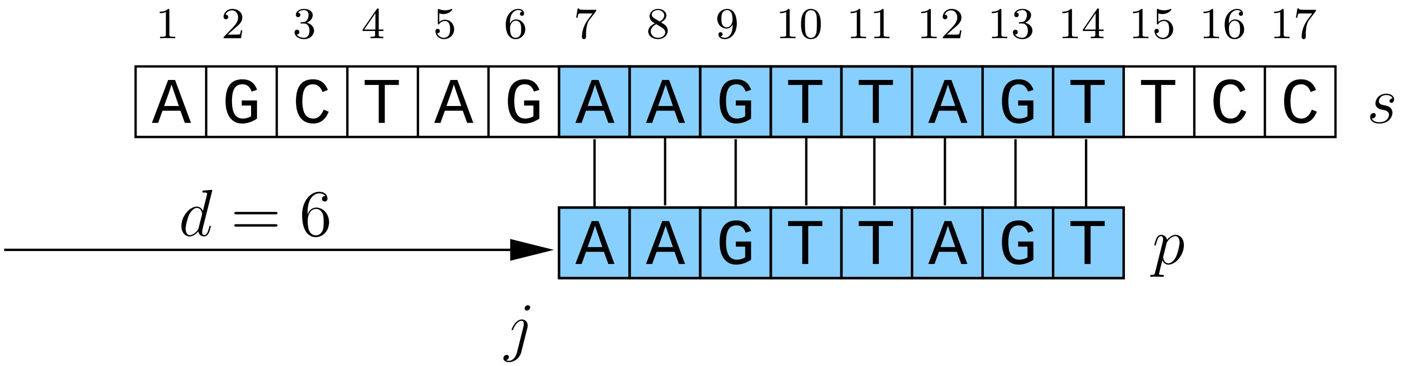
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
s	A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C
p							A	A	G	T	T	A	G	T			

$d = 6$

	A	C	G	T
λ	6	0	7	8

	0	1	2	3	4	5	6	7	8
γ	8	8	8	8	8	4	4	3	1

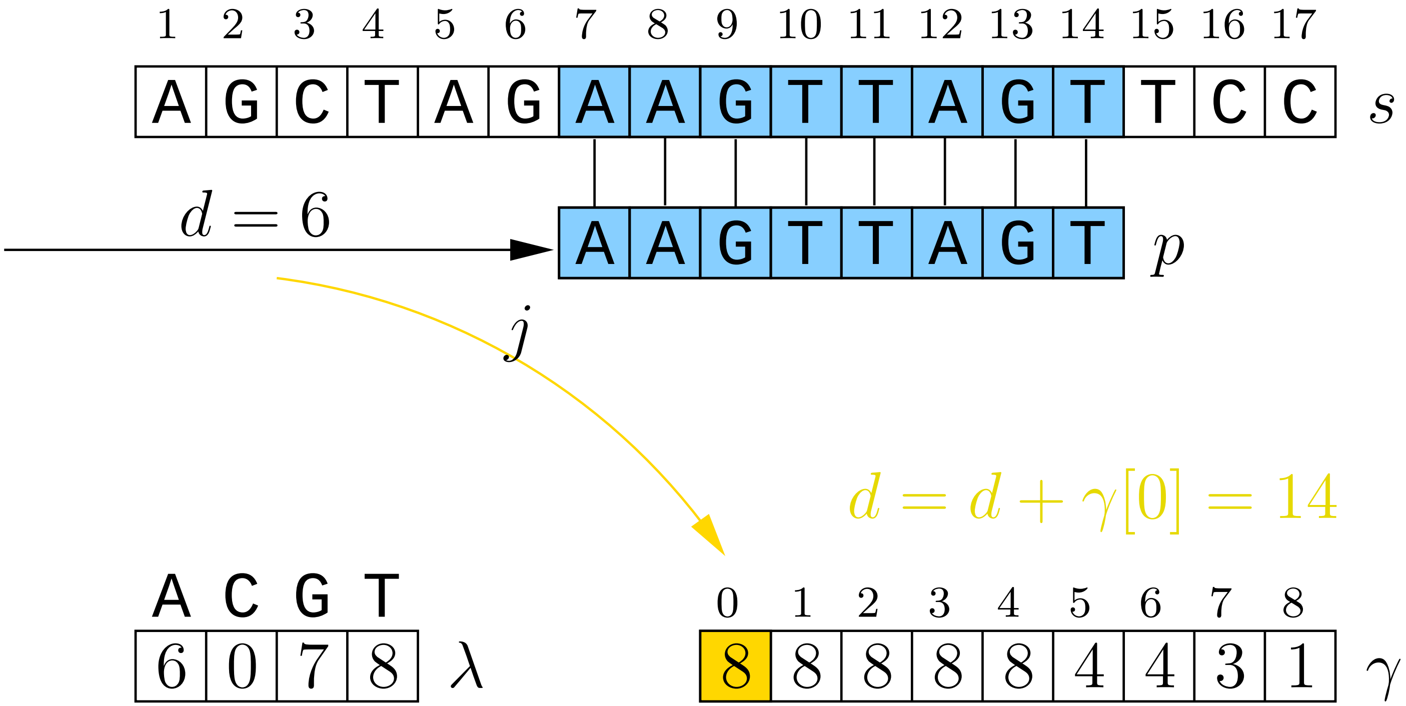
Boyer e Moore



A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

Boyer e Moore



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
A	G	C	T	A	G	A	A	G	T	T	A	G	T	T	C	C	s

$d = 6$

A	A	G	T	T	A	G	T	p

j

$d = d + \gamma[0] = 14$

A	C	G	T	
6	0	7	8	λ

0	1	2	3	4	5	6	7	8	
8	8	8	8	8	4	4	3	1	γ

Boyer e Moore

- O algoritmo BM tem tempo de execução

  $$O((n - m + 1)m + |\Sigma|)$$

  que supera o tempo de execução do algoritmo KMP e é equivalente ao tempo de execução do algoritmo ingênuo.

Boyer e Moore

- O algoritmo BM tem tempo de execução

  $$O((n - m + 1)m + |\Sigma|)$$

  que supera o tempo de execução do algoritmo KMP e é equivalente ao tempo de execução do algoritmo ingênuo.
- Muito eficiente na prática.