Projeto e Análise de Algoritmos

Processamento de Texto

Aplicações

- Pesquisas na Web (padrões)
- Aplicações linguísticas
- Cadeia de DNA
- Engenharia de software (controle de versões)

Cadeias de Caracteres

- “CGTAAACTGCTTTAATCAAACGC”

  Cadeias de moléculas chamadas bases:

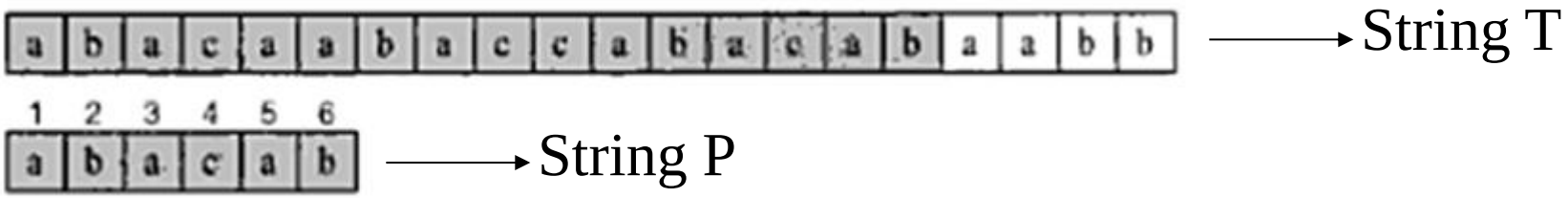
  adenina (A), guanina (G), citosina (C) e timina (T)

- “Botafogo é campeão brasileiro de 2011!”

- “http://www.iprj.uerj.br”

- “if ((x==0) && (y!=3))”

Encontrando padrões em textos (método da força bruta)



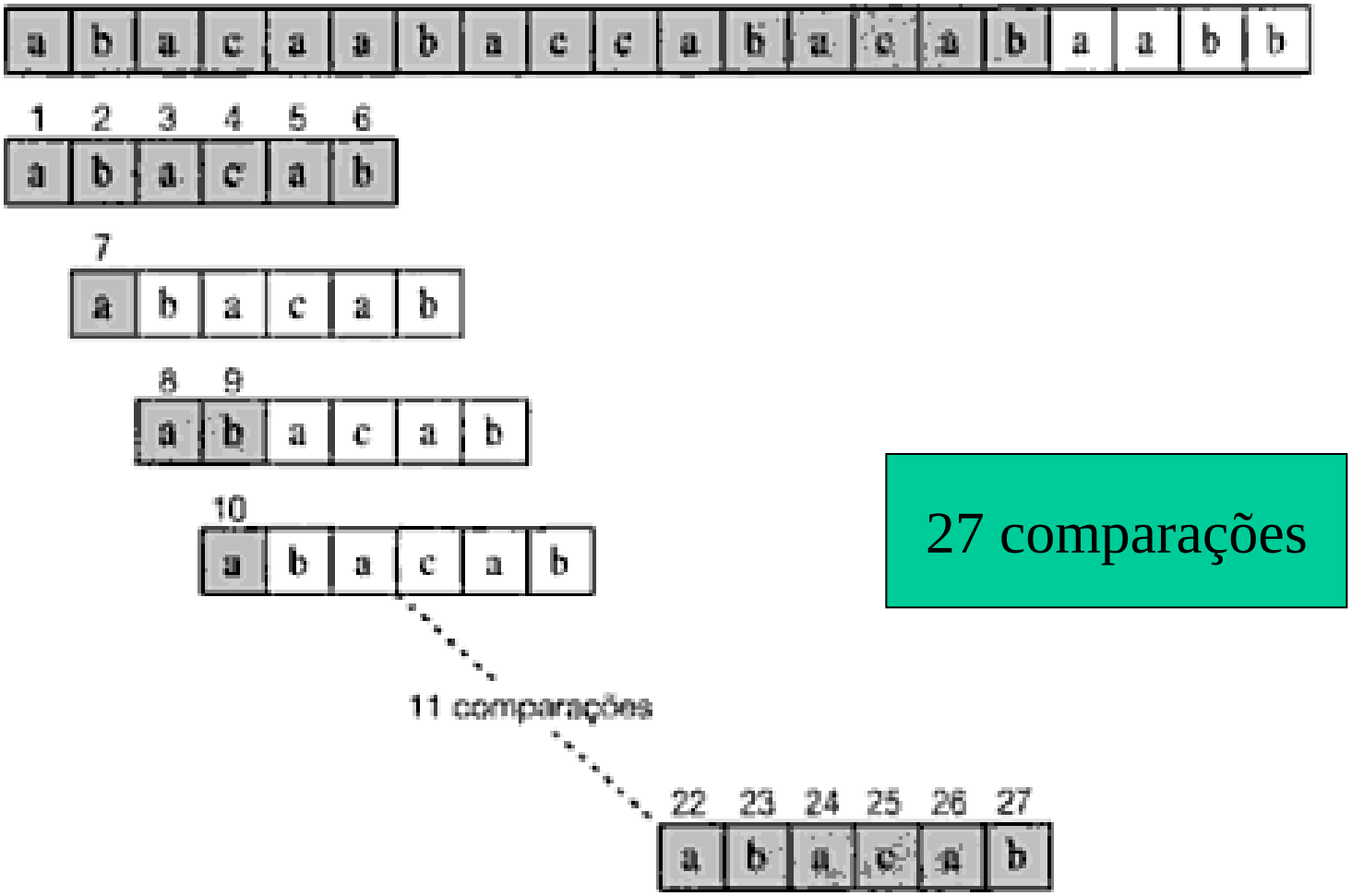
- Objetivo: descobrir se o string P (padrão) está em T
- Força bruta: caminha elemento a elemento da esquerda para a direita
- Exercício: escrever o algoritmo

Encontrando padrões em textos (força bruta)

```
Algoritmo ForcaBruta(T, P)
// T possui n caracteres e P possui m caracteres
Para i ← 0 até n - m faça ——————→ n-m+1 vezes
        j ← 0
        enquanto (j < m e T[i+j] == P[j]) faça ——→ m vezes
                j ← j + 1
        se j == m então
                retorne i
Retorne "Não existe substring em T igual a P."
```

O(nm). Se m = n/2, $O(n^2)$

Encontrando padrões em textos (força bruta)



27 comparações

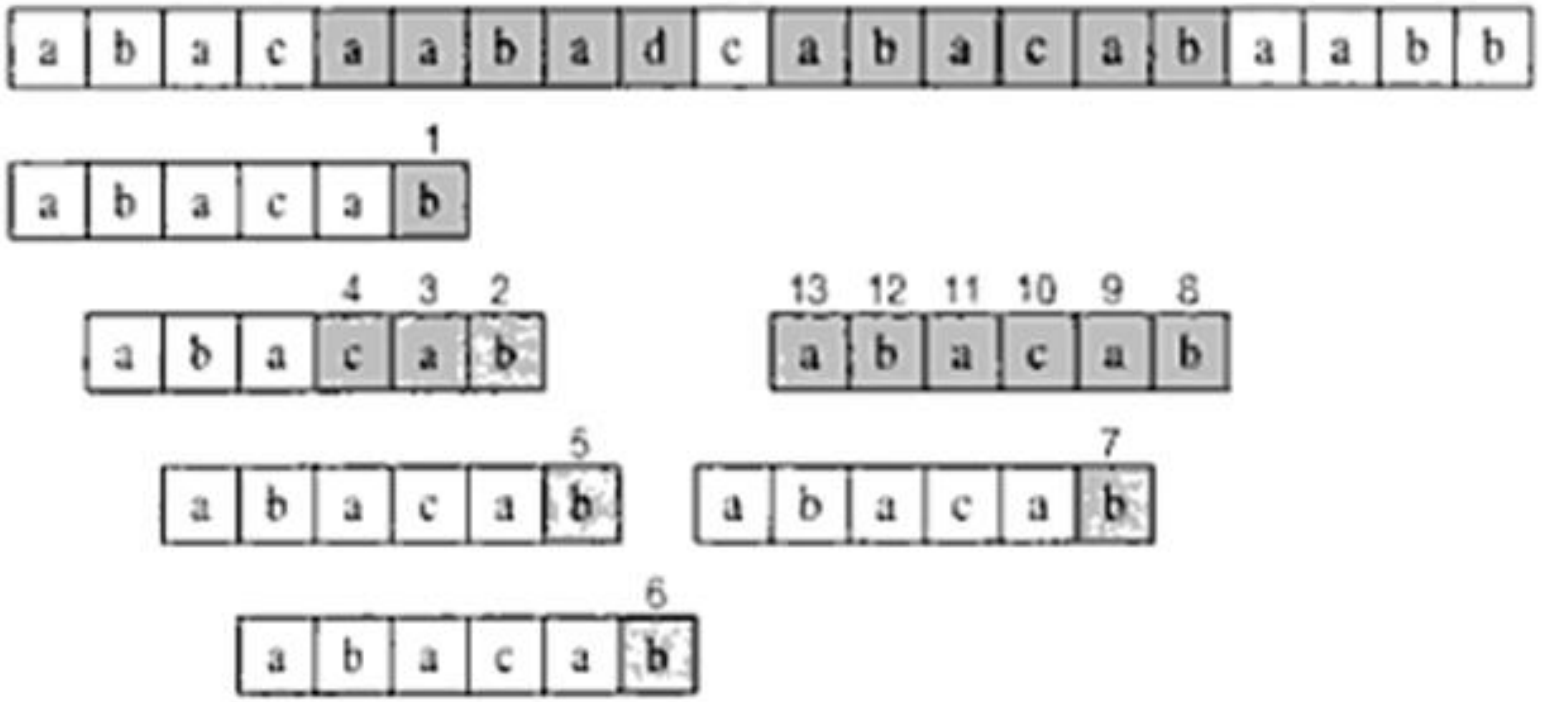
Encontrando padrões em textos (algoritmo Boyer-Moore)

- Objetivo: melhorar o tempo de execução do algoritmo de força bruta

- Heurística do espelho: comparação de forma invertida

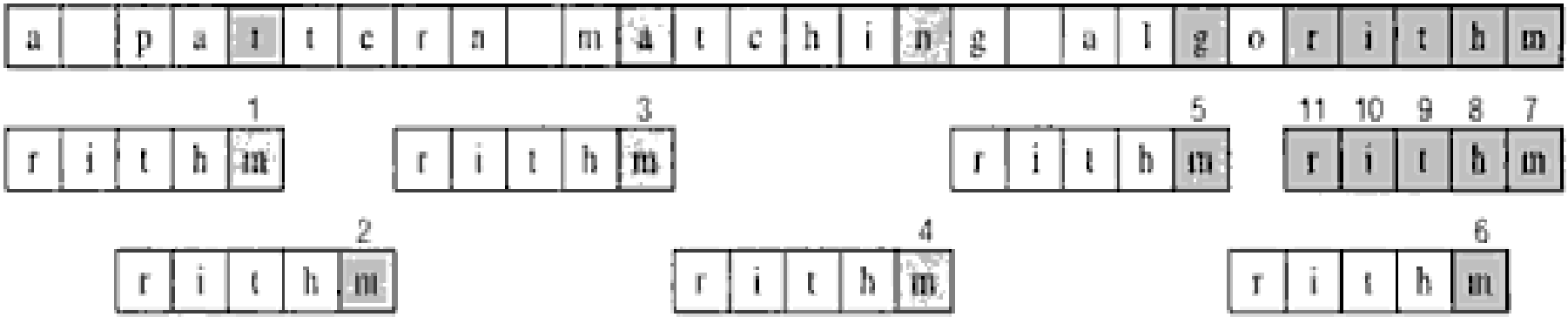
- Heurística do salto de caracteres
  - Imaginando T a sequência, P o padrão buscado e T[i] = c
    - Se c não pertente a P então movemos P completamente para depois de T[i] (salto grande)
    - Se c pertence a P, movemos P para frente até que uma ocorrência do caracter c em P esteja alinhada com T[i]

Encontrando padrões em textos (algoritmo Boyer-Moore)



13 comparações

Encontrando padrões em textos (algoritmo Boyer-Moore)



Exemplo em língua inglesa – Goodrich & Tamassia

Encontrando padrões em textos (algoritmo Boyer-Moore)

- Criar função last(c) para tratar a heurística

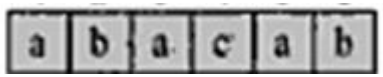
- Se c está em P, last(c) retorna o índice da última (mais a direita) ocorrência de c em P

- Senão, last(c) retorna -1

Exercício:
Pensar e tentar
fazer o algoritmo todo

A função last(*c*)

c	a	b	c	d
last(*c*)	4	5	3	−1



Encontrando padrões em textos (algoritmo Boyer-Moore)

Algoritmo BoyerMoore (T, P)

// T possui n caracteres e P possui m caracteres

```
i ← m − 1
j ← m − 1
repita
    se P [j] = T [i] então
        se j = 0 então
            retorne i        {achamos!}
        senão
            i ← i − 1
            j ← j − 1
    senão
            i ← i + m − min(j, 1 + last(T [i]))        {salto}
            j ← m − 1
até i > n − 1
retorne "Não existe substring em T igual a P."
```

Exercício:
Fazer o chinês