

Algoritmos de Busca em Strings

Brute Force & Rabin Karp

Busca em Strings

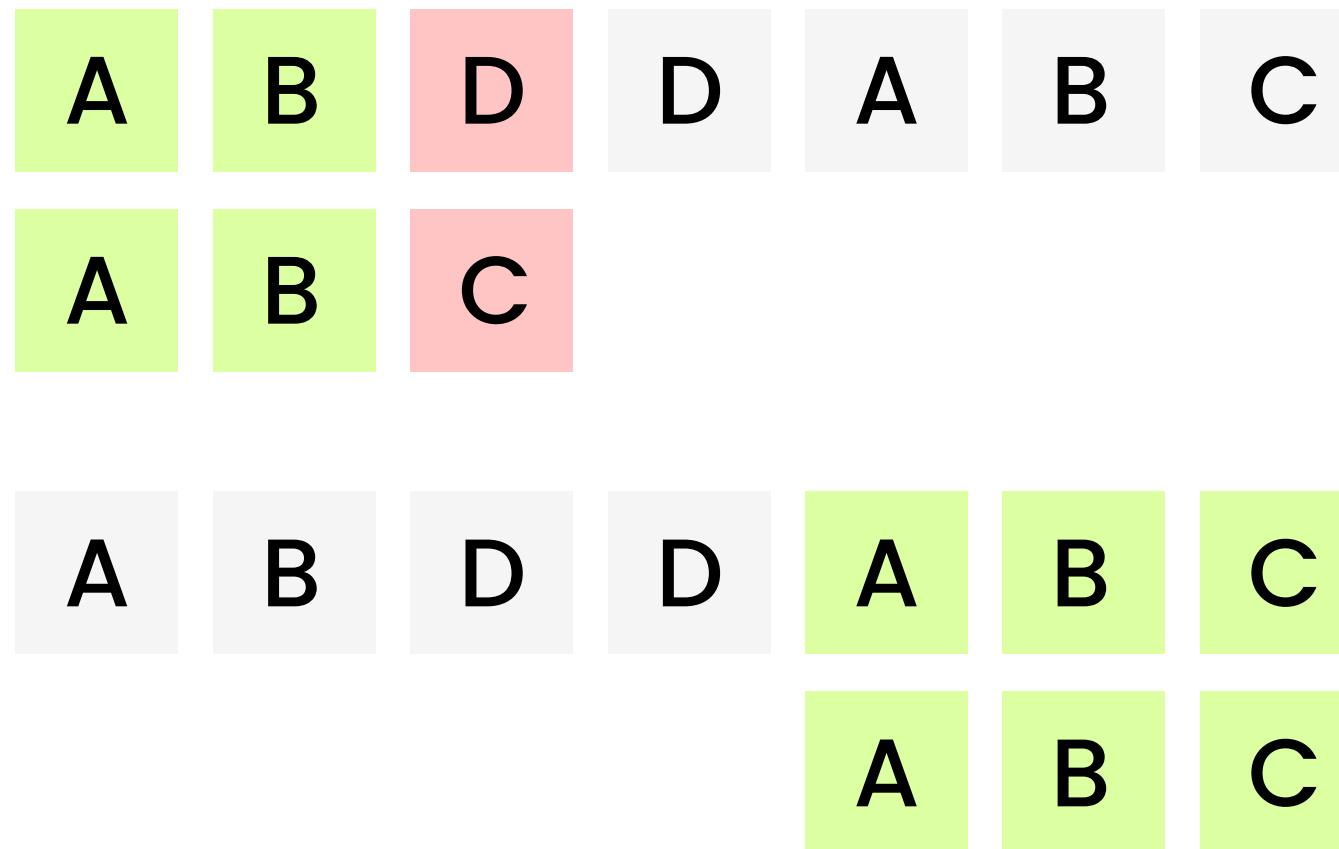
Os algoritmos de busca em string tem como objetivo identificar se um determinado **padrão** existe em uma determinada cadeia de caracteres.

PATTERN: “abc”

TEXT: “abcdabc”

Força Bruta

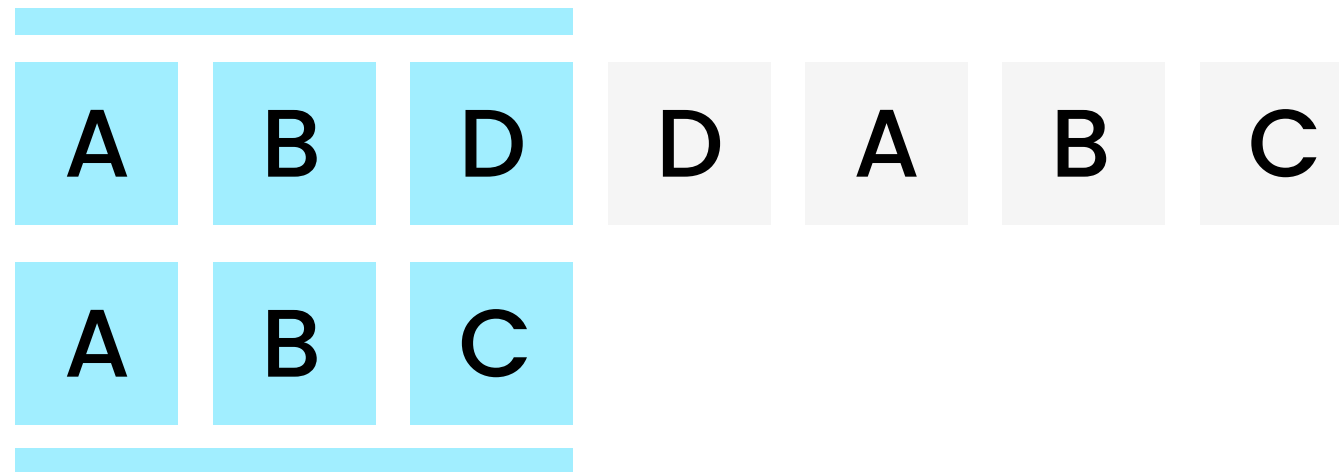
O algoritmo de busca mais simples é conhecido como força bruta. Ele tem como objetivo comparar cada posição do pattern com a janela de caracteres.



Rabin Karp

O algoritmo Rabin Karp é um algoritmo que não exige comparar cada caractere da janela. Isso acontece devido ao fato dele utilizar hashing como método de comparação.

$\text{hash}(\text{"abd"}) = y$



$\text{hash}(\text{"abc"}) = x$

Rabin Karp

A partir do momento que o valor dos hash's batem, é feito então a comparação de cada caracter. Isso deve ser feito devido ao fato de que existe a possibilidade de existirem hash iguais, mas contendo padrões diferentes.

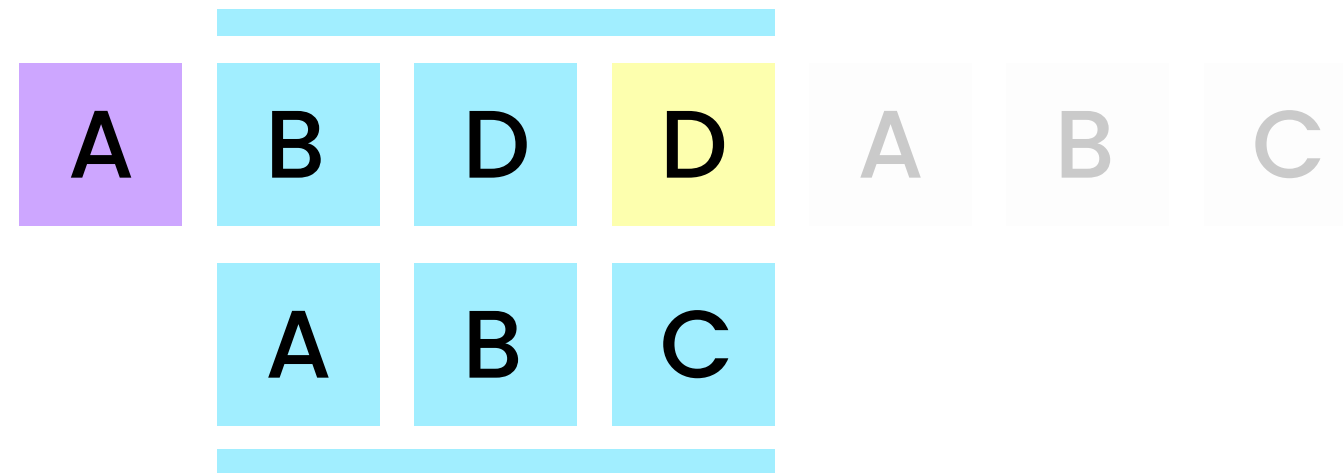
`hash("abd") == hash("abc") ?`





Rabin Karp

Quando os hash's não batem, ou a comparação 1:1 não é correta, uma nova janela é feita e assim também calculado um novo hash para essa mesma janela.

$\text{hash}(\text{"abd"}) \neq \text{hash}(\text{"abc"})$:



 Caracter antigo

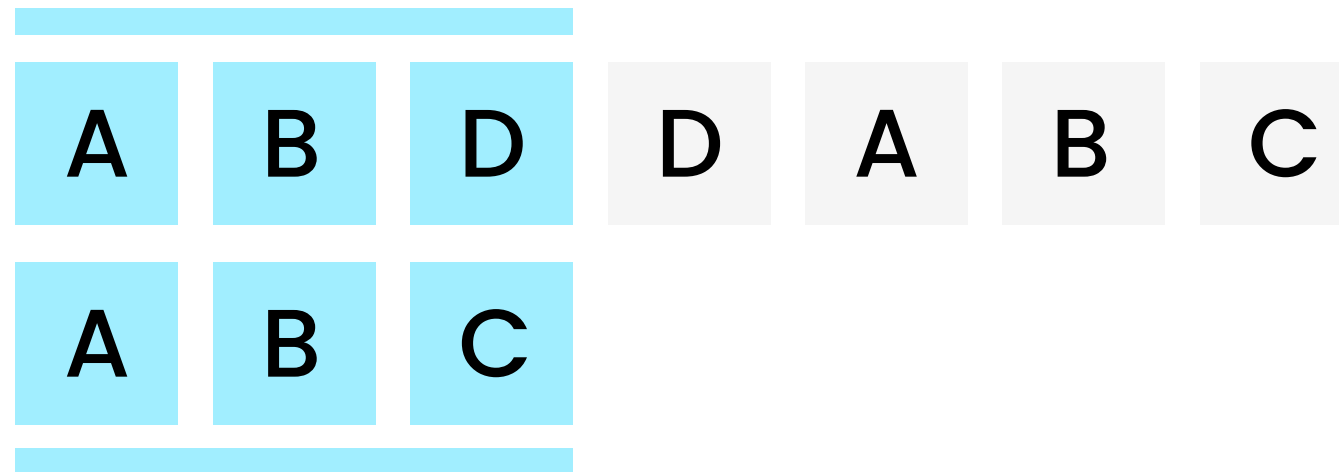
 Caracter novo

Hashing

Na primeira janela, é feito o calculo normal, sendo aplicado as mesmas condições para ambos – janela e pattern.

Primeira janela:

$$\text{hash}(\text{"abd"}) = (\text{base} * \text{hash} + \text{text}[i]) \% \text{prime}$$



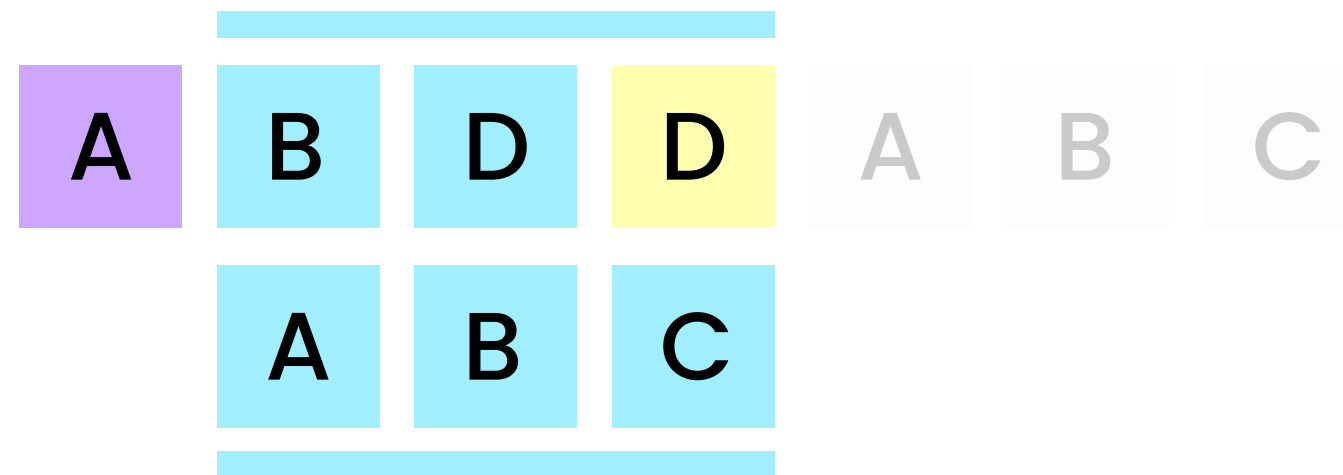
$$\text{hash}(\text{"abc"}) = (\text{base} * \text{hash} + \text{pattern}[i]) \% \text{prime}$$

Hashing



Nas seguintes janelas, ao invés de refazer todo hash da nova janela, é utilizado do caracter antigo, do hash antigo e do novo caracter para recalcular o novo valor.

Janelas seguintes

$$\text{hash}(\text{"abd"}) = (\text{base} * (\text{textHash} - \text{old} * \text{hash}) + \text{new}) \% \text{prime}$$



$$\text{hash}(\text{"abc"}) = (\text{base} * \text{hash} + \text{pattern}[i]) \% \text{prime}$$

-  Caracter antigo
-  Caracter novo


```
printf("the end")
```