

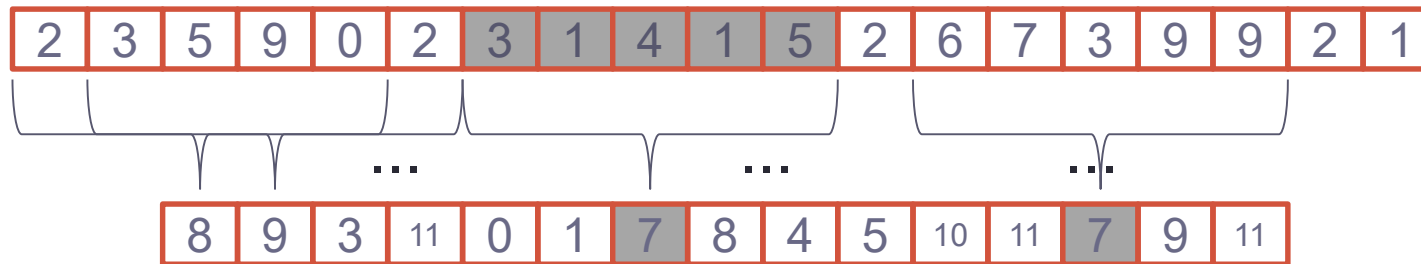
Rabin-Karp

- ✓ Princípio: Transforma o padrão procurado em um número, seguindo determinadas regras.
- ✓ O método é baseado em processar a função de assinatura de cada substring de m-caracteres no texto e checar se é igual a assinatura da função da palavra procurada.
- ✓ O padrão P ocorre no texto T se o valor calculado para P for igual ao valor calculado para qualquer substring X de T , de tamanho m , tal que $|X| = |P|$

Rabin-Karp

- ✓ Cada caractere será um dígito decimal
- ✓ 31415 corresponde ao n° decimal 31.415

Os padrões podem ser texto



Por isso precisamos verifica
a condição de igualdade

Rabin-Karp

- ✓ Acrescentando notação:

p – número correspondente de P ;

t_s – número correspondente de T ;

d – cardinalidade do alfabeto Σ ;

Então cada caractere é um dígito na base d .

q – número primo, como: 16647133;

- ✓ Então temos s válidos se, $p = t_s$.
- ✓ Como calcular p e t_s ?

Rabin-Karp

Com o alfabeto $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ e $|\Sigma| = 10$

P

1	9	9	1
---	---	---	---

Temos:

$$(P[1] * 10 + P[2]) = 19$$

$$(19 * 10) + P[3] = 199$$

$$(199 * 10) + P[4] = 1991$$

Σ = alfabeto

$|\Sigma|$ = tamanho de Σ

Dado um caractere, a representação numérica deste será sua posição no alfabeto

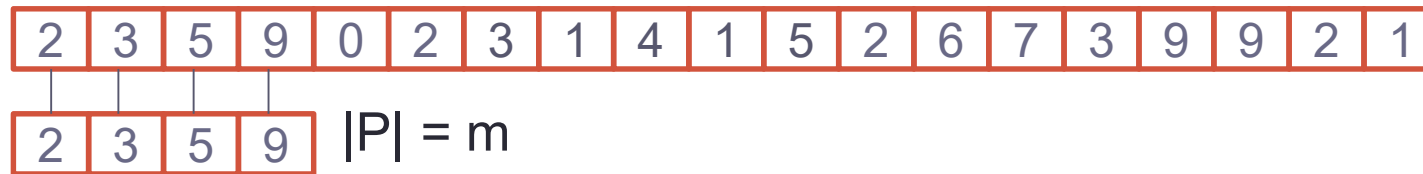
Σ

Generalizando:

$$P[m] + |\Sigma| (P[m-1] + |\Sigma| (P[m-2] + \dots + |\Sigma| (P[2] + |\Sigma| P[1])))$$

Rabin-Karp

- ✓ Realiza pré-processamento do padrão P em tempo $\Theta(m)$



Usando a regra de Horner, podemos calcular o valor de p no tempo $O(m)$

$$P = P[m] + 10(P[m-1] + 10(P[m-2] + \dots + 10(P[2] + 10P[1]) \dots))$$

Comparar P com as m 1ª posições de T .

O t_0 pode ser calculado de modo semelhante, mas os $t_1 \dots t_{n-m}$?