

Felipe Augusto Ferreira de Castro **Matrícula:** 11711BCC033

Fabício Fernandes Ziliotti **Matrícula:** 11711BCC020

Trabalho De Gerenciamento de Banco de Dados

Universidade Federal de Uberlândia

2021

Sumário

Sumário	1
1 FUNÇÃO HASH	2
1.1 Horner's rule	2
2 EXEMPLO DE USO	3
REFERÊNCIAS	4

1 Função Hash

1.1 Horner's rule

A função Hash usada foi a Horner's rule [1]. A qual é definida da seguinte forma.

Definição 1.1.1. seja s uma string de $n+1$ caracteres então a função hash dessa string se da por $H(s)$:

$$H_0(s) = s.charAt(n)$$

$$H_1(s) = X \times H_0(s) + s.charAt(n-1)$$

$$H_2(s) = X \times H_1(s) + s.charAt(n-2)$$

...

$$H_n(s) = X \times H_{n-1}(s) + s.charAt(n)$$

$$H(s) = \sum_{i=n}^0 s.charAt(i) \times X^i$$

Neste trabalho X foi definido como 31. Esta função é interessante por se assemelhar com a função de geração de números aleatórios e também se assemelha com a função hash apresentada em sala de aula.

A função para gerar números pseudo-aleatórios citada é a seguinte:

$$R_0 = seed$$

$$R_1 = x \times R_0 + B$$

$$R_2 = x \times R_1 + B$$

...

$$R_n = x \times R_{n-1} + B$$

Observe que esta função se assemelha em estrutura com a Hash apresentada, pois podemos estabelecer um relação entre cada componente de ambas.

$$R_i \sim H_i$$

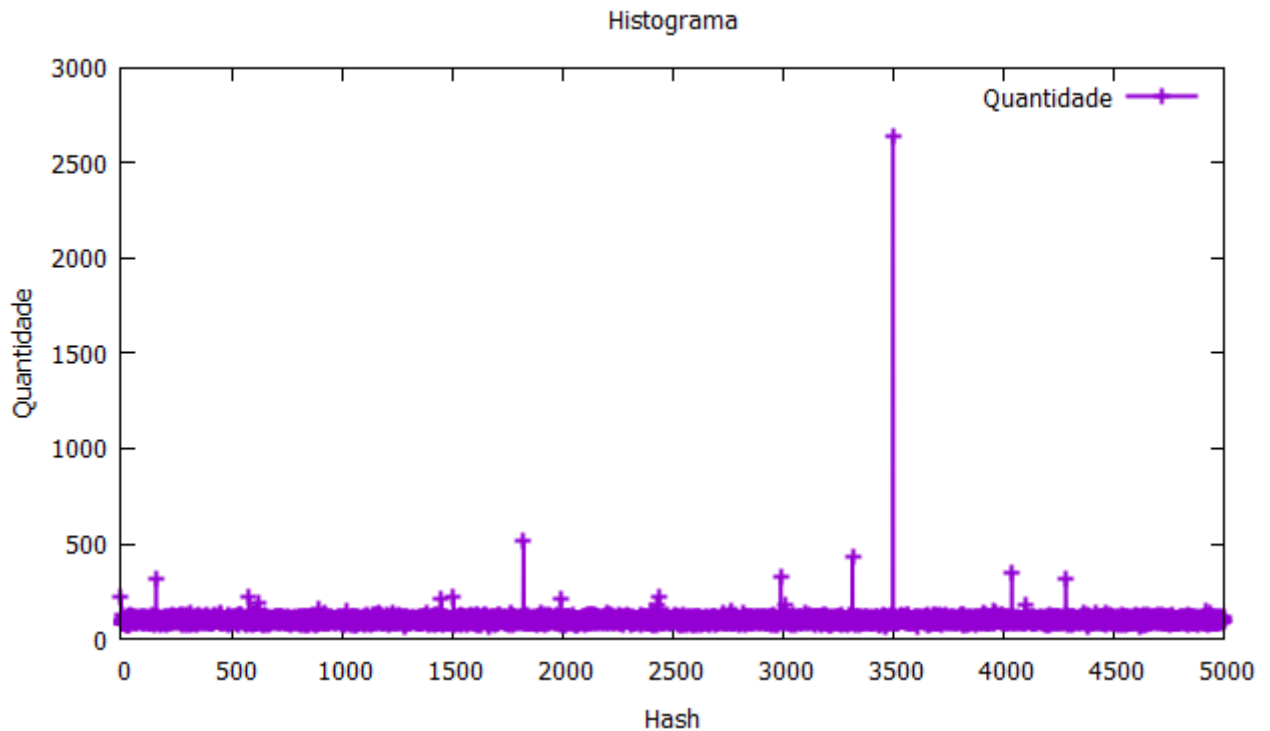
$$x \sim X$$

$$B \sim s$$

Isto pode ser um bom indicador da aleatoriedade atingida pela função hash de Horner.

[1]

2 Exemplo de Uso



Para entendermos devemos levar em consideração que utilizamos 5000 buckets, pois o arquivo de entrada tem 1.000.000.000 de registros e foi dado no enunciado que cada bucket tem 200 itens de tamanho máximo, logo são necessários 5000 buckets para distribuir igualmente todos os registros.

Assim podemos observar que de fato a maioria dos buckets não teve mais de 200 itens, apenas alguns estouraram sendo necessário usar buckets de estouro para poder atendê-los. além disso pelo gráfico podemos observar que a função de fato aparenta ter distribuição uniforme. Os buckets com valores muito maiores podem, provavelmente, atender registros com aparições mais frequentes do que os demais.

Referências

- [1] UCSD, Hash functions for strings. disponível em <https://cseweb.ucsd.edu/~kube/cls/100/Lectures/lec16/lec16-12.html>