



**UNIVERSIDADE FEDERAL DE RORAIMA
CENTRO DE CIÊNCIA E TECNOLOGIA - CCT
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO – DCC
DCC405 – ESTRUTURA DE DADOS II
PROF. MSC. ACAUAN C. RIBEIRO**

Nome(s): Kelvin Araújo Ferreira - 2019037653

Nota: _____

EXERCÍCIO - Aula 11.0 – Tabela Hash

1. Há um resultado matemático surpreendente chamado “paradoxo do aniversário” que afirma que, se há mais de 23 pessoas em uma sala, há mais de 50% de chance de que duas pessoas façam aniversário no mesmo dia. Explique porque este paradoxo é um exemplo do maior problema do hash.

Este paradoxo está diretamente ligado aos problemas de colisões em tabelas hash. Calcular essa probabilidade (e as relacionadas a ela) é o problema do aniversário, ou o “ataque de aniversário” que é nada mais do que uma força bruta. Os algoritmos de encriptação em funções de hash estão expostos a duas vulnerabilidades: ataque de força bruta e a criptoanálise.

O ataque de força bruta depende da quantidade do tamanho do valor da hash em bit. Enquanto uma criptoanálise, ao contrário, se baseia em uma falha específica no algoritmo criptográfico.

Sendo assim, ele não seria bem um grande problema, mas sim um meio utilizado para medir quão seguro é o algoritmo criptografado.

2. Defina com suas palavras o que é uma tabela hash como ela funciona.

Uma tabela hash, é uma estrutura de dados utilizada para tornar o processo de busca mais eficiente. Que associa chaves de pesquisa a valores. Seu objetivo é, a partir de uma chave simples, fazer uma busca rápida e obter o valor desejado, pois o valor do índice se comporta como uma chave para o valor dos dados. Em outras palavras, a tabela de hash armazena pares de valores-chave, mas a chave é gerada por meio de uma função de hash.

3. Quando se utiliza tabelas Hash, pode ocorrer colisão. Explique o que é uma colisão

Uma colisão ocorre quando duas ou mais chaves sinônimas são mapeadas para a mesma posição da tabela.

4. Quando se utiliza tabelas Hash, pode ocorrer colisão. Quais são os métodos de tratamento de colisão? Explique cada um deles com suas palavras.

Boas funções hash podem lidar com colisões através de dois métodos, o método de encadeamento e o método de endereçamento aberto.

Encadeamento: Construir uma lista linear para cada endereço da tabela. Todas as chaves com mesmo endereço são encadeadas em uma lista linear. Assim quando uma nova chave for indexada para um índice que já possui um elemento, o novo elemento deve ir para o fim da lista ligada.

Endereçamento aberto: Todos os elementos são armazenados na própria tabela e sem o uso de listas encadeadas. Logo, o fator de carga não pode exceder o valor 1. O espaço gasto com encadeamento é economizado e a colisão é tratada com a busca de uma posição vazia na própria tabela para inserção.

5. Cite duas características desejáveis quando definimos uma função Hash.

- 1 - Seu cálculo deve ser rápido;
- 2 - Deve gerar poucas colisões.

6. **Explique o método de divisão, usado na criação de funções Hash. Cite um possível problema deste método.**

Método da Divisão: A chave x é dividida pela dimensão da tabela m , e o resto da divisão é usado como endereço chave.

O problema desse método leva a um desempenho ruim, pois as chaves consecutivas são mapeadas para valores de hash consecutivos na tabela de hash.

7. **Discuta as vantagens e desvantagens de se utilizar o método de endereçamento aberto para tratamento de colisões.**

Vantagem: Economiza espaço e trata a colisão na própria tabela para inserção.

Desvantagem: Ele pode ser ineficiente em alguns casos, pois está calculando os mesmos locais várias vezes quando todas as chaves resultam no mesmo endereço de base.

8. **Qual a vantagem da utilização do método de dispersão dupla?**

O hash duplo é um dos melhores métodos disponíveis para endereçamento aberto, porque as permutações produzidas têm muitas características de permutações escolhidas aleatoriamente. É um dos métodos eficazes para resolver colisões.

9. **Demonstre a inserção das chaves 5, 28, 19, 15, 20, 33, 12, 7 e 10 numa tabela de hash com colisões resolvidas por encadeamento. Considere a tabela com $m = 9$ posições e a função hash como sendo $h(k) = k \% m$. Reconstrua a tabela para $m = 11$ (primo) e comente os resultados.**

Para a tabela 1, com inserção onde $m = 9$ teremos a tabela da seguinte forma:

0			0	33
1	28, 19, 10		1	12
2	20		2	
3	12		3	
4			4	15
5	5		5	5
6	15, 33		6	28
7	7		7	7
8			8	19
			9	20
			10	10

Para a tabela 2 temos zero colisões, isso ocorre, pois os slots possíveis que podemos indexar aumentou enquanto que a quantidade de chaves permaneceu igual.

10. **Como excluir itens de uma tabela hash que usa encadeamento para tratar colisões? Quais são as circunstâncias especiais que devem ser tratadas?**

Devemos acessar a tabela na posição do item utilizando a função hash, a partir daí verificamos se este índice possui mais de um elemento vinculado, se sim, devemos então procurar o elemento correto na lista ligada para então removê-lo, caso não haja mais de uma chave ligada ao índice basta realizarmos a operação de remoção normalmente.