

Tabelas de dispersão/hash

1. Suponha uma tabela de hash de tamanho $M = 10$ com *hashing fechado* para armazenar chaves no intervalo $[1, 999]$. Insira as seguintes chaves nessa tabela: 371, 121, 173, 203, 11 e 24, por esta ordem, considerando diferentes métodos de resolução de colisões:

a) Sondagem linear,

- (i) Usando a função hash: $h(k, i) = \text{mod}(k + i, M)$, com $i = 0, 1, \dots$;
- (ii) Remova a chave 121 da tabela;
- (iii) Verifique que a chave 11 está na tabela;
- (iv) Verifique que a chave 175 não está na tabela;
- (v) Insira a chave 252 na tabela.

b) Sondagem quadrática,

- (i) Usando a função hash: $h(k, i) = \text{mod}(k + i^2, M)$, com $i = 0, 1, \dots$;
- (ii) Insira a chave 34 na tabela;
- (iii) Remova a chave 173 da tabela;
- (iv) Verifique que a chave 34 está na tabela.

c) Sondagem quadrática, função hash: $h(k, i) = \text{mod}(k + 2i + i^2, M)$, com $i = 0, 1, \dots$

d) Hash duplo (duas funções de hash): $h_1(k) = \text{mod}(k, M)$, $h_2(k) = 7 - \text{mod}(k, 7)$

Nota: $\text{mod}(a, b) = \text{resto da divisão inteira de } a \text{ por } b$; $\text{mod}(14, 5) = 4$.

2.

- a)** Desenhe uma tabela de hash resultante da introdução das chaves 12, 44, 13, 88, 23, 94, 11, 39, 20, 16 e 5, usando a função de hash $h(k) = \text{mod}(2k+5, 11)$ e supondo que as colisões são tratadas por **hashing aberto**.

b) Implemente a tabela de hash para este caso.

c) Qual seria o resultado se as colisões fossem tratadas por sondagem linear.

d) Mostre o resultado supondo que as colisões são tratadas por sondagem quadrática, até o ponto em que o método falha porque nenhum local vazio é encontrado.

e) Qual o resultado supondo que as colisões são tratadas por hashing duplo usando como função secundária de hash, a função $h'(k) = 7 - \text{mod}(k, 7)$.

3. Considere uma tabela de hash de tamanho $m = 1000$ e a função de hash $h(k) = [m \cdot (k \cdot A)]$, com $A = 0.618$. Calcule os valores de hash das chaves 61, 62, 63, 64 e 65.

4. Considere o método da divisão para criar funções de hash. Considere que o universo das chaves, U , é o conjunto dos números inteiros não negativos. Se o tamanho da tabela for $m = 177$, encontre 3 chaves diferentes que tenham o mesmo valor de hash. Ache uma fórmula que, para qualquer chave k , lhe permita construir tantas chaves quantas desejar com o mesmo valor de hash de k .

5. Implemente uma tabela de hash com 3 posições, com resolução de colisões por *hashing aberto*, para o problema do exercício anterior. Para tal,
- Use para criar a função de hash o método da divisão. Qual é o fator de ocupação?
 - Escreva uma função que devolva a média de idades duma tripulação. Uma vez que necessária a consulta a todos os elementos armazenados, acha que a tabela de hash é uma boa escolha como estrutura de dados?
 - Implemente a tabela de hash para este caso.
6. Sabendo que uma tabela de hash usa para resolver colisões uma função de hash $h(k, i)$ quadrática com $m = 13$, $c_1 = 5$ e $c_2 = 3$, construa a tabela para a seguinte sequência de chaves: 10, 25, 37, 38, 26, 36 e 18.
7. Sabendo que uma tabela de hash usa, para resolver colisões, uma função de hash linear com $m = 13$, construa a tabela para a seguinte sequência de chaves: 10, 25, 37, 38, 26 e 36.
8. Utilizando a função de hash $h(k) = \text{mod}(k, 11)$, insira na tabela de hash a seguinte sequência de chaves (k): 82, 31, 28, 4, 45, 27, 59, 79 e 35.
9. Realize as mesmas operações do exercício anterior variando com os seguintes métodos:
- Rehash linear — $h'(k, i) = \text{mod}(k + a * i, 11)$, com $a = 1$ e $i \geq 1$, variando iterativamente a cada tentativa de se resolver a colisão.
 - Rehash quadrático — $h'(k) = \text{mod}(k + a * i^2, 11)$, com $a = 1$ e $i \geq 2$ (pois $i = 1$ recai no rehash linear).
 - external chaining — criar uma lista ligada em cada posição do tabela hash.
- 10.
- Desenhe a tabela de hash com 11 elementos resultante da aplicação da função de hash $h(k) = \text{mod}(3 * k + 5, 11)$, para inserir as seguintes chaves: 12, 44, 13, 88, 23, 94, 11, 39, 20, 16 e 5. Assuma que as colisões serão tratadas por *hashing aberto*.
 - Qual o resultado, assumindo que as colisões serão tratadas por sondagem linear?
 - Qual o resultado, assumindo que as colisões são tratadas por sondagem quadrática?
 - Qual o resultado, assumindo que as colisões são tratadas por hashing duplo usando uma função de hash secundária $h'(k) = 7 - \text{mod}(k, 7)$?
11. Implementar uma tabela de hash estático utilizando a função de hash $h(k) = \text{mod}(5 * k, 8)$.
12. Gerar a chave para a palavra "COVILHA", considerando os três tipos de *rehash* seguintes:
- Linear: $r = \text{mod}(2 * k + i, 6)$
 - Quadrático: $r = \text{mod}(k + i^2, 6)$
 - Duplo: $r_1 = \text{mod}(k + i^2, 6)$ e $r_2 = \text{mod}(2 * k + i, 6)$

- 13.** O uso de tabelas de hash com *hashing aberto* facilita a operação de remoção, pois apenas é preciso remover o elemento da lista ligada. No caso de *hashing fechado* a remoção deve ser feita de maneira diferente.
- a)** Qual operação é realizada sobre a chave removida?
- b)** Em caso de existirem muitos elementos marcados como removidos na tabela, perde-se a eficiência nas pesquisas; como é possível resolver este problema?
- 14.** Elabore uma função que traduza uma inserção numa tabela de hash que usa sondagem quadrática para resolver colisões, supondo que também se usa o truque de substituir os itens removidos por um objeto especial "desativado".
- 15.** Faça um algoritmo de inserção numa tabela de hash que insere uma chave k numa tabela T de inteiros utilizando a abordagem de *hashing fechado* para tratar colisões. Deve-se percorrer a tabela de forma circular.
- 16.** Faça um algoritmo de inserção numa tabela de hash que insere uma chave k numa tabela T de inteiros utilizando a abordagem de rehashing para tratar colisões. Deve-se percorrer a tabela de forma circular.
- 17.** Demonstre a inserção das chaves 5, 28, 19, 15, 20, 33, 12, 7 e 10 numa tabela de hash com colisões resolvidas por *hashing aberto*. Considere a tabela com $m = 9$ posições e a função hash como sendo $h(k) = \text{mod}(k, m)$. Reconstrua a tabela para $m = 11$ (primo) e comente os resultados.
- 18.** Considere uma tabela de hash $T[0..5]$, inicialmente vazia. Mostre os estados intermediários da tabela após a inserção de cada uma das seguintes chaves: 94, 19, 125, 61 e 40. Considere $h(k, i) = \text{mod}(k + i, 6)$ (*linear probing*).
- 19.** Desenhe a sequência de configurações da tabela de hash (de tamanho 7) obtida através da função hash dada por $h(k) = \text{mod}(k, 7)$, e colisões são resolvidas por *double hashing*, com a segunda função $h(y) = \text{mod}(y, 3) + 1$, quando as seguintes chaves são inseridas nesta ordem a partir da tabela vazia: 42, 56, 63, 70. Mostre a configuração após cada inclusão, indicando claramente onde ocorreu colisão.
- 20.** Considere a inserção das chaves 10, 22, 31, 4, 15, 28, 17, 88 e 59 numa tabela de hash de comprimento $m = 11$, usando *hashing fechado* para resolver as colisões com a função de hash primária $h(k) = \text{mod}(k, m)$. Mostre o resultado da inserção destas chaves na tabela usando:
- sondagem linear;
 - sondagem quadrática com $c_1 = 1$ e $c_2 = 3$;
 - hash duplo com $h_2(k) = 1 + \text{mod}(k, m-1)$

- 21.** Suponha uma tabela de hash de tamanho 10 que utiliza a função de hash $h(k) = \text{mod}(k, 10)$. Mostre a tabela após inserir as seguintes chaves pela ordem apresentada {18, 19, 29, 11, 20, 21, 28} utilizando:
- a)** resolução de colisões com listas ligadas;
 - b)** resolução de colisões com pesquisa linear;
 - c)** resolução de colisões com pesquisa quadrática ($c_1 = 2$ e $c_2 = 1$);
 - d)** resolução de colisões utilizando a função hash $h_2(k) = 5 - \text{mod}(k, 5)$.
- 22.** Numa tabela de hash com 100 entradas, as colisões são resolvidas usando listas encadeadas. Para reduzir o tempo de pesquisa, decidiu-se que cada lista seria organizada como uma árvore binária de pesquisa. A função utilizada é $h(k) = \text{mod}(k, 100)$. Infelizmente, as chaves inseridas seguem o padrão $k_i = 50 * i$, onde k_i corresponde à i -ésima chave inserida.
- a)** Mostre a situação da tabela após a inserção de k_i , com $i = 1, 2, \dots, 13$. (Faça o desenho).
 - b)** Depois de 1000 chaves serem inseridas de acordo com o padrão acima, inicia-se a inserção de chaves escolhidas de forma aleatória (isto é, não seguem o padrão das chaves já inseridas). Assim responda: Qual é a ordem do pior caso (isto é, o maior número de comparações) para inserir uma chave?
- 23.** Suponha um conjunto de n chaves x formado pelos n primeiros múltiplos do número 7. Quantas colisões seriam obtidas mediante a aplicação de cada uma das funções de hash que se seguem? Mostre como chegou nas suas respostas.
- a)** $\text{mod}(x, 7)$
 - b)** $\text{mod}(x, 14)$
 - c)** $\text{mod}(x, 5)$
- 24.** Quais as vantagens e desvantagens (com relação à criação da estrutura, inserção e remoção e pesquisa de elementos) de cada uma das estruturas abaixo, para o caso em que se deseja armazenar um dicionário com chaves compostas por strings com tamanho até 80 caracteres e as suas definições compostas por strings de tamanho até 2000 caracteres. Sabe-se que a quantidade de palavras pode variar entre 10.000 e 50.000.
- a)** Arranjo e pesquisa binária
 - b)** Lista encadeada e pesquisa sequencial
 - c)** Tabela de hash com *hashing fechado* e função de hash duplo
 - d)** Tabela hash com *hashing aberto*
- 25.** Considere uma empresa que tem o seu número de clientes limitado ao máximo de 1000. No entanto, o código do cliente (chave) é um número que começa em 4841200001 e termina em 4841201000. Como seria possível utilizar uma tabela de hash, com vetores, para implementar tal situação?