Tabelas de dispersão/hash

1. Considere uma tabela de hash de tamanho m = 1000 e a função de hash h(k)= [m.(k.A % 1)], com A=(√5 -1)/2. Calcule os valores de hash das chaves 61, 62, 63, 64 e 65.

2. 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.

3. Implemente uma tabela de hash usando endereçamento directo (hashing fechado) para o seguinte problema. Uma companhia aérea tem 200 tripulantes que são identificados por um inteiro entre 1 e 200 (código de tripulante). Queremos uma tabela de hash que permita guardar a constituição da tripulação que segue num determinado avião. O programa deve ler do ficheiro "aviao.txt" a tripulação para um voo e guardar a informação na tabela de hash.
Deve depois permitir responder à questão "O tripulante k viaja neste voo?" (onde k é um código de tripulante). Se a resposta for afirmativa deve mostrar os seus dados.
As estruturas que guardam a informação são do tipo seguinte:

```
struct {
   int codigo, idade;
   char nome[100];
 }
```

4. Implemente uma tabela de hash com 3 posições, com resolução de colisões por encadeamento (hashing fechado), para o problema do exercício anterior.

- **a)** Use para criar a função de hash o método da divisão. Qual é o fator de ocupação?
- **b)** Escreva uma função que devolva a média de idades duma tripulação. Sendo necessária a consulta a todos os elementos armazenados, acha que a tabela de hash é uma boa escolha como estrutura de dados?

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

- **a)** Sondagem linear, função hash: h(k) = k%M + i
- **b)** Sondagem quadrática, função hash: h(k) = k%M + i^2
- **c)** Sondagem quadrática, função hash: h(k) = k%M + 2i + i^2
- **d)** Hash duplo, função hash: $h_1(k)$ = k%M, função hash 2: $h_2(k)$ = 7 – (k%7)

6. Crie uma tabela de hash com registos para as cidades de Portugal. A chave de cada registo será o nome da cidade e não são necessários outros campos para este exercício. Comece por colocar os nomes destas cidades por ordem alfabética.

a) Examine a lista ordenada. Que padrões se nota que podem afetar escolha de uma função de hash?

b) Implemente uma função hash() que utilize alguma combinação dos códigos ASCII das letras do nome das cidades, mas de forma que se possa alterar o número de caracteres que são utilizados na combinação. Executar o hash() várias vezes, utilizando sempre um número diferente de caracteres e produzindo as seguintes estatísticas para cada execução:

- O número de colisões;
- O número de endereços com 0, 1, 2, 3, …, 10, ou mais de 10 cidades associadas.

Discutir os resultados dos testes em termos de efeitos da escolha de diferentes quantidades de caracteres e como eles se relacionam com o resultado que se poderia esperar de uma distribuição aleatória.

(Implemente e teste um ou mais dos métodos de hash descritos nas aulas, ou use um método inventado por si).

7. Utilizando os resultados do exercício anterior, responda:

a) Qual o fator de ocupação final da tabela de hash?

b) Quais as estratégias que aparentemente realizaram melhor espalhamento das chaves?

c) Como ficaria o mesmo cenário utilizando uma tabela de hash de tamanho 7 com encadeamento?

8. O uso de tabelas de hash com encadeamento (hashing aberto) facilita a operação de remoção, pois apenas é preciso remover o elemento da lista ligada. No caso de endereçamento aberto (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?

9. 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) = (2k+5) % 11 e supondo que as colisões são tratadas por encadeamento (hashing aberto).

10. Qual seria o resultado do exercício anterior se as colisões fossem tratadas por sondagem linear.

11. Mostre o resultado do exercício 10 supondo que as colisões são tratadas por sondagem quadrática, até o ponto em que o método falha porque nenhum local vazio é encontrado.

12. Qual o resultado do exercício 10 supondo que as colisões são tratadas por hashing duplo usando como função secundária de hash, a função h'(k) = 7 – (k % 7).

13. Elabore uma função em C 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".

14. Faça um algoritmo de inserção numa tabela de hash que insere uma chave k numa tabela T de inteiros utilizando a abordagem de endereçamento aberto (hashing fechado) para tratar colisoes. Deve-se percorrer a tabela de forma circular.

15. 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 colisoes. Deve-se percorrer a tabela de forma circular.

16. Demonstre a inserção das chaves 5, 28, 19, 15, 20, 33, 12, 7 e 10 numa tabela de hash com colisões resolvidas por encadeamento (hashing aberto). 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.

17. 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,t) = (k+t) % 6 (*linear probing*).

18. Desenhe a sequência de configurações da tabela de hash (de tamanho 7) obtida através da função hash dada por h(k) = k % 7, e colisões são resolvidas por *double hashing*, com a segunda função h(y) = (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.

19. Considere a inserção das chaves 10, 22, 31, 4, 15, 28, 17, 88 e 59 numa tabela de hash de comprimento m=11, usando endereçamento aberto (hashing fechado) para resolver as colisões com a função de hash primária h(k)= 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 $h_2(k) = 1 + (k \% (m-1))$

20. Suponha uma tabela de hash com 10 posições que utiliza a função de hash h(k)=k % 10. Mostre a tabela após inserir as seguintes chaves {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 $h_2(k) = 5 - (k \% 5)$.

21. 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 binaria de pesquisa. A função utilizada é h(k) = k % 100. Infelizmente, as chaves inseridas seguem o padrão $k_i = 50i$, 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,...,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 ja inseridas). Assim responda: Qual é a ordem do pior caso (isto é, o maior número de comparaçoes) para inserir uma chave?

22. Suponha um conjunto de n chaves x formado pelos n primeiros múltiplos do número 7. Quantas colisoes seriam obtidas mediante a aplicação de cada uma das funçoes de hash que se seguem? Mostre como chegou nas suas respostas.

a) x % 7

b) x % 14

c) x % 5

23. 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 endereçamento aberto (hashing fechado) e hashing duplo

d) Tabela hash com encadeamento (hashing aberto)

24. Considere uma empresa que tem o seu número de clientres 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?

25. Sabendo que uma tabela de hash usa para resolver colisões uma função de hash h(k,i) dupla com m = 13 para h(k,i) e $h_1(k)$, e m = 11 para $h_2(k)$, construa a tabela para a seguinte sequência de chaves: 10, 24, 23, 49 e 36.

26. 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.

27. Sabendo que uma tabela de hash usa para resolver colisões uma função de hash h(k,i) linear com m=13, construa a tabela para a seguinte sequência de chaves: 10, 25, 37, 38, 26 e 36.

28. Utilizando a função de hash: h(key) = key % 11, insira na tabela de hash a seguinte sequência de chaves: 82,31,28,4,45,27,59,79 e35.

29. Realize as mesmas operações do exercício anterior variando com os seguintes métodos:

a) Rehash linear — h'(k,i) = (k+a*i) % 11, com a = 1 e i ≥ 1, variando iterativamente a cada tentativa de se resolver a colisão;

b) Rehash quadrático — h'(k)=(k+a*i^2) % 11, com a = 1 e i ≥ 2 (pois i = 1 recai no rehash linear;

c) external chaining — criar uma lista ligada em cada posição do tabela hash.

30. Para cada uma das estratégias de resolução de colisão do problema anterios determinar:

a) Fator de ocupação;

b) Média do número de testes para encontrar a chave corrente na tabela

31. Implementar um dicionário utilizando uma tabela de hash com listas de colisões. Considere que a tabela tem tamanho 26 (uma posição para cada letra do alfabeto).

32. Desenhe a tabela de hash com 11 elementos resultante da aplicação da função de hash h(k)=(3k+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 encadeamento.

33. Qual será o resultado do exercício anterior se assumirmos que as colisões serão tratadas por sondagem linear?

34. Qual o resultado do exercício 32, assumindo que as colisões são tratadas por sondagem quadrática.

35. Qual o resultado do exercício 32, assumindo que as colisões são tratadas por hashing duplo usando uma função de hash secundária h'(k)=7 - (k % 7).

36. Implementar uma tabela de hash estático utilizando a função de hash h(k) = (5k) % 8.

37. Gerar a chave para a palavra "COVILHA", considerando os três tipos de *rehash* seguintes:

a) Linear: r = (2k + i) % 6

b) Quadrático: r = (k + i^2) % 6

c) Duplo: r_1 = (k + i^2) % 6 e r_2 = (2k + i) % 6