

1) Responda às seguintes questões sobre tabelas hash:

a) Em que situações a tabela hash deve ser utilizada?

Quando houverem muitos valores a serem adicionados

b) Quais as características de uma boa função de hash?

Eficiência e bom espalhamento (não apresentar uma grande quantidade colisões e de clusters)

c) O que é uma colisão em uma tabela hash?

É quando um dado tem a mesma chave de um outro anteriormente adicionado a tabela hash

d) Caracterize o endereçamento aberto e encadeamento separado.

No endereçamento aberto, a colisão é tratada diferente, pois nele, números que foram colididos não permanecem no mesmo “espaço”, eles são realocados para outro espaço na tabela que esteja vazio, já no endereçamento separado, cria-se uma espécie de lista onde se há colisão.

e) O que fazer quando uma tabela hash torna-se cheia demais?

Um novo vetor maior terá que ser criado e então inserir o conteúdo do antigo vetor pequeno naquele novo grande.

2) Suponha uma tabela hash armazenar valores de chaves. Insira as seguintes chaves nessa tabela: 36, 53, 70, 87, 54, 37, 71 e 40, nessa ordem. Considere os diferentes métodos:

a) Endereçamento Aberto (tabela hash de tamanho 17)

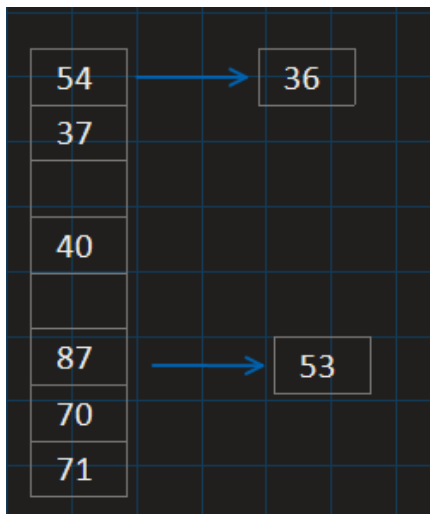
• Exploração linear: $\text{Passo}(k) = k \% \text{arraySize} + 1$

36	54	37	71	40	87	70	53
----	----	----	----	----	----	----	----

• Hash duplo: $h1(k) = k \% \text{arraySize}$; $h2(k) = 5 - (k \% 5)$

37	71	54	36	53	70	87	40
----	----	----	----	----	----	----	----

b) Encadeamento separado (tabela hash de tamanho 5)



3) Dependendo da implementação, quando se permitir que itens de dados com chaves duplicadas sejam usadas em tabelas hash, apenas o primeiro item de dados pode ser acessado.

a) Descreva com suas palavras como resolver esse problema

Para solucionar esse problema, pode haver a criação de uma tabela maior, ou a utilização do endereçamento separado, que irá “permitir” que tenham dados com as mesmas chaves.

4) Fazendo operações na tabela hash abaixo, preencha a tabela que se segue utilizando o duplo

hashing. Tem-se que:

- $h_1(k) = k \% \text{arraySize}$;
- $h_2(k) = 5 - (k \% 5)$.

51		*	85	4	39	*	24	42		27		38	20		32	*
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Nº do Item	Operação	Valor do Hash	Tamanho do Passo	Células na Sequência de Exploração	Status da Operação
1	BUSCA (39)	5	1	---	Encontrado
2	INSERE (32)	15	3	---	Inserido
3	BUSCA (85)	0	5	5 10 15 3	Encontrado
4	INSERE (20)	3	5	8 13	Inserido
5	REMOVE (27)	10	3	---	Removido
6	BUSCA (28)	11	2	---	Falhou

7	BUSCA (19)	2	1	---	Falhou
8	REMOVE (4)	4	1	---	Falhou
8	BUSCA (34)	0	1	---	Falhou
9	BUSCA (37)	3	3	---	Falhou
10	BUSCA(44)	10	1	---	Falhou