Professor
Marcio Feitosa

CURSO: Estruturas de Dados

TAD TABELA HASHING (ou Tabela de Hash)

A **Tabela Hashing** é um modelo de armazenamento de dados que, se bem dimensionado, otimiza o espaço em memória e garante rapidez no armazenamento de novos dados e na localização dos já armazenados.

A tabela se baseia em um vetor cujos índices correspondem às chaves de identificação de cada elemento armazenado. A chave é calculada por uma função, chamada função de hashing.

A tabela hashing não foi idealizada para armazenamento de elementos duplicados, portanto cada elemento deve ter um identificador único, numérico inteiro ou alfanumérico.

A função de hashing, para cálculo de chaves, mais utilizada para identificadores numéricos é a do cálculo do resto da divisão do identificador do elemento pelo tamanho da tabela.

Exemplo:

- Identificador = 341; tamanho da tabela = 25
- 341 mod 25 = 16
    - o elemento será armazenado na posição 16 do vetor.

Para os identificadores alfanuméricos pode-se utilizar a soma dos valores ascii dos caracteres da string.

Exemplo:

o código ATX-45 geraria um valor de 65+84+88+45+52+53=387. Feito isso entra-se na função para cálculo de valores inteiros. Aqui cabe observar que identificadores não costumam ter muitos caracteres (raramente excedem 20), de forma que a obtenção da somatória ASCII não deve demandar muito processamento.

Colisões: ocorrem quando os identificadores de dois elementos distintos geram a mesma chave de armazenamento. Por exemplo, o elemento (exemplo anterior) de código ATX-45 e outro de código ATX-54. A soma dos valores ASCII resultará no mesmo valor e, portanto, os dois deverão ser armazenados no mesmo local, o que não será possível. A este fato de dá o nome de colisão e uma solução deverá ser dada.

Professor
Marcio Feitosa

Estratégias de armazenamento - as mais utilizadas são duas:

A) Endereçamento aberto:

- **Sondagem linear:**

Os elementos são armazenados no vetor e o limite de armazenamento é o tamanho do vetor. Estando o vetor lotado, o próximo que chegar ficará em uma fila até vagar um lugar. Essa estratégia é utilizada quando os elementos são armazenados de forma temporária, entrando e saindo ao longo do tempo[1].

Havendo colisão, procura-se a próxima posição vaga e armazena-se aí o novo elemento. Neste último caso o novo elemento não ficou na posição que deveria, pois estava ocupada. Armazenou-se na próxima vaga, que não é a dele. O próximo que chegar, caso seja este o endereço de direito, terá que procurar o próximo vago.

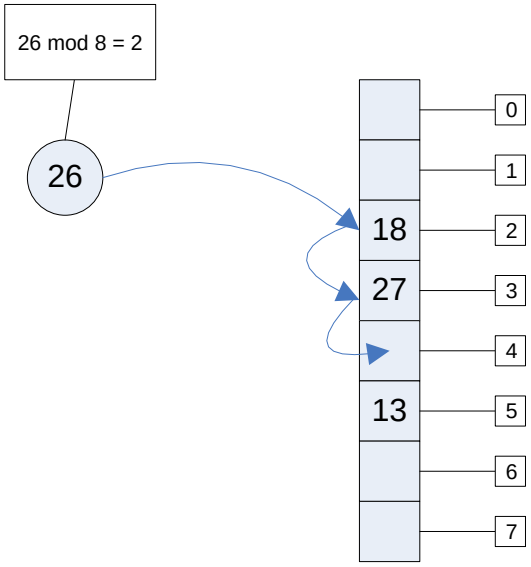

Figura 1

Essa estratégia é a que mais faz jus ao nome. No idioma inglês, hash quer dizer confusão, coisa picotada, erro grave.

[1] A pilha e a fila também costumam ser utilizadas para armazenamento temporário, mas a filosofia de extração é diferente da tabela hashing. Na pilha e na fila o atendimento se baseia na ordem de chegada e na tabela hashing pela identificação do elemento, independentemente da sua ordem de chegada (que, aliás, nem é registrada).

Há algumas variações para busca da próxima posição vaga. Basicamente adota-se uma fórmula para cálculo desta posição:

- **Sondagem quadrática** - $h(k,i) = [h(k) + i^2] \, mod \, n$

Onde o fator i é o salto em posições, sendo que na primeira tentativa o valor é 1, na segunda é 2, na terceira é 3 e assim sucessivamente até encontrar uma posição vaga.

1. Na primeira tentativa salta $1^2 = 1$
2. Na segunda tentativa salta $2^2 = 4$
3. Na terceira tentativa salta $3^2 = 9$
4. ... e assim por diante até encontrar vaga.

- **Duplo hash** - $h(k,i) = [h(k) + i \times h'(k)] \, mod \, n$

Onde h'(k) é uma segunda função hash.

B) Encadeamento:

Aqui os elementos são armazenados no vetor da mesma forma que na estratégia de endereçamento aberto, porém, ao haver colisões, passa-se a encadear os elementos em listas simplesmente encadeadas (listas ligadas). Dessa forma, teoricamente, não há um limite para armazenamento e é um modelo bastante funcional para armazenamentos, em memória, de mais longo prazo.

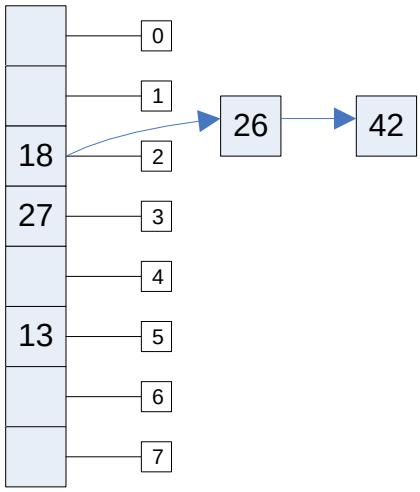

Figura 2

Professor
Marcio Feitosa

Para armazenamento de volumes extremamente grandes utiliza-se a extensão da chave com árvores binárias.

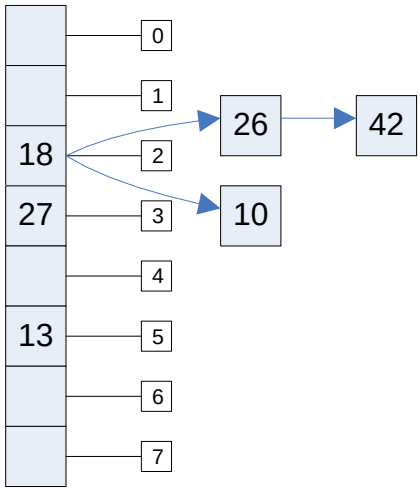

Figura 3

-o-o-o-o-o-o-o-o-o-o-o-o-