

Exercícios de fixação 07 - Tabelas hash extensíveis

Entrega 20 out em 23:59

Pontos 1

Perguntas 4

Limite de tempo Nenhum

Instruções

Este questionário contém questões sobre tabelas *hash* extensíveis.

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	Tentativa 1	4 minutos	1 de 1

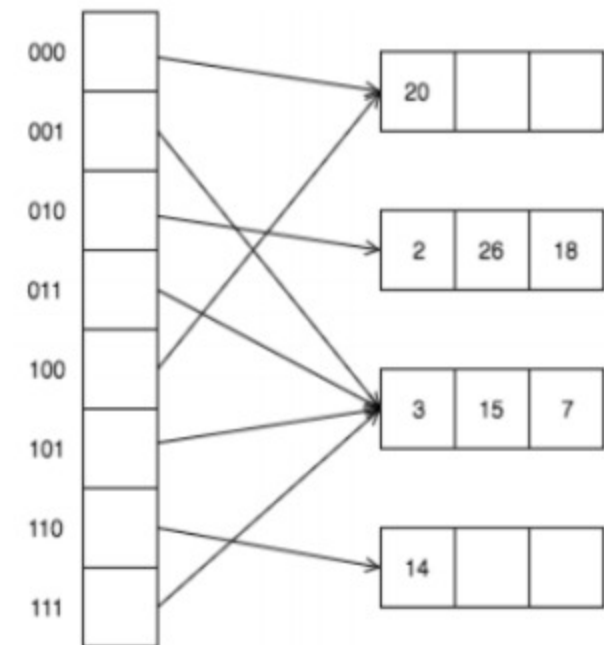
Pontuação deste teste: **1** de 1
Enviado 20 out em 17:15
Esta tentativa levou 4 minutos.

Detalhes do envio:	
Tempo:	4 minutos
Pontuação atual:	1 de 1
Pontuação mantida:	1 de 1

Pergunta 1

0,25 / 0,25 pts

Considere a seguinte tabela *hash* extensível:



Suponha que façamos a inserção das chaves 30, 21, 12, 42 e 31, nesta ordem. Que chave provocará o primeiro aumento de profundidade do diretório?

Correto!

42

42 (com margem: 0)

Considerando a função *hash* $h(x) = x \bmod 2^p$, a chave 30 caberá no quarto *bucket*. A chave 21 será alocada para o terceiro *bucket*, provocando a sua divisão, mas esse *bucket* tem profundidade local menor que a profundidade global, então não promove a duplicação do diretório. A chave 12 ficará no primeiro *bucket*. A chave 42 será direcionada para o segundo *bucket*, que deve ser dividido. No entanto, como a sua profundidade local é igual à profundidade global, será necessário o aumento do diretório.

Pergunta 2

0,25 / 0,25 pts

Qual é a vantagem do uso de cestos (*buckets*) nas tabelas *hash* em disco?

- ☐ Permitir a criação de funções *hash* mais eficientes.
- ☒ Aumentar a eficiência no armazenamento e na recuperação dos dados em disco.
- ☐ Criar uma tabela *hash* de tamanho variável.
- ☐ Reduzir o tamanho da tabela e, assim, o espaço em disco ocupado por ela.

Correto!

Os cestos (*buckets*) são estruturas que comportam um conjunto de elementos em um único endereço de uma tabela *hash*. Isso reduz, proporcionalmente, o número de colisões. Dessa forma, acelera o armazenamento de elementos, como também otimiza a leitura, uma vez que facilita a recuperação mais ágil dos elementos. É importante considerar que a leitura de um *bucket* ou de um único elemento geralmente envolve o mesmo custo de acesso a disco.

Pergunta 3

0,25 / 0,25 pts

Em uma tabela *hash* extensível, é possível afirmar que:

- ☐ A cada vez que o diretório duplica seu tamanho, os *buckets* também são duplicados.
- ☒ Quando a tabela cresce, apenas as entradas no *bucket* dividido precisam ser reposicionadas.
- ☐ Os *buckets* são acessados por meio de em uma lista encadeada de *buckets*.
- ☐ A divisão de um *bucket* de profundidade menor que a profundidade do diretório não provoca qualquer mudança no diretório.

Correto!

A tabela *hash* extensível pode crescer ou diminuir de acordo com a necessidade. Quando um *bucket* é dividido, o diretório deve ser atualizado (duplicado, se necessário) e apenas as entradas desse *bucket* dividido precisam ser reposicionadas, pois podem ficar tanto no *bucket* antigo quanto no *bucket* recém-criado.

Pergunta 4

0,25 / 0,25 pts

O que pode ser calculado a partir da profundidade do diretório de uma tabela *hash* extensível?

- ☐ A quantidade de *buckets* presentes na tabela *hash*.
- ☐ A quantidade máxima de elementos em cada *bucket*.
- ☐ A quantidade de elementos inseridos na tabela *hash*.
- ☒ A quantidade de endereços no diretório.

Correto!

O diretório de uma tabela *hash* é uma estrutura que aponta para cada *bucket*. É importante, portanto, que tenha uma quantidade maior ou igual de endereços que a quantidade de *buckets* existentes na tabela. Esses endereços (ponteiros para os *buckets*) são calculados a partir da profundidade do diretório. Assim, um diretório de profundidade 3 consegue apontar para até $2^3 = 8$ *buckets*. O número de *buckets*, porém, pode ser inferior ao número de endereços, pois o ponteiro de um *bucket* pode ser usado em mais de um endereço.

Pontuação do teste: **1** de 1