


Revisar envio do teste: Semana 4 - Atividade Avaliativa

| | |
|------------------------|---|
| Usuário | LIZIS BIANCA DA SILVA SANTOS |
| Curso | Estruturas de Dados - COM160 - Turma 009 |
| Teste | Semana 4 - Atividade Avaliativa |
| Iniciado | 31/08/23 19:34 |
| Enviado | 31/08/23 19:39 |
| Data de vencimento | 01/09/23 05:00 |
| Status | Completada |
| Resultado da tentativa | 8,57 em 10 pontos |
| Tempo decorrido | 5 minutos |
| Instruções | Atividade para avaliação Consulte os gabaritos dessa disciplina no menu lateral. Olá, estudante! 1. Para responder a esta atividade, selecione a(s) alternativa(s) que você considerar correta(s); 2. Após selecionar a resposta correta em todas as questões, vá até o fim da página e pressione "Enviar teste". 3. A cada tentativa, as perguntas e alternativas são embaralhadas Pronto! Sua atividade já está registrada no AVA. |
| Resultados exibidos | Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente |

Pergunta 11,43 em 1,43 pontos

**afirmações:**

I. A função hash(chave) deve ser determinística. Para uma determinada chave, a função sempre retorna o mesmo valor de hash.

II. Por ser utilizada como uma função de indexação, a função de hash deve sempre retornar um valor de hash dentro dos limites da tabela [0,N], em que N é o tamanho da tabela.

III. O método aproveita a possibilidade de acesso randômico à memória para alcançar uma complexidade média de $O(\log(n))$.

Assinalar a alternativa correta:

Resposta Selecionada: ☒ b. Apenas a I

Respostas:

☐ a. Apenas I e II.

☒ b. Apenas a I


☐ c. Apenas I e III

☐ d. Apenas a II.

☐ e. Todas estão corretas, I, II e III.

Comentário da resposta: **Apenas I**
JUSTIFICATIVA
Todas as afirmações são verdadeiras. A função hash(chave) deve ser determinística. Para uma determinada chave a função sempre retorna o mesmo valor de hash. Por ser utilizada como uma função de indexação, a função de hash deve sempre retornar um valor de hash dentro dos limites da tabela [0,N-1], onde N é o tamanho da tabela. O método aproveita a possibilidade de acesso randômico à memória para alcançar uma complexidade média por operação de $O(1)$, sendo o pior caso, entretanto, $O(n)$.

Pergunta 20 em 1,43 pontos

**Seja uma tabela hash implementada como um vetor de tamanho 13, com elementos indexados de 0 a 12. Nesse caso, para obter a posição a partir de uma chave, a função de espalhamento computa o resto da divisão da chave por 13. Indicar a alternativa correta que apresenta a chave (93,2,24) e o índice resultado da função de espalhamento aplicada a essa chave.**

Resposta Selecionada:

h(93)=1

h(2)=3

☒ a. h(24)=11

Respostas:

h(93)=1

h(2)=3

☐ a. h(24)=11

h(93)=2

h(2)=2

☐ b. h(24)=2

h(93)=1

h(2)=2

☐ c. h(24)=11

h(93)=2

h(2)=3

☐ d. h(24)=11

h(93)=2

h(2)=2

☒ e. h(24)=11

Comentário da resposta: **Justificativa**
Basta computar o resto na divisão por 13 para cada valor da chave.


Aplicando a função de espalhamento:

h(93)=2

h(2)=2

h(24)=11

Pergunta 31,42 em 1,42 pontos

**Um método de busca bastante utilizado, conhecido como hash, baseia-se na utilização que mapeia chaves em endereços de memória, de modo que os dados associados a cada chave possam ser rapidamente localizados e lidos. Quando há conflitos de localização, algum algoritmo de separação é adotado.**

Considere uma tabela hash armazenada em um arquivo no disco rígido. Supondo-se que a mesma possua uma função de *hash* razoavelmente protegida de conflitos, o número médio de acessos ao disco, necessários para localizar uma chave em um universo de N chaves, é mais próximo de:

Resposta Selecionada: ☒ d. 2

Respostas:

☐ a. $N / \log_2(N)$

☐ b. $N/2$


☐ c. $N*\log_2(N)$

☒ d. 2

☐ e. $\log_2(N)$

Comentário da resposta: **JUSTIFICATIVA**
A complexidade média de busca em uma tabela hash é $O(1)$.

Pergunta 41,42 em 1,42 pontos

**Uma tabela recebe chaves do tipo string e armazena os dados internamente como um vetor. A função de espalhamento da tabela Hash utiliza o seguinte procedimento para mapear as strings em inteiros:**

1 – Mapeamento de caracteres: os três primeiros caracteres são mapeados em inteiros da forma:

De a até f: mapeado para 1

De g até n: mapeado para 3

De o até s: mapeado para 5

De t até z: mapeado para 11

2 – Os inteiros associados a cada um dos três primeiros caracteres são multiplicados entre si.

3 – O resto da divisão por 11 é computado, dado que o vetor possui tamanho 11.

Dadas as seguintes strings: ULISSES, DANIELLE e LARISSA, aplicando a função de espalhamento apresentada, indique a alternativa correta que apresenta a string e a posição obtida.

Resposta Selecionada:

h(ulisses) = 0

h(danielle) = 3

☒ b. h(larissa) = 4

Respostas:

h(ulisses) = 0

h(danielle) = 0

☐ a. h(larissa) = 0

h(ulisses) = 0

h(danielle) = 3

☒ b. h(larissa) = 4

h(ulisses) = 0

h(danielle) = 0

☐ c. h(larissa) = 4

h(ulisses) = 8

h(danielle) = 0

☐ d. h(larissa) = 4

h(ulisses) = 8

h(danielle) = 3

☐ e. h(larissa) = 0

Comentário da resposta: **JUSTIFICATIVA**
A tabela foi preenchida com as strings acima gerando a configuração:


As seguintes *strings* exemplificam essa função de espalhamento.

h(ulisses) = $u*a*i=11*3*3 \bmod 11 = 0$

h(danielle) = $d*a*n=1*1*3 \bmod 11 = 3$

h(larissa) = $l*a*r=3*1*5=15 \bmod 11 =4$

Pergunta 51,42 em 1,42 pontos

**Seja f a função de espalhamento ou mapeamento e x a chave, o endereço de memória será atribuído por f(x). Os valores serão distribuídos em um vetor de N posições, sendo formado em um intervalo entre 0 e N-1.**

Sobre a utilização da função de mapeamento ou função *hash*, avalie se são (V) verdadeiras ou (F) falsas as afirmativas a seguir.

I. () Utilizada para guardar uma coleção de strings.

II. () Utilizada para obter os registros de maneira eficiente em tempo constante.

III. () Utilizada para acessar os arquivos no computador.

IV. () Utilizada para ter acesso a uma determinada aplicação.

Assinale a alternativa que apresenta a sequência CORRETA.

Resposta Selecionada: ☒ a. V - V - F - F

Respostas:

☒ a. V - V - F - F

☐ b. V - F - V - F

☐ c. F - V - F - V

☐ d. V - V - F - V

☐ e. F - V - F - F


Comentário da resposta: **JUSTIFICATIVA**
A afirmativa I é verdadeira, pois a principal utilização da função *hash* é armazenar uma coleção de dados (que podem ser strings), que será depois utilizada.

A afirmativa II é verdadeira, porque a principal utilização da função *hash* é, após ter armazenado os dados, ter acesso rápido $O(n)$ a eles toda vez que a aplicação necessitar.

A afirmativa III é falsa, visto que a principal utilização da função *hash* é acessar os dados armazenados, e não os arquivos, pois os dados são de diversos tipos de estruturas de dados, e não arquivos no computador.

A afirmativa IV é falsa, porque a aplicação será responsável pela função *hash*, para armazenar a informação e depois acessá-la, e não para dar acesso a uma determinada aplicação.

Pergunta 61,44 em 1,44 pontos

**Um procedimento natural para resolver os problemas de colisões consiste em guardar as chaves sinônimas em listas encadeadas. Existem duas opções: as listas podem se localizar no exterior da tabela ou compartilhar o mesmo espaço da tabela.**

O encadeamento exterior consiste em manter _____, uma para cada endereço-base possível. Os _____ correspondentes aos endereços-base serão apenas os principais dessas listas. Um campo para o encadeamento deve ser adicionado a cada nó. A _____ interna consiste nos nós que correspondem a cada endereço de encadeamento possível.

Preencha as lacunas escolhendo a alternativa CORRETA:

Resposta Selecionada: ☒ a. listas encadeadas; nós; cadeia

Respostas:

☒ a. listas encadeadas; nós; cadeia

☐ b. listas encadeadas; endereços; cadeia


☐ c. ponteiros; endereços; chave

☐ d. listas encadeadas; endereços; chave

☐ e. ponteiros; nós; chave

Comentário da resposta: **JUSTIFICATIVA**
O encadeamento exterior consiste em manter listas encadeadas, uma para cada endereço-base possível. Os nós correspondentes aos endereços-base serão apenas os principais dessas listas. Um campo para o encadeamento deve ser adicionado a cada nó. A cadeia interna consiste nos nós que correspondem a cada endereço de encadeamento possível.

Pergunta 71,44 em 1,44 pontos

**A tentativa linear $h(x,k)$ é uma implementação muito simples, em que o endereço-base x é $h'(x)$ ($k=0$), suponha que existe outra chave, x' , ocupando o mesmo endereço de $h'(x)$. A ideia da tentativa linear é buscar armazenar um novo nó, no endereço próximo, que existe em $h'(x) + 1$ ($k=1$), se por acaso já estiver ocupado, ele irá tentar em $h'(x) + 2$ ($k=2$), e assim sucessivamente.**

Com base nos aspectos que existem no método de tentativa linear, assinale a alternativa que descreve a função da tentativa linear para a $(k+1)$ -ésima tentativa.

Resposta Selecionada: ☒ c. $h(x, k) = (h'(x) + k) \bmod m, 0 \leq k \leq m - 1$ ↕

Respostas:

☐ a. $h(x, k+1) = (h(x, k - 1) + k) \bmod m, 0 < k \leq m$

☐ b. $h(x) = k \bmod m$

☒ c. $h(x, k) = (h'(x) + k) \bmod m, 0 \leq k \leq m - 1$ ↕

☐ d. $h(x, k) = (h'(x) + k+1) \bmod m$

☐ e. $h(x, k) = (h'(x) + k+1) \bmod m, 0 < k \leq m+1$

Comentário da resposta: **JUSTIFICATIVA**
A resposta correta é “ $h(x, k) = (h'(x) + k) \bmod m, 0 \leq k \leq m - 1$ ” pois se a posição $h'(x) + k$ estiver ocupada tenta-se na posição seguinte $h(x, k+1)$ (que é a $(k+2)$ -ésima tentativa).