Prova III - Teoria

Entrega 20 jun em 10:30 **Pontos** 10 **Perguntas** 32
Disponível 20 jun em 8:30 - 20 jun em 10:30 2 horas
Limite de tempo 120 Minutos

Instruções

Nossa 3a prova de AEDs II tem duas partes: teórica e prática. Cada uma vale 10 pontos. A prova teórica será realizada no Canvas e a prática, no Verde.

A prova teórica será realizada no dia 20 de junho, no horário da aula teórica. Essa prova tem 30 questões de verdadeiro ou falso e duas questões abertas de código. As questões de verdadeiro ou falso valem 0,1 e as abertas, 3.5. Abaixo, seguem as regras para a prova.
1) O código de acesso à prova será fornecido pelo professor no início da prova.
2) Após o envio de uma questão não é permitido que o aluno volte na mesma.
3) A prova é individual e é permitida a consulta à cola que contém o nome do aluno.
4) A interpretação faz parte da prova.
5) Se existir algum erro, após a divulgação do gabarito, peça a anulação da questão.
6) Os alunos da manhã 1/manhã farão a prova nos lab 9.
7) Os alunos da turma 2/manhã farão a prova nos lab 2.
8) Os alunos da tarde farão a prova no lab 10.

Desejamos uma excelente prova para todos.

Este teste foi travado 20 jun em 10:30.

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	Tentativa 1	82 minutos	5,2 de 10

Pontuação deste teste: **5,2** de 10
Enviado 20 jun em 10:17
Esta tentativa levou 82 minutos.

Pergunta 1 — 0,1 / 0,1 pts

No método de transformação (*hashing*), os registros armazenados em uma tabela são diretamente endereçados a partir de uma

transformação aritmética sobre a chave de pesquisa. Com relação às funções de transformação e colisões, podemos afirmar que uma função de transformação deve mapear as chaves em números inteiros, dentro de um intervalo da forma *[0, m - 1]*, em que *m* representa o valor da chave (TRE-PI'16, adaptado).

- Verdadeiro
- Falso

Correto!

A afirmação é falsa, pois *m* representa o tamanho da tabela.

Pergunta 2

0 / 0,1 pts

Considere uma tabela *hash* contendo um *array* de árvores binárias de pesquisa. O melhor caso acontece com custo $\Theta(1)$ e o pior, com o custo $\Theta(n)$.

esposta correta

- Verdadeiro

ocê respondeu

- Falso

A afirmação é verdadeira. O melhor caso acontece quando o elemento desejado está na raiz da árvore representando sua posição. O pior caso acontece quando todos os elementos inseridos possuem o mesmo valor na função de transformação e eles foram inseridos de tal forma que cada nível da árvore tem um elemento (e.g., inseridos em ordem crescente). Além disso, nesse caso, o elemento desejado encontra-se na folha da árvore ou a pesquisa pelo mesmo passa por essa folha.

Pergunta 3

0,1 / 0,1 pts

Considere a tabela *hash* [3 7 8 1 2 5 9] que tem quatro posições e mais três para área de reserva onde o 3 está na posição zero e o 2, 5 e 9 estão na área de reserva, para pesquisar o elemento 9 faremos apenas três comparações. Lembre-se que a área de reserva está implementada como uma lista sequencial não ordenada conforme apresentada na sala.

- Verdadeiro
- Falso

Correto!

A afirmação é falsa e a pesquisa proposta efetua quatro comparações. Uma nas quatro posições da tabela e mais três na área de reserva.

Pergunta 4

0 / 0,1 pts

Os métodos *hashing* envolvem um processo de transformação de uma chave em um endereço. Sobre esses métodos, podemos afirmar que quando duas ou mais chaves possuem o mesmo endereço primário ocorre uma colisão. Mesmo que se obtenha uma função *hash* que distribua as chaves de forma uniforme, existe grande chance de haver colisões. Além disso, deve haver uma forma de tratar as colisões. Uma das formas de se resolver as colisões é construindo uma lista encadeada para cada endereço da tabela. Assim, todas as chaves com mesmo endereço são encadeadas (TCE-RS'14, adaptada).

esposta correta

- Verdadeiro

ocê respondeu

- Falso

A afirmação é verdadeira. Ela aborda as colisões são o maior desafio no projeto das tabelas *hash*. A resolução adequada das colisões implica em reduzir o custo de pesquisa.

Pergunta 5

0,1 / 0,1 pts

Considere uma tabela *hash* com *rehash* contendo cinco posições e usando as funções de transformação apresentadas abaixo, podemos afirmar que após a inserção dos elementos 1, 6, 5, 10, 3 e 4, a tabela resultante será [5 1 6 3 4].

```
int hash(int key){
    return key%5;
}

int rehash(int key){
    return ++key%5;
}
```

Correto!

- (●) Verdadeiro
- ( ) Falso

A afirmação é verdadeira, pois o 10 não será inserido.

Pergunta 6

0,1 / 0,1 pts

A técnica de inserção com fragmentação na descida na 2.3.4 pode aumentar, reduzir ou manter o número de nós do tipo 4.

Falso

Correto!

Verdadeiro

A afirmação é verdadeira. Na inserção com fragmentação na descida, quando temos um nó do tipo 4 esse será fragmentado o que pode levar a uma redução do número de nós do tipo 4. Por outro lado, quando inserimos o elemento em uma folha do tipo 3, essa vira do tipo 4, aumentando o número de nós desse tipo. Temos ainda que durante a inserção podemos não ter fragmentações, mantendo o número de nós do tipo 4 existentes na árvore. Esse número também é mantido quando para cada Fragmentação, temos um nó do tipo 3 que vira do tipo 4.

Pergunta 7

0,1 / 0,1 pts

Altura de uma árvore 2.3.4 só aumenta quando existe uma fragmentação da raiz da árvore.

Falso

Correto!

Verdadeiro

A afirmação é verdadeira. Nas demais fragmentações, o elemento do meio "sobe" para seu pai. No caso da fragmentação da raiz, como não existe um pai, esse é criado.

Pergunta 8

0,1 / 0,1 pts

Nas árvores 2.3.4, uma das vantagens da técnica de inserção com fragmentação na descida sobre a na ascensão é que a primeira

normalmente consome menos espaço de memória.

Correto!

- (•) Falso
- ( ) Verdadeiro

A afirmação é falsa, pois a fragmentação na descida adianta algumas fragmentações, criando mais nós e, assim, consumindo mais memória.

Pergunta 9

0,1 / 0,1 pts

Dadas duas árvores 2.3.4 A e B obtidas a partir de uma árvore alvinegra, A e B possuem os mesmos números de nós e arestas.

Correto!

- (•) Verdadeiro
- ( ) Falso

A afirmação é verdadeira. O número de nós na alvinegra é igual ao número de elementos existentes na 2.3.4. O número de arestas da alvinegra é igual à soma entre o número de arestas da 2.3.4 e o de relacionamento entre gêmeos da 2.3.4.

Pergunta 10

0,1 / 0,1 pts

Na árvore 2.3.4, após uma inserção utilizando fragmentação na descida, podemos garantir a inexistência de nós do tipo 4 no caminho entre a raiz e a folha em que aconteceu a inserção.

Verdadeiro

Correto!

Falso

A afirmação é falsa. Na inserção com fragmentação na descida, quando chegamos em um nó do tipo 4, esse é fragmentado. Nessa fragmentação, se o pai era do tipo 3, ele ficará sendo do tipo 4. Da mesma forma, se inserirmos em uma folha do tipo 3, essa ficará sendo do tipo 4.

Pergunta 11

0,1 / 0,1 pts

Uma das estruturas de dados utilizadas na modelagem de sistemas de software denomina-se árvore rubro-negra. Nesse caso, um nó será vermelho quando, na árvore 2-3-4 correspondente, o valor desse nó for gêmeo do nó pai. Caso contrário, o nó é preto. Em uma árvore rubro-negra temos que se um nó é vermelho, seus filhos são vermelhos.

Verdadeiro

Correto!

Falso

A afirmação é falsa conforme os conceitos de árvores alvinegras apresentados na sala de aula.

Pergunta 12

0,1 / 0,1 pts

Uma das estruturas de dados utilizadas na modelagem de sistemas de software denomina-se árvore rubro-negra. Nesse caso, um nó será vermelho quando, na árvore 2-3-4 correspondente, o valor desse nó

for gêmeo do nó pai. Caso contrário, o nó é preto. Em uma árvore rubro-negra temos que a quantidade de nós vermelhos é sempre par.

Correto!

- (●) Falso
- ( ) Verdadeiro

A afirmação é falsa conforme os conceitos de árvores alvinegras apresentados na sala de aula.

Pergunta 13

0,1 / 0,1 pts

Uma das estruturas de dados utilizadas na modelagem de sistemas de software denomina-se árvore rubro-negra. Nesse caso, um nó será vermelho quando, na árvore 2-3-4 correspondente, o valor desse nó for gêmeo do nó pai. Caso contrário, o nó é preto. Em uma árvore rubro-negra temos que o nó raiz é preto.

- ( ) Falso

Correto!

- (●) Verdadeiro

A afirmação é verdadeira conforme os conceitos de árvores alvinegras apresentados na sala de aula.

Pergunta 14

0,1 / 0,1 pts

Uma das estruturas de dados utilizadas na modelagem de sistemas de software denomina-se árvore rubro-negra. Nesse caso, um nó será vermelho quando, na árvore 2-3-4 correspondente, o valor desse nó

for gêmeo do nó pai. Caso contrário, o nó é preto. Em uma árvore rubro-negra temos que se um nó é preto, seus filhos são pretos.

Correto!

- (•) Falso
- ( ) Verdadeiro

A afirmação é falsa conforme os conceitos de árvores alvinegras apresentados na sala de aula.

Pergunta 15

0,1 / 0,1 pts

Em uma árvore alvinegra, quando um nó tem 2 filhos pretos, fazemos a inversão das cores entre os filhos e o pai.

Correto!

- (•) Verdadeiro
- ( ) Falso

A afirmação é verdadeira conforme os conceitos de árvores alvinegras apresentados na sala de aula.

Pergunta 16

0,1 / 0,1 pts

A AVL é uma árvore de busca autobalanceada. Isso significa que cada nó da árvore possui até três descendentes (SEFAZ-PI'15, adaptado).

Correto!

- (•) Falso
- ( ) Verdadeiro

A afirmação é falsa conforme o conceito de árvore AVL.

Pergunta 17

0,1 / 0,1 pts

A AVL é uma árvore de busca autobalanceada. Isso significa que as alturas das duas sub-árvores a partir de cada nó são exatamente iguais (SEFAZ-PI'15, adaptado).

- Verdadeiro
- Falso

Correto!

A afirmação é falsa conforme o conceito de árvore AVL.

Pergunta 18

0,1 / 0,1 pts

Todos os n nomes de uma lista de assinantes de uma companhia telefônica foram inseridos, em ordem alfabética, em três estruturas de dados: uma árvore binária de busca, uma árvore AVL e uma árvore bicolor. As alturas resultantes das três árvores são $\Theta(n)$, $\Theta(lg(n))$ e $\Theta(lg(n))$, respectivamente (PETROBRAS'12, adaptada)

- Falso
- Verdadeiro

Correto!

A afirmação é verdadeira. A AB não é balanceada e as demais são.

Pergunta 19

0,1 / 0,1 pts

A AVL é uma árvore de busca autobalanceada. Isso significa que pode possuir até duas raízes (SEFAZ-PI'15, adaptado).

- Verdadeiro
- Falso

Correto!

A afirmação é falsa conforme o conceito de árvore AVL.

Pergunta 20

0,1 / 0,1 pts

Considere a assinatura das duas funções abaixo na linguagem C++:

```
void troca(double &x, double &y)
void troca(double* x, double* y)
```

Podemos afirmar que a primeira função utiliza passagem de parâmetros por referência e a segunda, por valor.

- Falso
- Verdadeiro

Correto!

A afirmação é verdadeira considerando os conceitos sobre passagem de parâmetros na linguagem C++ apresentados na sala de aula.

Pergunta 21

0,1 / 0,1 pts

Avaliando os códigos abaixo nas linguagens C e C++, é correto afirmar que os dois imprimem os mesmos valores para as variáveis "x" e "y" no final de suas execuções. Isso acontece mesmo com a passagem de parâmetros do primeiro código sendo por valor e a do segundo, sendo por referência.

Código em C:

```
void troca(double *x, double *y){
        double aux = *x;
        *x = *y;
        *y = aux;
}

int main(){
        double x = 7.0;
        double y = 5.0;
        troca(&x, &y);
        printf("X: %lf Y: %lf", x, y);
}
```

Código em C++:

```
void troca(double& x, double& y){
       double aux = x;
       x = y;
       y = x;
}

int main(){
        double x = 7.0;
        double y = 5.0;
        troca(x, y);
        cout << "X: " << x;
```

```
        cout << "Y: " << y;
}
```

- Falso

Correto!

- Verdadeiro

A afirmação é verdadeira conforme discutido em sala de aula.

Pergunta 22

0,1 / 0,1 pts

Podemos afirmar que o código abaixo em C++ imprime "1 1" como saída

```
void f (int val, int& ref) {
   val ++; ref++;
}

void main () {
   int i = 1, j = 1;
   f(i, j);
  printf("%i -- %i", i, j);
}
```

Correto!

- Falso

- Verdadeiro

A afirmação é falta. A variável ref foi passada por referência, assim, seu valor final é dois.

Pergunta 23

0 / 0,1 pts

Considere que a Manausprev armazena os nomes dos beneficiários de aposentadorias em uma Árvore Binária de Busca (ABB). Ao se armazenar, nesta ordem, os nomes Marcos, José, Carolina, Paula, Rui, Pedro e Maria, a ABB resultante é uma árvore completa (MANAUSPREV'15, adaptada).

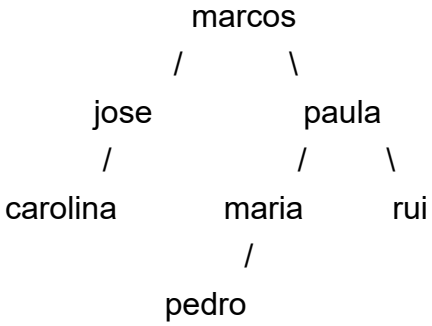
ocê respondeu

- Verdadeiro

esposta correta

- Falso

A afirmação é falsa e a árvore resultante é mostrada abaixo.



Pergunta 24

0,1 / 0,1 pts

As rotações são utilizadas no balanceamento das árvores de tal forma que uma árvore desbalanceada para a esquerda será rotacionada para à direita. Uma desbalanceada para a direita será para à esquerda. Nós temos quatro tipos de rotações: simples para esquerda, dupla esquerda-direita, simples para à direita e a dupla direita-esquerda. As duas primeiras são efetuadas em árvores desbalanceadas para à esquerda e as outras duas, desbalanceadas para à direita.

- Verdadeiro

Correto!

- Falso

A afirmação é falsa. As rotações simples para esquerda e a dupla direita-esquerda são efetuadas em árvores desbalanceadas para à esquerda e as outras duas, desbalanceadas para à direita.

Pergunta 25

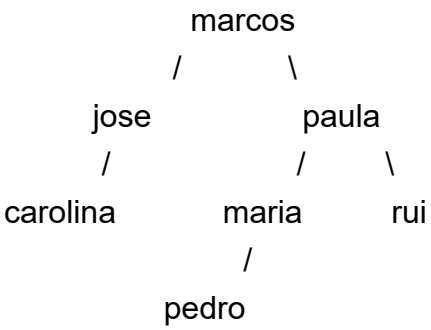
0,1 / 0,1 pts

Considere que a Manausprev armazena os nomes dos beneficiários de aposentadorias em uma Árvore Binária de Busca (ABB). Ao se armazenar, nesta ordem, os nomes Marcos, José, Carolina, Paula, Rui, Pedro e Maria, a ABB resultante requer no máximo 4 comparações para localizar qualquer um dos 7 nomes (MANAUSPREV'15, adaptada).

Correto!

- (●) Verdadeiro
- ( ) Falso

A afirmação é verdadeira e a árvore resultante é mostrada abaixo.



Pergunta 26

0,1 / 0,1 pts

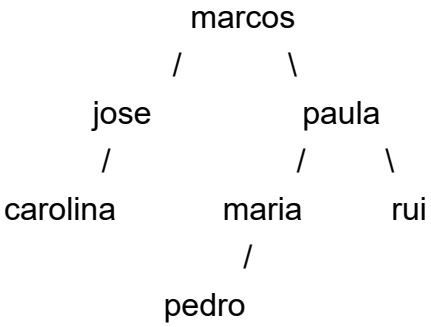
Considere que a Manausprev armazena os nomes dos beneficiários de aposentadorias em uma Árvore Binária de Busca (ABB). Ao se

armazenar, nesta ordem, os nomes Marcos, José, Carolina, Paula, Rui, Pedro e Maria, a ABB resultante tem os nomes Rui e Maria localizados em nós do tipo folha (MANAUSPREV'15, adaptada).

Correto!

- (◉) Falso
- ( ) Verdadeiro

A afirmação é falsa e a árvore resultante é mostrada abaixo.



Pergunta 27

0,1 / 0,1 pts

Nas linguagem Java e C++, ao instanciarmos um objeto usando new, fazemos com que tal objeto seja acessível a partir de um ponteiro/referência. Entretanto, diferente do Java, a linguagem C++ também permite a criação literal de objetos.

Correto!

- (◉) Verdadeiro
- ( ) Falso

Nas linguagem Java e C++, o comando new cria um objeto e retorna seu endereço de memória que deve ser armazenado em um ponteiro/referência. A linguagem Java não permite a criação literal de um objeto o que é possível em C++.

Pergunta 28

0,1 / 0,1 pts

A linguagem C++ permite criar funções cuja implementação do código acontece “fora” da classe. Nesse caso, “dentro” da mesma temos a assinatura da função e,”fora”, implementamos a função usando o nome da classe e o operador “::”. O exemplo abaixo mostra uma classe Cubo com sua função getVolume implementada de forma externa.

```
class Cubo {
   public:
      int lado;
      int getVolume();
}

int Cubo :: getVolume() {
   return lado*lado*lado;
}
```

- Falso

Correto!

- Verdadeiro

A afirmação é verdadeira e a implementação externa acontece conforme o exemplo apresentado.

Pergunta 29

0,1 / 0,1 pts

O código C++ abaixo tem um objeto chamado "quadrado_teste" do tipo Quadrado que pode ser usado globalmente.

```
class Quadrado {
   private:
      int lado;
```

```
    public:
        void set_values (int);

    //...
} ;

int main() {
    Quadrado quadrado_teste;

    //...
    return 0;
}
```

- Verdadeiro
- Falso

Correto!

A declaração da classe e a criação do objeto foram feitos de maneira adequada. Entretanto, como o objeto foi declarado dentro da função main, o escopo de vida do mesmo é aquela função a partir do seu ponto de declaração.

Pergunta 30

0,1 / 0,1 pts

Nas linguagens C/C++, acessamos um atributo de um registro/objeto apontado por um ponteiro fazendo ponteiro->atributo.

- Falso
- Verdadeiro

Correto!

O símbolo "->" permite acessarmos os atributos/funções de um registro/objeto apontados por um ponteiro.

Pergunta 31

0,5 / 3,5 pts

Seja a árvore Trie apresentada na sala de aula implemente um método para PESQUISAR palavras na árvore. Considere que a classe Nó da árvore Trie contém uma LISTA FLEXÍVEL para armazenar o endereço dos nós filhos. Apresente a ordem de complexidade do seu método. Sua resposta deve conter somente o método solicitado e sua complexidade.

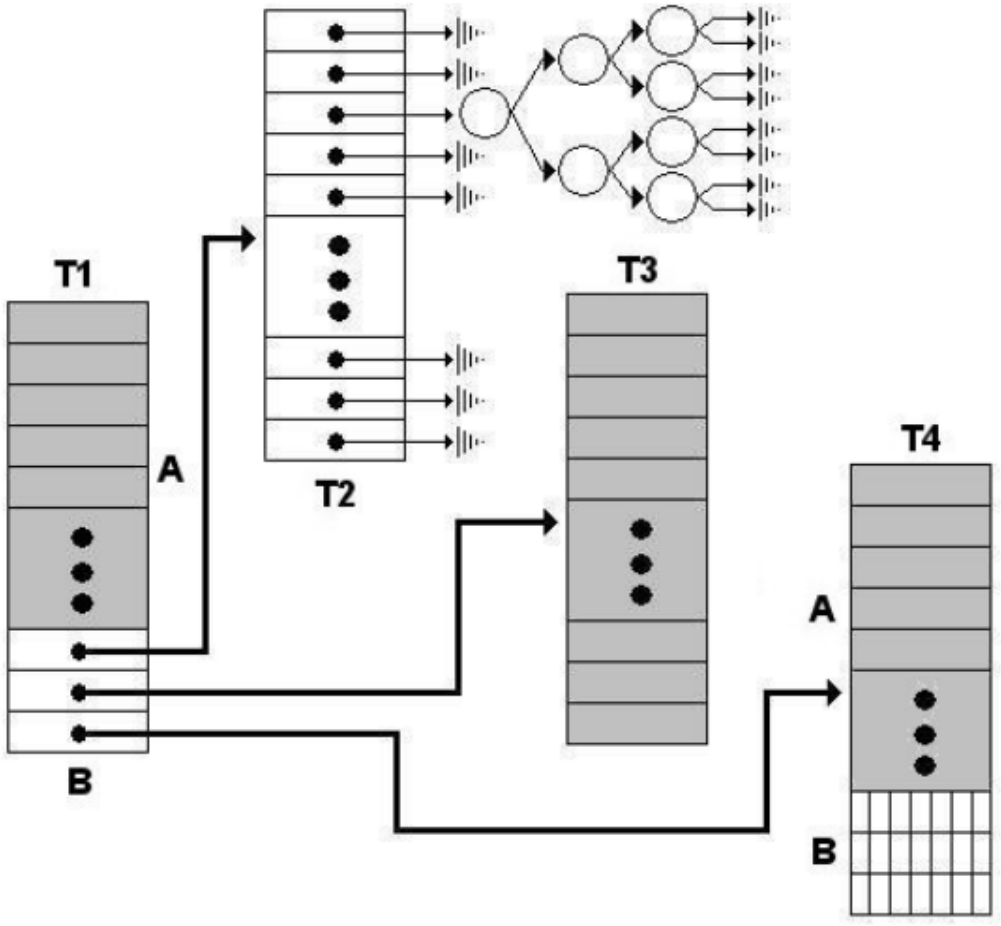
questao1.txt
(https://pucminas.instructure.com/files/6327171/download)

Ver gabarito.

Pergunta 32

2 / 3,5 pts

Suponha a estrutura de dados abaixo em que T1 é uma hash com área de reserva (cor branca) "virtual" que nos direciona para T2, T3 e T4. T2 é uma tabela de NoAVL. T3 é uma tabela com duas funções de rehash, int rehashT3A() e int rehashT3B(). T4 é uma tabela com área de reserva. Além disso, o tamanho das quatro tabelas é TAM e que T4 possui mais TAMRESERVA posições para sua área de reserva. Faça um método que recebe um elemento e o insere em nossa estrutura. Considere que todos os métodos hash estão implementados. Além disso, faça (e justifique) uma análise de complexidade do seu método.



doidona.java
(https://pucminas.instructure.com/files/6327377/download)

1) if(t2[i] == null) 2) T2 tem várias árvores 3) Complexidade?

Pontuação do teste: **5,2** de 10