Prova III - Teoria

Entrega 15 de dez de 2021 em 10:40 **Pontos** 10 **Perguntas** 32
Disponível 15 de dez de 2021 em 8:40 - 15 de dez de 2021 em 10:40
aproximadamente 2 horas
Limite de tempo 120 Minutos

Instruções

Esta prova TEÓRICA vale 10 pontos. Ela tem 30 questões de verdadeiro ou falso e duas questões abertas de código. As questões de verdadeiro ou falso valem 0,2 e as abertas, 2.5. O somatório dessas notas é igual a 10. Fizemos isso porque, novamente, NÃO será permitido retornar nas questões: nem fechadas, nem abertas. A submissão das questões abertas será efetuada em campo de arquivo texto.

A prova teórica da manhã acontecerá de 8:50 às 10:30 e a da tarde, de 13:30 às 15:10. A prova será aberta 10 minutos mais cedo e fechará 10 minutos mais tarde.

Este teste não está mais disponível, pois o curso foi concluído.

Histórico de tentativas

	Tentativa	Tempo	Pontuação
MAIS RECENTE	Tentativa 1	110 minutos	6,6 de 10

As respostas corretas não estão mais disponíveis.

Pontuação deste teste: **6,6** de 10
Enviado 15 de dez de 2021 em 10:32
Esta tentativa levou 110 minutos.

Pergunta 1 — 0,2 / 0,2 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

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 2

0,2 / 0,2 pts

Considere uma tabela *hash* com *rehash* contendo duas posições e usando as funções de transformação apresentadas abaixo, podemos afirmar que após a inserção dos elementos 1, 4, 5, 2, e 7, a tabela resultante será [4 1].

```
int hash(int key){
    return key%2;
}

int rehash(int key){
    return ++key%2;
}
```

- Verdadeiro
- Falso

A afirmação é verdadeira, pois o 5, 2 e 7 não serão inseridos.

Pergunta 3

0,2 / 0,2 pts

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

- Verdadeiro
- Falso

A afirmação é falsa. 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, nesse caso, o elemento desejado encontra-se em uma das folhas da árvore ou o mesmo não está contido na árvore.

Pergunta 4

0,2 / 0,2 pts

Os métodos *hashing* envolvem um processo de transformação de uma chave em um endereço. Sobre esses métodos, podemos afirmar que o tempo gasto com pesquisas depende do tamanho da tabela e da quantidade de elementos inseridos. Além disso, a função *hash* de transformação deve envolver uma operação simples sobre a chave (TCE-RS'14, adaptada).

- Falso
- Verdadeiro

A afirmação é verdadeira. Primeiro, a dependência do tamanho da tabela e da quantidade de elementos acontece porque tais fatores influenciam nas colisões e o quão maior o número de colisões, maior o custo para encontrar elementos. Segundo, é importante que a operação de transformação seja simples, evitando o *overhead* de processamento.

Incorreta

Pergunta 5

0 / 0,2 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 devido ao fato das transformações nas chaves serem aritméticas, uma função *hashing* aceita como chave apenas um valor numérico. Assim, não é possível passar uma chave não numérica, pois não é possível fazer transformação da chave (TRE-PI'16, adaptado).

- (●) Verdadeiro
- ( ) Falso

A afirmação é falsa. Nas aulas vimos, por exemplo, o uso de strings e datas como chaves de pesquisa.

Pergunta 6

0,2 / 0,2 pts

A árvore 2.3.4 é uma estrutura de pesquisa cujos nós são de três tipos (2-nó, 3-nó ou 4-nó) e as folhas estão situadas no mesmo nível.

Falso

Verdadeiro

A afirmação é verdadeira dado que os três tipos de nós foram citados e nesta árvore sempre inserimos novos valores nas folhas.

Pergunta 7

0,2 / 0,2 pts

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

Falso

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.

Incorreta

Pergunta 8

0 / 0,2 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.

Verdadeiro

Falso

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

Incorreta

Pergunta 9

0 / 0,2 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.

- ○ Falso
- ◉ Verdadeiro

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 10

0,2 / 0,2 pts

A inserção dos números 2, 12, 5, 8, 1, 11, 9, 7, 4, 13, 10, 6 e 3 em uma 2.3.4 usando as técnicas de fragmentação por ascenção e na descida gera a mesma árvore resultante.

- ○ Falso
- ◉ Verdadeiro

A afirmação é verdadeira. A inserção usando as duas técnicas é igual até a inserção do 10. Elas ficam diferentes após a inserção do 6 e voltam a ficar iguais com a inserção do 3.

Pergunta 11

0,2 / 0,2 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.

- ◉ Verdadeiro
- ○ Falso

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

Incorreta

Pergunta 12

0 / 0,2 pts

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

- ○ Verdadeiro
- ◉ Falso

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

Incorreta

Pergunta 13

0 / 0,2 pts

Considerando que em uma Árvore Alvinegra, um nó é do tipo quatro quando seus dois filhos possuem cor preta (true). O método abaixo recebe o endereço de um nó e retorna true quando esse nó juntamente com seus dois filhos.

```
boolean isTipoQuatro(No i){
   return (i.esq.cor == true && i.dir.cor == true);
}
```

Podemos afirmar que o método acima está correto.

- ◉ Verdadeiro
- ○ Falso

A afirmação é falsa. O método proposta apresenta erro de compilação quando ele for avaliar um nó contendo zero ou um filho.

Pergunta 14

0,2 / 0,2 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.

- ○ Verdadeiro
- ◉ Falso

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

Pergunta 15

0,2 / 0,2 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.

- ◉ Falso
- ○ Verdadeiro

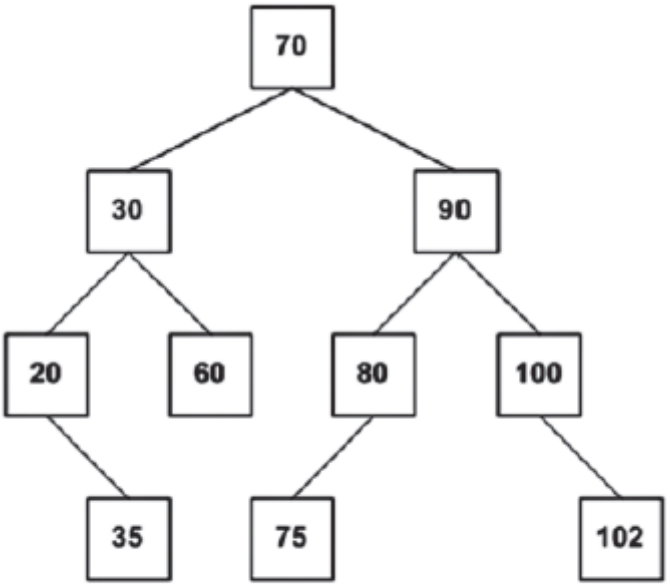
A afirmação é falsa conforme os conceitos de árvores alvinegras apresentados na sala de aula.

Incorreta

Pergunta 16

0 / 0,2 pts

A estrutura de dados AVL é uma árvore binária de busca balanceada criada pelos soviéticos Adelson, Velsky e Landis em 1962. Podemos afirmar que a figura abaixo representa uma árvore AVL (PETROBRAS'12, adaptada).



- (●) Verdadeiro
- ( ) Falso

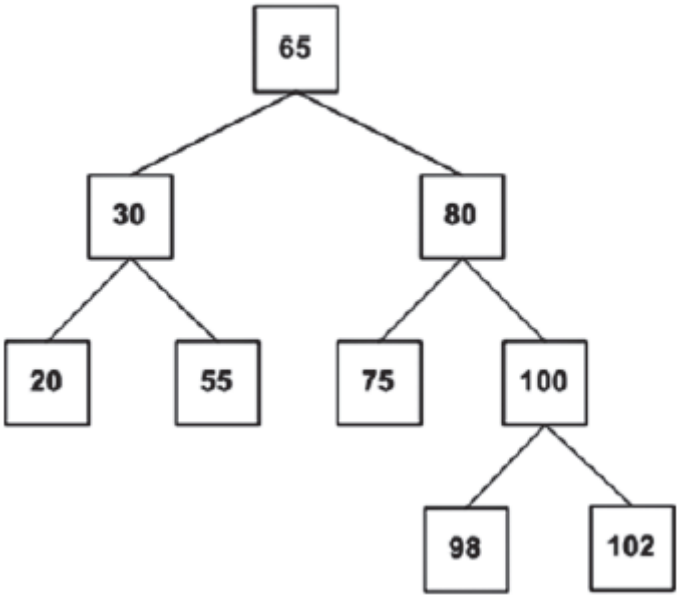
A afirmação é falsa conforme o conceito de árvore AVL.

Pergunta 17

0,2 / 0,2 pts

A estrutura de dados AVL é uma árvore binária de busca balanceada criada pelos soviéticos Adelson, Velsky e Landis em 1962. Podemos

afirmar que a figura abaixo representa uma árvore AVL (PETROBRAS'12, adaptada).



- Falso
- Verdadeiro

A afirmação é verdadeira conforme o conceito de árvore AVL.

Pergunta 18

0,2 / 0,2 pts

A AVL é uma árvore de busca autobalanceada. Isso significa que as alturas das duas sub-árvores a partir de cada nó diferem no máximo em duas unidades (SEFAZ-PI'15, adaptado).

- Verdadeiro
- Falso

A afirmação é falsa conforme o conceito de árvore AVL.

Pergunta 19

0,2 / 0,2 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).

- (●) Falso
- ( ) Verdadeiro

A afirmação é falsa conforme o conceito de árvore AVL.

Pergunta 20

0,2 / 0,2 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);
}
```

- ( ) Verdadeiro

Falso

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

Pergunta 21

0,2 / 0,2 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
- Verdadeiro

A afirmação é verdadeira conforme discutido em sala de aula.

Pergunta 22

0,2 / 0,2 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.

- Verdadeiro
- Falso

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

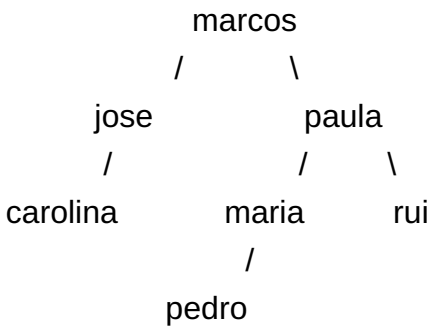
Pergunta 23

0,2 / 0,2 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).

- (●) Verdadeiro
- ( ) Falso

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



Pergunta 24

0,2 / 0,2 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.

- (●) Falso

Verdadeiro

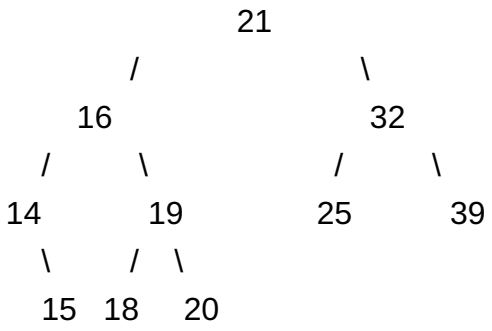
A afirmação é falsa. As rotações simples para esquerda e a dupla direita-esquerda são efetuadas em árvores desbalanceadas para à esquerda e as outras duas, desbalanceadas para à direita.

Incorreta

Pergunta 25

0 / 0,2 pts

Dada a árvore binária abaixo e o código do método pesquisar, podemos afirmar que a chamada do método pesquisar(18) imprime na tela duas vezes a letra "A".




```
public boolean pesquisar(int x) {
    return pesquisar(x,raiz);
}

private boolean pesquisar(int x, No i) {
   boolean encontrado;
   if(i == null) {
      encontrado = false;
   } else if(x == i.elemento) {
      encontrado = true;
   } else if(x < i.elemento ) {
      encontrado = pesquisar(x, i.esq);
      System.out.println("A");
   } else {
      encontrado = pesquisar(x, i.dir);
   }
```

```
    return encontrado;
}
```

Verdadeiro

Falso

A afirmação é verdadeira. A impressão acontece da letra acontece quando o algoritmo caminha para a esquerda a partir dos nós contendo os números 21 e 19.

Pergunta 26

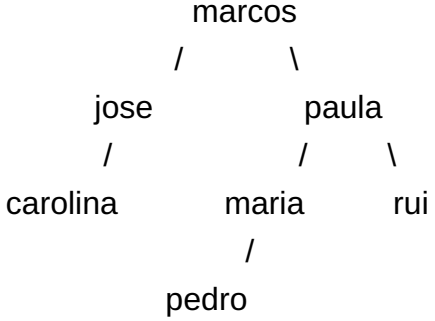
0,2 / 0,2 pts

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).

Verdadeiro

Falso

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



Pergunta 27

0,2 / 0,2 pts

Nas linguagens Java e C++, os atributos de uma classe podem ter as visibilidades public, private ou protected. Um atributo público é acessível dentro e fora da classe. Um privado, somente dentro da classe. Um protegido, somente dentro da classe ou de classes filhas.

- (●) Verdadeiro
- ( ) Falso

As linguagens Java e C++ são orientadas por objetos e possuem os três tipos de visibilidade apresentados com suas respectivas definições.

Pergunta 28

0,2 / 0,2 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

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 29

0,2 / 0,2 pts

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

Verdadeiro

Falso

A afirmação está correta para a linguagem Java, contudo, nas linguagens C/C++, devemos usar o símbolo "->".

Pergunta 30

0,2 / 0,2 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
- Verdadeiro

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

Pergunta 31

2 / 2 pts

Seja a árvore Trie apresentada na sala de aula modifique o método INSERIR supondo que nossa árvore permite a existência de prefixos. Por exemplo, assim, podemos ter as palavras AMO, AMOR, AMORA, AMORES. Apresente a ordem de complexidade do seu método. Sua resposta deve conter somente o método solicitado e sua complexidade.

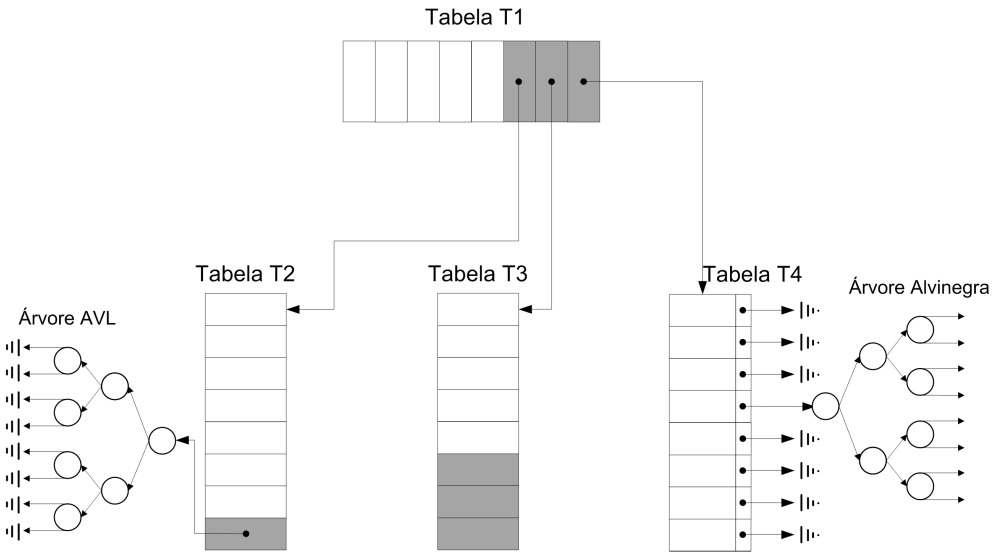
prova3q31.txt (https://pucminas.instructure.com/files/5196252/download)

o respondida

Pergunta 32

0 / 2 pts

Crie a estrutura de dados Doidona (mostrada na figura abaixo) para armazenar números inteiros. Nesta questão, você deve implementar (em Java) a classe Doidona com seus atributos, um construtor e o método public void inserir(int elemento). A estrutura doidona tem quatro tabelas. T1 é uma hash com área de reserva. T2 é uma hash com duas funções de rehash e uma área de reserva que corresponde a uma árvore AVL. T3 é uma hash com área de reserva de tamanho três. T4 é uma hash indireta com árvores alvinegras. Suponha que todos os métodos de hash e rehash estão implementados (int hashT1(int elemento), int hashT2(int elemento), int hashT3(int elemento) e int hashT4(int elemento), int rehash1T2(int elemento), int rehash2T2(int elemento), respectivamente.



Pontuação do teste: **6,6** de 10