

ESTRUTURAS DE DADOS HETEROGÊNEAS

1.

Em um projeto de software para gerenciamento de bibliotecas, utiliza-se a estrutura de dados heterogênea para modelar livros. Cada livro é representado por uma estrutura contendo título, autor, ano de publicação e status de empréstimo.

Qual estrutura de dados em C é mais apropriada para armazenar os detalhes de cada livro no projeto descrito?

- ☒ Estrutura (struct).
- ☐ Ponteiro para void.
- ☐ Lista ligada de caracteres.
- ☐ Array de inteiros.
- ☐ Matriz bidimensional de floats.

Data Resp.: 15/02/2024 23:35:19

Explicação:

A alternativa correta é a estrutura (struct) porque permite agrupar diferentes tipos de dados (como strings para título e autor, int para ano de publicação, e um booleano para o status de empréstimo), o que é ideal para representar livros em um sistema de gerenciamento de biblioteca.

2.

Para melhorar a performance de um jogo, os desenvolvedores precisam armazenar e manipular eficientemente as posições dos personagens em um mundo virtual tridimensional.

Qual tipo de dado em C é mais adequado para representar as posições tridimensionais dos personagens?

- ☐ Lista ligada de structs.
- ☒ Estrutura contendo três floats.
- ☐ Array de caracteres.
- ☐ Variável do tipo double.
- ☐ Ponteiro para inteiro.

Data Resp.: 15/02/2024 23:35:21

Explicação:

Uma estrutura contendo três floats é ideal para representar posições tridimensionais, pois permite armazenar as coordenadas x, y, e z de forma organizada e acessível, facilitando o cálculo de movimentos e a renderização gráfica no jogo.

3.

Em um sistema de gerenciamento de estoque, utiliza-se uma estrutura para armazenar informações dos produtos, como código, descrição, quantidade e

preço. É necessário acessar rapidamente essas informações para atualizações frequentes.

Como se deve acessar os membros de uma estrutura em C para realizar atualizações?

- ☐ Chamada de função.
- ☒ Indexação de array.
- ☐ Operador ponto (.).
- ☐ Operador de desreferência.
- ☐ Operador seta (->).

Data Resp.: 15/02/2024 23:35:24

Explicação:

O operador ponto (.) é usado para acessar membros de uma estrutura quando se trabalha com uma instância da estrutura. Este método é eficaz para atualizações rápidas em sistemas de gerenciamento de estoque, permitindo a manipulação direta dos dados dos produtos.

4.

(IADES/2018) A sigla FIFO refere-se a estruturas de dados do tipo fila. Como é o funcionamento em uma FIFO?

- ☐ Uma FIFO e uma LIFO possuem as mesmas características de entrada e de saída dos objetos.
- ☐ O programador irá definir a ordem de entrada e de saída dos objetos em uma FIFO.
- ☐ O primeiro objeto inserido na fila é o último a ser removido.
- ☒ O primeiro objeto inserido na fila é também o primeiro a ser removido.
- ☐ O último objeto inserido na fila é o primeiro a ser removido.

Data Resp.: 15/02/2024 23:35:25

Explicação:

Em uma estrutura de dados do tipo FIFO (First In, First Out), o primeiro elemento a entrar na fila é o primeiro a sair, ou seja, o elemento que foi inserido primeiro será o primeiro a ser removido da estrutura. Isso ocorre porque a fila segue uma ordem linear de entrada e saída de elementos, sem possibilidade de inserção ou remoção em posições intermediárias.

5.

(UFAM/2022) Considere o programa a seguir, elaborado na linguagem ANSI C:

```
#include <stdio.h >
```

```
int* pa, a;
```

```
int main() { a = 10; pa = &a; a = 8; printf("%d", a); printf(" %d", *pa); }
```

Assinale a alternativa correta sobre a saída (output) obtida, ao se executar o programa:

- ☒ 8 10
- ☐ 8 8
- ☐ 9 10
- ☐ 10 10
- ☐ 10 8

Data Resp.: 15/02/2024 23:35:28

Explicação:

Vamos entender o que acontece em cada linha do código:

Declara-se duas variáveis: pa (um ponteiro para inteiro) e a (um inteiro).

Atribui-se o valor 10 à variável a.

Atribui-se o endereço da variável a ao ponteiro pa.

Atribui-se o valor 8 à variável a. Nesse momento, a variável pa também aponta para o valor 8.

Imprime-se o valor de a na tela. O valor de a é 8.

Imprime-se o valor apontado por pa na tela. O valor apontado por pa é 8.

Portanto, a saída do programa é "8 8".

6.

Sobre estruturas de dados, assinale a alternativa CORRETA.

- ☐ Filas são comumente implementadas sobre arrays ou grafos.
- ☒ Listas duplamente ligadas são estruturas em que cada nó possui uma referência tanto ao nó que o antecede quanto ao nó que o sucede. Além disso, o último nó da lista também possui uma referência para o primeiro nó da lista.
- ☐ Pilhas são tipos de dados abstratos caracterizadas pela política "primeiro a entrar, último a sair".
- ☐ Grafos são estruturas de dados em que cada nó possui um valor e um conjunto de relações unidirecionais com os demais nós.
- ☐ Árvores de busca de binárias são estruturas nas quais nós filhos possuem valores numericamente inferiores aos dos nós pais.

Data Resp.: 15/02/2024 23:35:34

Explicação:

A resposta correta é: Pilhas são tipos de dados abstratos caracterizadas pela política "primeiro a entrar, último a sair".

MODULARIZAÇÃO

1.

Observe o trecho de código abaixo, escrito na linguagem C.

```
void imprimecabecalho() {  
  
    ...  
  
}  
  
void calcula() {  
  
    int soma;  
  
    ...  
  
    imprimecabecalho();  
  
}
```

Com base nesse código, é correto afirmar que:

- ☐ O escopo e o tempo de vida da variável soma são iguais e contidos pela função `imprimecabecalho()`.
- ☐ O escopo da variável soma é contido pela função `imprimecabecalho()`.
- ☒ O escopo da variável soma é dinâmico e se estende durante toda execução do programa.
- ☐ O tempo de vida da variável soma estende-se durante o tempo em que a função `imprimecabecalho()` é executada.
- ☐ O escopo da variável soma se estende da função `calcula()` para a função `imprimecabecalho()`.

Data Resp.: 15/02/2024 23:38:12

Explicação:

Resposta correta: o tempo de vida da variável soma estende-se durante o tempo em que a função `imprimecabecalho()` é executada.

2.

(NUCEPE/2015 - adaptada) A modularização é importante para organizar melhor o código, facilitar a manutenção, entre outras coisas.

Sobre a modularização, assinale a alternativa correta.

- ☐ As variáveis locais são declaradas no escopo do programa inteiro.
- ☒ A passagem de parâmetros para um subprograma pode ser somente por valor.

- ☐ As variáveis definidas no escopo de cada função são acessíveis em todo o programa.
- ☐ É um procedimento que sempre retorna um valor ao programa.
- ☐ É um método que consiste em decompor um programa em uma série de subprogramas individuais.

Data Resp.: 15/02/2024 23:38:20

Explicação:

A modularização consiste em decompor um programa em uma série de subprogramas individuais. Trata-se de um método utilizado para facilitar a construção de grandes programas, através de sua divisão em pequenas etapas.

3.

Essas variáveis que são declaradas fora da sub-rotina, podendo ser acessíveis em todos os escopos, em qualquer lugar de um programa, disponíveis durante toda a execução do programa. Marque a alternativa que apresenta o tipo de variável descrito acima.

- ☐ Global.
- ☒ Formais.
- ☐ Local.
- ☐ Paramétricas.
- ☐ Única.

Data Resp.: 15/02/2024 23:38:22

Explicação:

As variáveis globais são declaradas fora da sub-rotina. São acessíveis em todos os escopos, em qualquer ponto de um programa, mesmo em outros módulos. Podem, portanto, ser usadas ou modificadas por qualquer sub-rotina do programa onde estão declaradas. Em outras palavras, as variáveis globais estão disponíveis durante toda a execução do programa.

4.

As sub-rotinas são procedimentos e funções da programação modularizada. Um parâmetro passado por valor para uma sub-rotina se comportará como uma variável local, isto é, qualquer modificação no valor desta variável não será visível fora da sub-rotina. Sobre a passagem de parâmetros, analise as afirmativas abaixo.

I - Na passagem por valor ocorre uma troca de valores entre variáveis na sub-rotina.

II - Na passagem por valor ocorre uma cópia do valor da variável em todas as posições de memória, sendo possível alterar o conteúdo da variável original a qualquer momento.

III - A passagem por referência nunca deve ser utilizada.

Marque a alternativa que apresenta a(s) afirmativa(s) correta(s).

- ☐ Apenas as afirmativas I e III estão corretas.
- ☒ As afirmativas I, II e III estão corretas.

- ☐ Apenas as afirmativas II e III estão corretas.
- ☐ **Apenas a afirmativa I está correta.**
- ☐ Apenas as afirmativas I e II estão corretas.

Data Resp.: 15/02/2024 23:38:26

Explicação:

A afirmativa II está incorreta. Na passagem por valor, é feita uma cópia do valor da variável, mas essa cópia não é armazenada em todas as posições de memória. Alterações feitas na cópia da variável dentro da sub-rotina não afetarão o conteúdo da variável original fora da sub-rotina.

A afirmativa III está incorreta. A passagem por referência pode ser utilizada em determinadas situações, quando se deseja que uma sub-rotina seja capaz de modificar diretamente o conteúdo da variável original. Ela envolve a passagem do endereço onde se encontra a variável usada como argumento na chamada da sub-rotina.

5.

(FUNCAB/2015 - adaptada) Na linguagem C, o cabeçalho de biblioteca `stdio.h` possui definições de subrotinas relativas às operações de entrada/saída padrão. Uma das funções tratadas na biblioteca, associada a esse cabeçalho, é a função.

- ☒ `Locatime.`
- ☐ **`Printf.`**
- ☐ `Stderr.`
- ☐ `Trunc.`
- ☐ `Strcpy.`

Data Resp.: 15/02/2024 23:38:29

Explicação:

As funções de E/S padrão são responsáveis pelas operações de entrada e saída de dados. Quando um programa na linguagem C é executado, o sistema operacional é responsável por abrir três arquivos: O arquivo de entrada padrão (`stdin`), o arquivo de saída padrão (`stdout`) e o erro padrão (`stderr`). Invocando funções de E/S padrão `printf`, `scanf`, `getchar`, `putchar`.

6.

Há duas maneiras de se passar argumentos ou parâmetros para funções: por valor e por referência. Sobre passagem de parâmetros, analise as seguintes afirmativas:

- I. Na passagem por referência, o que é passado como argumento no parâmetro formal é o endereço da variável.
- II. Na passagem por valor, o valor é copiado do argumento para o parâmetro formal da função.
- III. Por exemplo, quando duas variáveis inteiras `i1` e `i2` são passadas por valor à função `troca()` chamada pelo programa principal, elas também são alteradas no programa principal.
- IV. Na passagem por referência, dentro da função, o argumento real utilizado na chamada é acessado através do seu endereço, sendo assim alterado.

V. Na passagem por valor, quaisquer alterações feitas nestes parâmetros dentro da função não irão afetar as variáveis usadas como argumentos para chamá-la.

Está CORRETO o que se afirma em:

- ☒ V, apenas
- ☐ II e IV, apenas
- ☐ I, III e V, apenas
- ☐ I, II, IV e V, apenas
- ☐ I e III

Data Resp.: 15/02/2024 23:38:32

Explicação:

Resposta correta: I, II, IV e V, apenas



Não Respondida



Não Gravada



Gravada

Exercício por Temas iniciado em 15/02/2024 23:37:53.

LISTAS, PILHAS E FILAS

1.

O acesso ao elemento de uma estrutura de dados tipo pilha se restringe ao mais recente na pilha. Já o acesso a um elemento de uma estrutura tipo fila ocorre ao dado há mais tempo na fila. Sobre pilhas e filas, avalie as assertivas a seguir:

I - Uma forma de evitar o desperdício de memória numa fila em alocação sequencial é utilizar-se lista circular.

II - Em uma pilha em alocação encadeada, a complexidade da remoção é $O(n)$.

III - Pilhas têm a propriedade de inverter a ordem de cadeias, enquanto as filas mantêm a ordem.

A opção que contém todas as assertivas corretas é:

- ☐ I e III.
- ☐ II e III.
- ☐ I.
- ☒ I e II.

☐ II.

Data Resp.: 15/02/2024 23:38:39

Explicação:

A resposta correta é: I e III.

2.

Sobre listas duplamente encadeadas, afirma-se:

I) Cada nó usa o dobro do número de campos ponteiro de uma lista simplesmente encadeada.

II) A complexidade de remoção é metade da complexidade de remoção em lista simplesmente encadeada.

III) Não permitem a inserção de nó no meio da lista.

É correto apenas:

☒ I e III.

☐ I.

☐ II.

☐ II e III.

☐ III.

Data Resp.: 15/02/2024 23:38:42

Explicação:

A resposta correta é: I.

3.

(IBFC/2022 - Adaptada) Assinale, das alternativas abaixo, a única que identifica respectivamente uma Estrutura de Dados do tipo FIFO (*First In, First Out*) e uma outra com a Estrutura de dados do tipo LIFO (*Last In, First Out*):

☐ vetor - lista

☐ fila - pilha

☐ pilha - fila

☐ lista - vetor

☒ matriz - vetor

Data Resp.: 15/02/2024 23:38:44

Explicação:

Filas e Pilhas são estruturas de dados lineares que permitem o armazenamento e acesso a elementos. Ambas são utilizadas para armazenar coleções de elementos, mas possuem diferenças importantes no que diz respeito ao princípio de acesso aos elementos.

Uma fila é uma estrutura de dados baseada no princípio *First In, First Out* (FIFO), ou seja, o primeiro elemento a ser adicionado é o primeiro a ser removido. É como se fosse uma fila de pessoas, onde a primeira pessoa a chegar é a primeira a ser atendida. As operações de inserção e remoção são chamadas de enfileirar e desenfileirar, respectivamente.

Já uma pilha é uma estrutura de dados baseada no princípio *Last In, First Out* (LIFO), ou seja, o último elemento a ser adicionado é o primeiro a ser removido. É como se fosse uma pilha de pratos, onde o último prato colocado é o primeiro a ser retirado. O LIFO é a base para as operações de empilhamento e desempilhamento, que permitem adicionar e remover elementos na pilha, respectivamente.

4.

Uma lista ordenada alocada sequencialmente possui como desvantagem:

- ☐ Tamanho limitado de memória alocada para lista.
- ☒ Impossibilidade de remoção no meio da lista.
- ☐ Complexidade $O(n)$ para a busca.
- ☐ Impossibilidade de acesso direto.
- ☐ A reserva de memória em posições contíguas.

Data Resp.: 15/02/2024 23:38:46

Explicação:

A resposta correta é: Tamanho limitado de memória alocada para lista.

5.

(IBADE/2022) Uma estrutura de dados onde existe uma coleção ordenada de entidades sendo a metodologia de busca com base no deslocamento relativo ao primeiro (cabeça) da coleção, chama-se:

- ☐ pilha.
- ☐ árvore.
- ☒ lista.
- ☐ fila.
- ☐ árvore binária.

6.

(FCC/ 2013) Insira os dados de entrada numa fila. Em seguida, retire cada dado da fila e insira numa pilha. Mostre a pilha. Depois retire os dados da pilha e insira na fila. Mostre a fila.

Dados de entrada: 11, 12, 23, 14, 25, 50, 8, 18, 29, 10

As estruturas mostradas ficam

I. Pilha: (topo) 10 - 29 - 18 - 8 - 50 - 25 - 14 - 23 - 12 - 11

II. Fila: (começo) 11 - 12 - 23 - 14 - 25 - 50 - 8 - 18 - 29 - 10 (fim)

III. Fila: (começo) 10 - 29 - 18 - 8 - 50 - 25 - 14 - 23 - 12 - 11 (fim)

IV. Pilha: (topo) 11 - 12 - 23 - 14 - 25 - 50 - 8 - 18 - 29 - 10

V. A fila mostrada fica com os elementos em ordem invertida dos dados de entrada

Está correto o que se afirma APENAS em:

- ☐ I, IV e V.
- ☐ II e IV.
- ☒ III e IV.
- ☐ I, III e V.
- ☐ I, II e III.

Data Resp.: 15/02/2024 23:38:51

Explicação:

Ao inserir na fila temos: 10,29,18,8,50,25,14,23,12,11

Ao inserir na pilha temos:

10
29
18
8
50
25
14
23
12
11

Retirando os dados da pilha e inserindo na fila: 10,29,18,8,50,25,14,23,12,11

7.

Várias estruturas de dados podem ser utilizadas para armazenar dados de um aplicação. Em relação ao assunto, assinale a alternativa correta.

- ☐ A estrutura de dados do tipo pilha, sempre retira os elementos que foram inseridos primeiro na estrutura.
- ☐ A estrutura de dados do tipo pilha, para retirar o elemento do topo da pilha, é necessário retirar o elemento da base da pilha.
- ☐ A estrutura de dados do tipo fila, sempre retira os elementos que entraram por último na fila.
- ☒ A estrutura de dados do tipo fila, utiliza a ideia do primeiro a ser inserido, será o primeiro a ser retirado.
- ☐ A estrutura de dados do tipo lista, utiliza a ideia do primeiro a chegar, primeiro a ser servido para inserir elementos.

Data Resp.: 15/02/2024 23:39:02

Explicação:

A resposta correta é: A estrutura de dados do tipo fila, utiliza a ideia do primeiro a ser inserido, será o primeiro a ser retirado.

 Não Respondida

 Não Gravada

 Gravada

Exercício por Temas iniciado em 15/02/2024 23:37:55.

ORDENAÇÃO

1.

Sobre o método da bolha é correto afirmar que:

- ☒ O tempo de execução pode ser linear em relação ao tamanho da entrada se a instância apresentada já estiver ordenada.
- ☐ O tempo de execução pode ser linear em relação ao tamanho da entrada se a instância apresentada estiver ordenada em ordem reversa a desejada.
- ☐ O tempo de execução é definido pela complexidade computacional sempre, independentemente da instância apresentada.
- ☒ A complexidade computacional deste algoritmo é $O(n \log n)$.
- ☐ O algoritmo executa sempre no mesmo tempo para instâncias de mesmo tamanho n .

Data Resp.: 15/02/2024 23:39:08

Explicação:

A resposta correta é: O tempo de execução pode ser linear em relação ao tamanho da entrada se a instância apresentada já estiver ordenada.

2.

(CESGRANRIO/2014) Considere utilizar o algoritmo Bubble Sort para ordenar, em ordem crescente, a sequência de números

17, 43, 37, 31, 8, 77, 52, 25.

Se a sequência original for a iteração zero, qual será a sequência de números da segunda iteração?

- ☒ 17, 25, 37, 31, 8, 43, 52, 77
- ☐ 17, 31, 37, 43, 8, 77, 52, 55
- ☐ 8, 17, 43, 37, 31, 77, 52, 25
- ☐ 17, 31, 8, 25, 37, 43, 77, 52
- ☐ 17, 31, 8, 37, 43, 25, 52, 77

Data Resp.: 15/02/2024 23:39:10

Explicação:

Para resolver essa questão, vamos simular a execução do algoritmo Bubble Sort na sequência dada e acompanhar as trocas feitas na segunda iteração.

Sequência original: 17, 43, 37, 31, 8, 77, 52, 25

Primeira iteração:

17, 37, 31, 8, 43, 52, 25, 77 (comparação e troca: 43 e 37)

17, 31, 8, 37, 43, 25, 52, 77 (comparação e troca: 37 e 31)

17, 31, 8, 37, 25, 43, 52, 77 (comparação e troca: 43 e 25)

17, 31, 8, 37, 25, 43, 52, 77 (sem trocas)

Segunda iteração (a partir da sequência obtida na primeira iteração):

17, 31, 8, 37, 25, 43, 52, 77 (comparação e troca: 31 e 8)

17, 8, 31, 37, 25, 43, 52, 77 (comparação e troca: 37 e 8)

17, 8, 31, 25, 37, 43, 52, 77 (sem trocas)

17, 8, 31, 25, 37, 43, 52, 77 (sem trocas)

A sequência obtida na segunda iteração é: 17, 8, 31, 25, 37, 43, 52, 77.

3.

(MPE-BA/2023) Algoritmos de ordenação são responsáveis por ordenar elementos de uma estrutura de dados de forma completa ou parcial. Sobre a complexidade dos algoritmos de ordenação, assinale, a seguir, o algoritmo de ordenação que, no pior caso, tem complexidade igual a $O(n \log n)$.

- ☐ Quick sort.

- ☐ Insertion sort.
- ☒ Bubble sort.
- ☐ Merge sort.
- ☐ Selection sort.

Data Resp.: 15/02/2024 23:39:17

Explicação:

O Merge sort é um algoritmo de divisão e conquista que divide repetidamente a lista em sublistas menores, até que cada sublista contenha apenas um elemento. Em seguida, ele combina as sublistas em pares, ordenando-as ao mesmo tempo, e continua mesclando-as em sublistas maiores até que a lista esteja completamente ordenada. A complexidade do Merge sort no pior caso é $O(n \log n)$, onde "n" é o número de elementos a serem ordenados.

4.

Todos os algoritmos de ordenação interna devem ter complexidade de espaço de:

- ☐ $O(n^2)$
- ☐ $O(n^3)$
- ☐ $O(n \log n)$
- ☒ $O(n)$
- ☐ $O(1)$

Data Resp.: 15/02/2024 23:39:20

Explicação:

A resposta correta é: $O(n)$

```

1. // laço com a quantidade de elementos do vetor (n = 10)
2. for(n = 1; n <= 10; n++) {
3.     for(i = 0; i <= 8; i++) {
4.         if(vet[i] > vet[i+1]) {
5.             aux = vet[i];
6.             vet[i] = vet[i+1];
7.             vet[i+1] = aux;
8.         }
9.     }
10. }
```

5.

(UEAP/2014) Leia o seguinte trecho de código-fonte escrito em Java.

O algoritmo de ordenação implementado nesse trecho de código-fonte é o

- ☒ Merge Sort.
- ☐ May Sort.
- ☐ Quick Sort.
- ☐ Bubble Sort.

Explicação:

Bubble Sort indica um algoritmo de ordenação de elementos, que é um dos algoritmos mais simples e conhecidos para ordenar um conjunto de dados. O Bubble Sort funciona percorrendo a lista de elementos várias vezes, comparando pares de elementos adjacentes e trocando-os se estiverem na ordem errada. O processo é repetido até que nenhuma troca seja necessária, o que significa que a lista está ordenada.

6.

(FUMARC/2014 - Adaptada) Considere uma estrutura de dados do tipo vetor de tamanho 6 (seis) inicializado com as chaves abaixo:

	1	2	3	4	5	6
Vetor	79	82	68	69	78	65
Passo 1	65	82	68	69	78	79
Passo 2	65	68	82	69	78	79
Passo 3	65	68	69	82	78	79
Passo 4	65	68	69	78	82	79
Passo 5	65	68	69	78	79	82

Analise a ilustração dos passos intermediários de um método de ordenação das chaves do vetor:

O método de ordenação apresentado é:

- ☐ Ordenação por inserção.
- ☐ Quicksort.
- ☐ Shellsort.
- ☒ Ordenação por seleção.
- ☐ Mestrellasort.

Data Resp.: 15/02/2024 23:39:30

Explicação:

A ordenação por seleção é um algoritmo de ordenação que consiste em encontrar o menor valor no vetor e colocá-lo na primeira posição (ou o maior valor, dependendo da ordem desejada). Em seguida, encontra-se o segundo menor valor e o coloca na segunda posição, e assim por diante, repetindo o processo para os (n-1) elementos restantes, até os últimos dois elementos.

☐ Não Respondida

☐ Não Gravada

☐ Gravada

Exercício por Temas iniciado em 15/02/2024 23:37:58.

1.

Ano: 2015 Banca: NUCEPE Órgão: SEFAZ - PI Prova: Analista - Sistemas Pleno

Árvore AVL é uma árvore de busca autobalanceada. Isso significa que:

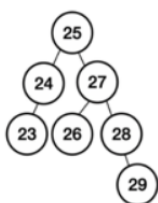
- ☐ As alturas das duas subárvores a partir de cada nó são exatamente iguais.
- ☒ Pode possuir até duas raízes.
- ☐ Cada nó da árvore possui até três descendentes.
- ☐ As alturas das duas subárvores a partir de cada nó diferem no máximo em uma unidade.
- ☐ As alturas das duas subárvores a partir de cada nó diferem no máximo em duas unidades.

Data Resp.: 15/02/2024 23:39:39

Explicação:

Resposta correta: as alturas das duas subárvores a partir de cada nó diferem no máximo em uma unidade.

2.



Seja a seguinte árvore binária de busca abaixo, marque a sequência correta do percurso em pós-ordem:

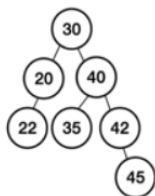
- ☐ 23,24,25,26,27,29,28
- ☐ 23,24,26,29,28,27,25
- ☐ 25,24,23,27,26,28,29
- ☒ 23,24,29,26,28,25,27
- ☐ 25,24,27,23,26,29,30

Data Resp.: 15/02/2024 23:39:42

Explicação:

O percurso é definido pela recursão, percorrer recursivamente a esquerda, percorrer recursivamente a direita da raiz considerada e, finalmente, visitar a raiz.

3.



Seja a seguinte árvore, marque a opção correta que indica o porquê a árvore abaixo não é uma árvore binária de busca:

- ☐ Não é uma árvore binária de busca pois esta árvore deve estar com os níveis de suas folhas todas igualmente perfeitas.
- ☒ Não é uma árvore binária de busca pois o nó 35 deveria estar inserido à direita do nó 20.
- ☐ Não é uma árvore binária de busca pois está desbalanceada.
- ☐ Não é uma árvore binária de busca pois o nó 22 deveria estar inserido à direita do nó 20.

- ☐ Não é uma árvore binária de busca pois essa árvore deve estar perfeitamente balanceada.

Data Resp.: 15/02/2024 23:39:47

Explicação:

Uma árvore binária de busca são árvores que obedecem às seguintes propriedades:

- Dado um nó qualquer da árvore binária, todos os nós à esquerda dele são menores ou iguais a ele.
- Dado um nó qualquer da árvore binária, todos os nós à direita dele são maiores ou iguais a ele.

Observe que a sub-árvore 20-22 não respeita a regra básica, portanto, o nó 22 deveria estar a direita do nó 20.

4.

As árvores binárias de busca são especializações das árvores binárias que permitem uma melhor organização dos algoritmos de busca. Sobre a inserção de uma nova chave em uma árvore binária de busca é correto afirmar que:

- ☐ O algoritmo de inserção em árvores binárias de busca é estático, isto é, é necessário recalcular toda árvore para inserir uma nova chave.
- ☐ Todas as chaves são inseridas em folhas, a posição da folha é determinada pela busca.
- ☒ Toda nova chave é inserida obrigatoriamente na raiz.
- ☐ Para determinar a posição da nova chave é necessário calcular o percurso em ordem simétrica da árvore obtida. Com este percurso, verifica-se se a sequência está ordenada em ordem crescente. Caso esteja, a posição da nova chave está correta.
- ☐ A complexidade da inserção é sempre $O(n)$, independentemente da altura da árvore.

Data Resp.: 15/02/2024 23:39:50

Explicação:

Todas as chaves serão inseridas em folhas, considerando que na árvore binária de busca o objetivo é buscar pela posição correta de inserção dos dados e inserir (a esquerda ou a direita) o nó considerado, além de que em árvores binárias de busca não existem regras que mantenham o balanceamento da árvore.

```
class NoArvore:
    def __init__(self, chave = None, esquerda = None, direita = None):
        self.chave = chave
        self.esquerda = esquerda
        self.direita = direita

if __name__ == '__main__':
    raiz = NoArvore(55)

    raiz.esquerda = NoArvore(35)
    raiz.direita = NoArvore(75)

    raiz.direita.esquerda = NoArvore(65)
    raiz.direita.direita = NoArvore(85)
    raiz.esquerda.esquerda = NoArvore(25)
    raiz.esquerda.direita = NoArvore(45)
```

5.

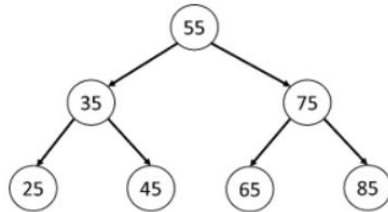
Seja o seguinte código em Python cujo principal objetivo é implementar uma árvore binária. Marque a alternativa correta quanto a execução do código:

- ☐ A árvore criada no código acima não é binária de busca.
- ☐ A classe NotArvore implementa regras que garantem que os nós inseridos respeitam a ordem de inserção dos nós (maiores a direita e menores a esquerda).
- ☒ A árvore criada no código acima é uma árvore binária de busca com todas as folhas no último nível.
- ☒ A árvore criada no código é binária de busca com altura 6, isto é, com 6 níveis distintos.
- ☐ Não é possível inferir a topologia da árvore com base no código.

Data Resp.: 15/02/2024 23:39:54

Explicação:

Apesar de que a função NoArvore não tem regras quanto à inserção dos nós, a forma (e a ordem) de como os nós foram inseridos (55-35-75-65-85-25-45) formam uma árvore binária de busca.



Resultado:

☒ Não Respondida ☒ Não Gravada ☒ Gravada

Exercício por Temas iniciado em 15/02/2024 23:38:02.