

Disc.: **ESTRUTURA DE DADOS**Acertos: **2,0** de 2,0

12/09/2023

**1ª** QuestãoAcerto: **0,2 / 0,2**

O uso de funções recursivas pode facilitar a implementação de diversos algoritmos. Toda recursão depende de dois elementos: o caso base e o passo recursivo. Dentre as opções a seguir, a que apresenta um passo recursivo é:

- ☒  $\text{fat}(n) = n * \text{fat}(n-1)$
- ☐  $\text{par}(n) = \text{par}(n)$
- ☐  $\text{fib}(n) = n-1 + n-2$
- ☐  $\text{fat}(1) = 1$
- ☐  $f(n) = g(n-1)$

Respondido em 12/09/2023 02:25:25

**Explicação:**

O passo recursivo é o elemento que faz o cálculo da função recursiva mover-se em direção ao resultado. Deve envolver a chamada da própria função com um valor diferente de entrada. Isso só acontece na resposta correta:  $\text{fat}(n) = n * \text{fat}(n-1)$ , passo recursivo da função de cálculo de fatorial.

$\text{fat}(1) = 1$  é o caso base dessa mesma função.  $\text{par}(n) = \text{par}(n)$  é uma tautologia, e não uma recursão. As demais respostas são funções que não chamam a si mesmas, não podendo ser passos recursivos.

**2ª** QuestãoAcerto: **0,2 / 0,2**

Uma lista L encadeada e ordenada está armazenada em memória seguindo o exemplo abaixo. Após a remoção do nó de chave 3, quais alterações terão ocorrido?

Endereço	Chave	Próximo
128	5	64
64	8	32
32	11	null
L-----> 24	3	128

- ☐ O endereço 24 conterá a chave 5 e próximo 64.
- ☐ O conteúdo armazenado no endereço 32 será apagado.
- ☐ L terá sido apagada.
- ☒ A variável L apontará para 128.
- ☐ O endereço 32 terá seu campo próximo apontando para 24.

Respondido em 12/09/2023 02:26:21

#### Explicação:

A remoção solicitada é do primeiro elemento da lista encadeada. Para realizar esse tipo de remoção, basta apontar a variável que guarda o primeiro elemento (L) para o endereço do segundo elemento. Este endereço está armazenado no campo próximo do primeiro elemento. Ou seja, a variável L deverá apontar para 128.

A resposta endereço 24 conterá a chave 5 está errada pois na lista encadeada, os elementos não precisam ser puxados após uma remoção.

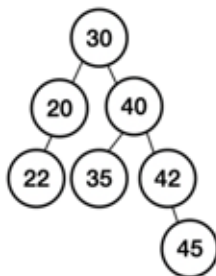
A resposta endereço 32 terá seu campo próximo alterado está errada, pois isso adicionaria um elemento ao final da lista, no caso tornando-a circular.

As demais respostas estão erradas pois nada será apagado.

#### 3ª Questão

Acerto: 0,2 / 0,2

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



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

**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 à direita do nó 20.



4ª Questão

Acerto: 0,2 / 0,2

As árvores AVL constituem uma importante estrutura de dados que disponibilizam operações de busca, inserção e remoção. Classifique como verdadeiro ou falso as afirmativas abaixo:

I - As árvores de Fibonacci são as árvores de altura máxima  $h$  com número mínimo de nós  $n$  e altura proporcional a  $\log n$ .

II - As árvores completas são árvores AVL.

III - É possível construir uma topologia de uma árvore AVL que não seja nem completa nem de Fibonacci com altura proporcional a  $\log n$ .

IV - Uma vez que a altura das árvores AVL é proporcional a  $\log n$ , podemos garantir que a busca ocorre numa complexidade de  $O(\log n)$ .

V - Na remoção, pode ser necessário realizar todas as rotações, no pior caso, do pai de uma folha que está sendo removida até a raiz. Por esta razão, a complexidade da remoção é maior que  $O(\log n)$ .

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

Respondido em 12/09/2023 02:28:41

**Explicação:**

Nem sempre é necessário realizar todas as operações, visto que a remoção pode eliminar uma folha e não causar desbalanceamento na árvore.



5ª Questão

Acerto: 0,2 / 0,2

Ao usar a biblioteca numpy para criar arrays, existem diversas facilidades que um programador pode utilizar, como funções específicas para somar todos os elementos, encontrar valores mínimo e máximo dos elementos, entre outros.

Entretanto uma desvantagem de usar array da biblioteca numpy é:

- ☒ Todos os elementos devem ter o mesmo tamanho.
- ☐ Não é possível adicionar novos elementos ao array.
- ☐ Diminuição no tempo de programação.
- ☐ Não é possível remover elementos do array.
- ☐ Os índices passam a ser contados a partir de 1.

Respondido em 12/09/2023 02:29:16

**Explicação:**

A desvantagem é que os elementos do array devem ocupar o mesmo espaço de memória, então devem ser de mesmo tamanho. Isso não permite que você crie arrays com elementos de tamanho assimétricos. Os índices continuam sendo contados a partir de 0 e as operações de inserção e remoção continuam sendo possíveis. A diminuição no tempo de programação é uma vantagem.

## 6ª Questão

Acerto: 0,2 / 0,2

Uma Deque é uma estrutura de dados mais generalista que as pilhas e filas. Para implementá-la de forma eficiente, você pode usar:

- ☐ Fila com 2 variáveis: início e final.
- ☐ Lista simplesmente encadeada com nó cabeça.
- ☒ Lista duplamente encadeada com 2 variáveis: início e final.
- ☐ Lista contígua com 1 variável: início.
- ☐ Pilha com 1 variável: topo.

Respondido em 12/09/2023 02:30:02

### Explicação:

Para implementar uma deque eficientemente, você precisa ter um ponteiro para o início e o final da deque, permitindo inserções e remoções em ambas as pontas com complexidade  $O(1)$ , sem a necessidade de percorrer a estrutura, o que seria  $O(n)$ .

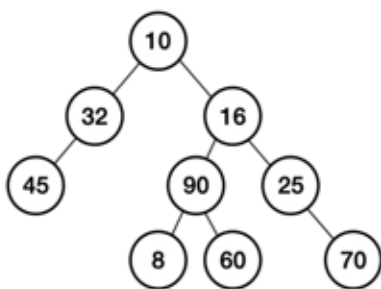
Além disso, a fila é uma especialização da deque. Ou seja, toda fila é um deque, mas nem toda deque é uma fila. Podemos assim eliminar a resposta contendo fila. A resposta restante que possui 2 variáveis é a correta. Lista duplamente encadeada. Ela permite a inserção e remoção nas extremidades com complexidade  $O(1)$ .

A lista contígua e a simplesmente encadeada com nó cabeça levariam a operação de inserção e remoção ao final da fila terem complexidade  $O(n)$  por precisarem percorrer toda a estrutura, sendo também descartadas.

## 7ª Questão

Acerto: 0,2 / 0,2

Seja a seguinte Árvore Binária. Marque a opção correta:



- ☐ A árvore acima possui raiz de valor 3.
- ☐ É possível inserir mais um filho a esquerda no nó de valor 90.
- ☒ A quantidade de folhas da árvore é 4.
- ☐ A quantidade de nós da árvore é de  $n \geq 1$ , sem considerar o nó raiz.
- ☐ Não é possível inserir nós filhos ao nó 70.

Respondido em 12/09/2023 02:30:39

### Explicação:

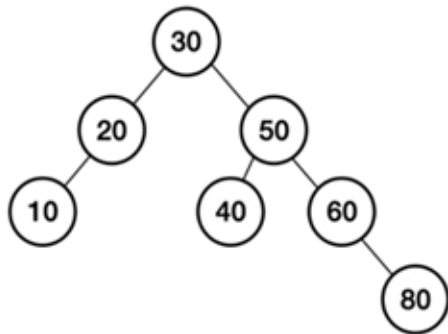
A quantidade de folhas da árvore é 4, ou seja, são aqueles nós que possuem grau zero.



8ª Questão

Acerto: 0,2 / 0,2

Seja a seguinte árvore AVL abaixo. Com a inserção da chave 90, marque a opção que indica exatamente o que acontecerá com a árvore resultante após essa inserção:

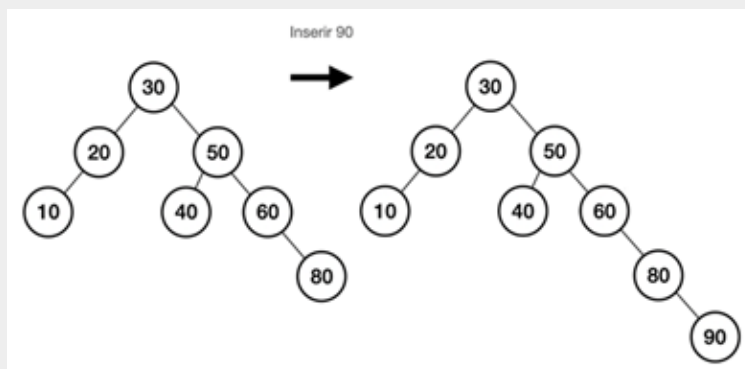


- ☐ A árvore resultante irá manter o balanceamento geral da árvore.
- ☐ A árvore resultante irá desbalancear à esquerda do nó de chave 10.
- ☐ A árvore resultante irá desbalancear à direita do nó de chave 40.
- ☐ A árvore resultante irá desbalancear à direita do nó de chave 80.
- ☒ A árvore resultante irá desbalancear à esquerda do nó de chave 60.

Respondido em 12/09/2023 02:32:17

#### Explicação:

Ao inserir o nó de chave 90, ele é maior que o nó 80, sendo assim, inserido ao lado direito de 80, causando desbalanceamento do nó 60 que tem altura da subárvore direita 2 e esquerda 0.



9ª Questão

Acerto: 0,2 / 0,2

Dada a seguinte matriz M, inicializada com o código:

M=[[1,2,3,4],[5,6,7,8],[9,10,11,12],[13,14,15,16]]

O código em Python para imprimir cada elemento da coluna iniciada pelo elemento 3 é:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

- ☒ `for linha in M:`  
`print(linha[2])`

- ☐ for linha in M:  
    print(linha[3])
- ☐ for linha in M:  
    print(linha)
- ☐ print(M[2])
- ☐ for coluna in M:  
    print(coluna)

Respondido em 12/09/2023 02:34:02

**Explicação:**

O laço deve percorrer uma coluna, iterando linha a linha e extraíndo dela o seu terceiro elemento, ou seja linha[2]. A resposta correta itera pelas linhas e imprime o elemento [2] de cada uma.

Dentre as respostas erradas, apenas escrever `print(linha)` imprimirá cada linha como um todo, resultando na impressão de toda a matriz, linha a linha.

A resposta "print(coluna)" terá o mesmo resultado pois para o código linha e coluna são apenas nomes escolhidos pelo programador. Poderia ser i, aux ou qualquer outra variável escolhida.

Já "print(linha[3])" está com o índice errado, imprimindo os elementos da coluna iniciada por 4. E `print(M[2])` imprime toda a linha iniciada por 9.

**10ª** Questão

Acerto: **0,2 / 0,2**

Em uma implementação da estrutura de dados do tipo fila, você possui um espaço de memória contíguo a ela alocada com capacidade para M nós. A variável da fila é F, e duas variáveis guardam os índices do início e final da fila (inícioF e finalF). Em uma implementação otimizada de F, como podemos identificar que a fila está cheia?

- ☐ InícioF = M
- ☒ InícioF==(finalF+1)mod M
- ☐ FinalF== M
- ☐ InícioF== finalF
- ☐ InícioF==finalF + 1

Respondido em 12/09/2023 02:34:45

**Explicação:**

Em uma implementação otimizada da fila, é usado um sistema modular, onde o início e o final da fila se movem a cada inserção e remoção. A cada inserção, finalF aumenta em 1, até o máximo M, depois volta para 0 e assim por diante. A cada remoção inícioF aumenta em 1, até o máximo M e depois volta a 0. dessa forma a fila está cheia quando  $(finalF+1)modM$  é igual a início.