

AVALIAÇÃO DE CONHECIMENTO ESPECÍFICO

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

01. Leia as afirmações abaixo sobre Listas implementadas por contiguidade física e marque a opção **CORRETA**.

- I. Apresenta como principal vantagem o acesso direto a qualquer elemento da Lista;
- II. Apresenta a desvantagem a necessidade de pré-estimar o tamanho da Lista.
- III. Apresenta tempo constante para acessar qualquer elemento da Lista;

- a) Apenas I está correta.
- b) Apenas II está correta.
- c) Apenas III está correta.
- d) Apenas I e III estão corretas.
- e) I, II e III estão corretas.

02. Considerando as Listas Simplesmente Encadeadas é **CORRETO** afirmar que:

- a) As operações de Inserção no início e Remoção no início são as mais custosas computacionalmente.
- b) As operações de Inserção no início e Remoção no fim são as mais custosas computacionalmente.
- c) As operações de Inserção no fim e Remoção no fim são as mais custosas computacionalmente.
- d) As operações de Inserção no fim e Remoção no início são as menos custosas computacionalmente.
- e) Todas as alternativas estão corretas.

03. Sobre Listas Encadeadas Dinâmicas, leia as afirmações abaixo e marque a opção **INCORRETA**:

- a) Para determinar a quantidade de elementos de uma Lista Encadeada Dinâmica é necessário percorrer toda a Lista para fazer a contagem dos elementos.
- b) Uma desvantagem da Listas Encadeadas Dinâmicas é o fato de que seus elementos sempre ocuparem um espaço contíguo de na memória.
- c) Uma Lista Encadeada Dinâmica pode ser representada por um ponteiro para o seu primeiro elemento.
- d) Em uma Lista Encadeada Dinâmica, para cada elemento inserido na estrutura, aloca-se um espaço de memória para armazená-lo.
- e) Um nó em uma lista encadeada dinâmica é representado por uma estrutura que contém, conceitualmente, dois campos: informação armazenada e ponteiro para o próximo elemento da lista.

04. Analise as sentenças abaixo sobre Listas Duplamente Encadeadas e marque a opção **CORRETA**.

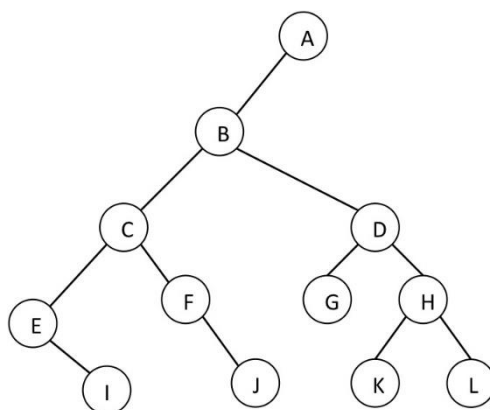
- I. Uma lista duplamente encadeada é aquela que possui duas referências ao invés de uma, permitindo que a lista seja percorrida nos dois sentidos.
- II. Na inserção de um novo nó em uma lista duplamente encadeada só é necessário atualizar o campo elo do nó anterior.
- III. A principal característica das listas duplamente circulares é que o primeiro nó da lista tem como antecessor o último nó, e o último, por sua vez, tem o primeiro nó como o seu próximo.

- a) Apenas I está correta.
- b) Apenas II está correta.
- c) Apenas III está correta.
- d) Apenas I e III estão corretas.
- e) I, II e III estão corretas.

05. Tendo em vista uma Lista de nomes Simplesmente Encadeada Circular com três elementos é **CORRETO** afirmar que:

- a) Todos os Nós apontam para o início da Lista.
- b) Nenhum dos Nós aponta para o início da Lista.
- c) Todos os Nós são definidos com apenas um campo apontador.
- d) Todos os Nós são definidos com dois campos apontadores.
- e) Apenas o último nó deve conter dois campos apontadores.

06. Considerando a árvore binária da figura abaixo, é correto afirmar que seu percurso em pós-ordem resulta em:



- a) ABCEIFJDGHKL
- b) IEJFCGKLHDBA
- c) LHKDGBCFJAEI
- d) IEJFCLHDBKGA.
- e) EICFJBGDKHLA.

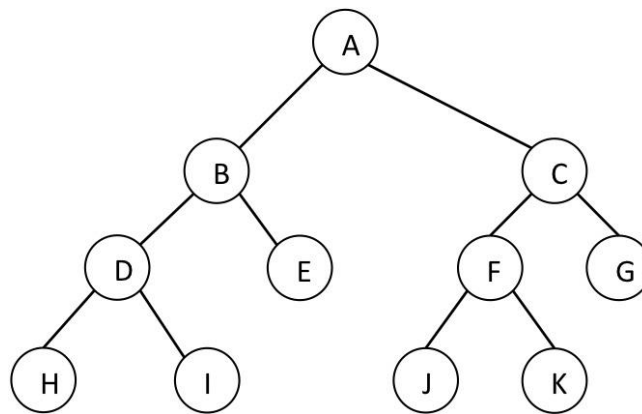
07. Analise os itens abaixo sobre árvores AVL:

- I. Para qualquer nó de T as alturas de suas duas sub-árvores, direita e esquerda, diferem em módulo de até uma unidade
- II. Toda árvore AVL é completa
- III. Toda árvore completa é AVL
- IV. Todos os nós de uma árvore AVL são nós regulados
- V. Toda árvore AVL é balanceada

Assinale a alternativa correta.

- a) Todas as alternativas são corretas
- b) As alternativas III e V são corretas
- c) As alternativas I, II, IV e V são corretas
- d) Somente a alternativa III é errada
- e) As alternativas I e V são erradas

08. Da árvore representada na figura abaixo pode-se dizer que:

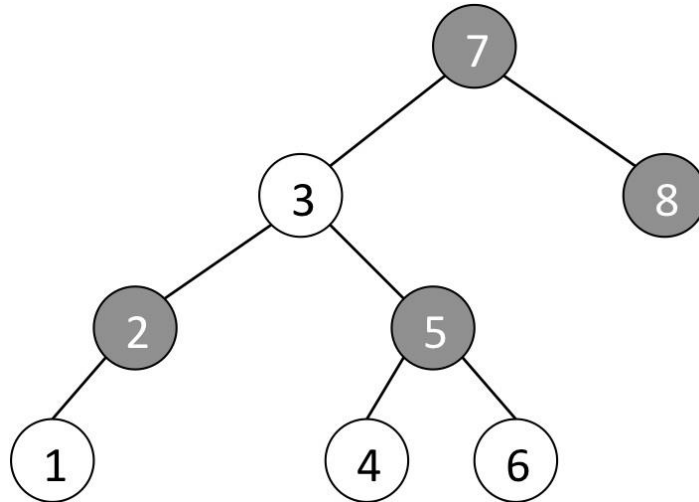


- I. É uma árvore estritamente binária
- II. A profundidade da árvore é 3
- III. É uma árvore binária quase completa de profundidade 3
- IV. O nó E é descendente dos nós B e A não é descendente do nó C
- V. A leitura em pós-ordem resulta em HIDEBJKFGCA

Assinale a alternativa correta.

- a) Todas as alternativas estão corretas
- b) As alternativas I, II e IV são corretas
- c) Somente a alternativa I está errada
- d) Somente a alternativa V está errada
- e) As alternativas I, IV e V são corretas

09. Indique qual a ordem de inserção dos nós de valores 1, 2, 3, 4, 5, 6, 7, 8, não necessariamente nesta ordem, que resulte na árvore rubro-negra da figura abaixo. OBS: Na figura abaixo nós de cor escura são negros (pretos) e nós de cor clara são rubros (vermelhos).



Assinale a alternativa correta.

- a) 78342165
- b) 14625378
- c) 12345678
- d) 73218546
- e) 37825146

10. Com relação a árvore rubro-negra produzida pela inserção dos nós 4, 2, 1, 6, 7, 8, 5, 3, é correto afirmar que:

- I. É uma árvore balanceada
- II. A profundidade da árvore é 3
- III. Apenas os nós 3, 5, 6, 8 são rubros (vermelhos)
- IV. O nó 2 é a raiz
- V. O nó 1 é descendente do nó 6

Assinale a alternativa correta.

- a) Somente a alternativa III está correta
- b) Somente as alternativas I, IV e V são corretas
- c) Todas as afirmações são corretas
- d) Somente as alternativas II e III erradas
- e) Somente as alternativas I, V são erradas

11. Notações assintóticas são utilizadas para representar o comportamento assintótico das funções de complexidade de tempo dos algoritmos, bem como relacionar o comportamento das funções de complexidade de dois algoritmos. Uma vez identificadas quais são as notações geralmente utilizadas para denotar o tempo de execução de um algoritmo, pode-se seguir alguns princípios para analisar programas. Com relação aos princípios de análise, afirma-se:

- I. O tempo de execução de um comando de atribuição pode ser considerado constante;
- II. O tempo de execução de uma sequência de comandos é determinado pelo maior tempo de execução de qualquer comando da sequência;
- III. Quando o programa possui procedimentos recursivos, o tempo de execução de cada procedimento é computado separadamente.

Baseado no exposto, pode-se concluir que:

- a) Somente a afirmativa I é Verdadeira;
- b) Todas as afirmativas são Falsas;
- c) Somente as afirmativas I e III são verdadeiras;
- d) Todas as afirmativas são Verdadeiras;
- e) Somente a Afirmativa III é Falsa.

12. Com relação a Algoritmos, afirma-se:

- I. Um algoritmo é dito correto se, para cada instância de entrada, ele pára com a saída correta;
- II. A escolha de um algoritmo de ordenação para uma determinada aplicação depende, entre outros fatores, do número de itens a serem ordenados, da extensão em que os itens já estão ordenados de algum modo, de possíveis restrições sobre os valores de itens e da espécie de dispositivo de armazenamento a ser usado: memória principal, discos etc.
- III. Algoritmos criados para resolver o mesmo problema caracterizam-se por sempre possuírem a mesma eficiência;
- IV. O conjunto de problemas NP-completos tem a propriedade de que, se existe um algoritmo eficiente para qualquer um deles, então existem algoritmos eficientes para todos.

Baseado no exposto, pode-se concluir que:

- a) Somente as afirmativas I e II são Verdadeiras;
- b) Todas as afirmativas são Falsas;
- c) Somente as afirmativas I e III são verdadeiras;
- d) Todas as afirmativas são Verdadeiras;
- e) Somente a Afirmativa III é Falsa.

13. Com relação às expressões booleanas (em algoritmos ou em linguagens de programação), é **FALSO** afirmar:

- a) Comandos de seleção não podem ser controlados por expressões booleanas;
- b) Uma expressão booleana é qualquer expressão que seja ou verdadeira ou falsa;
- c) Expressões booleanas podem ser usadas para controlar laços de repetição;
- d) Um comando if-else (se-senão) escolhe entre dois caminhos alternativos, com base no valor de uma expressão booleana;
- e) As expressões booleanas não precisam ficar todas entre parênteses. Se os parênteses forem omitidos, adota-se a precedência-padrão.

14. Com relação à Programação Orientada a Objetos (POO), é **FALSO** afirmar:

- a) A herança é uma característica importante da POO, pois permite que um objeto receba atributos e métodos ("comportamentos") de outro objeto;
- b) Uma classe é um gabarito que define atributos e métodos de um objeto;
- c) Corpo do método é a parte do método contendo instruções que serão executadas quando o método for acionado;
- d) Sobrecarga de métodos significa que dois ou mais métodos são definidos com nomes distintos, mas com a mesma lista de argumentos (parâmetros);
- e) O encapsulamento permite ao programador implementar um sistema de restrições e verificações, colocando os atributos e os procedimentos dentro de uma classe e definindo regras dentro da própria classe para controlar o acesso a eles.

15. É **FALSO** afirmar que:

- a) Analisar um algoritmo significa estimar os recursos computacionais requeridos pelo algoritmo;
- b) O tempo de execução do pior caso de um algoritmo é um limite superior sobre o tempo de execução para qualquer entrada;
- c) Os algoritmos que seguem uma abordagem de dividir e conquistar desmembram o problema em vários subproblemas que são semelhantes ao problema original, mas menores em tamanho, resolvem os subproblemas recursivamente e depois combinam essas soluções com o objetivo de criar uma solução para o problema original;
- d) Quando um algoritmo contém uma chamada recursiva a si próprio, seu tempo de execução frequentemente pode ser descrito por uma equação de recorrência, que descreve o tempo de execução global sobre um problema de tamanho n em termos do tempo de execução sobre entradas menores;
- e) Algoritmos de tempo polinomial resolvem problemas em tempo polinomial. Ou seja, esses problemas podem ser resolvidos no tempo $O(n^k)$ para alguma constante n , onde k é o tamanho da entrada para o problema.

16. Sobre recursividade é INCORRETO afirmar que:

- a) Uma implementação recursiva geralmente é mais eficiente em termos de tempo do computador do que uma solução iterativa por causa das operações auxiliares que levam em si as ativações suplementares às funções.
- b) Um subprograma recursivo é um subprograma que é ativado por si mesmo, direta ou indiretamente.
- c) Se uma função, procedimento ou método invoca a si mesmo, este é denominado recursão direta.
- d) A recursividade indireta é produzida quando um subprograma ativa outro, que terminará ativando novamente o primeiro.
- e) Qualquer programa que pode ser resolvido por recursão também pode ser resolvido por iteração.

17. Uma estrutura recursiva que envolve a repetição de um conjunto de instruções pode ser representada pelo algoritmo de:

- a) busca seqüencial.
- b) busca binária.
- c) controle de laço.
- d) ordenação por inserção.
- e) nenhuma das anteriores.

18. Considerando conceitos sobre funções/iteração, julgue os itens abaixo e, em seguida, assinale a opção correta.

- I. Deve-se considerar uma solução recursiva somente quando uma solução iterativa simples não for possível.
- II. Deve-se utilizar uma solução recursiva somente quando a execução e a eficiência da memória da solução estejam dentro dos limites aceitáveis do sistema.
- III. Sendo possível as duas soluções (iterativa e recursiva) a solução recursiva sempre irá requerer mais tempo e mais espaço por causa das ativações adicionais efetuadas.
- IV. A ordem da chamada de funções antes ou após a chamada recursiva pode mudar a execução da recursividade.

Estão CORRETAS as opções:

- a) I e III.
- b) I, II, III.
- c) II e IV
- d) III e IV
- e) Todas estão corretas

19. Observe o pseudocódigo abaixo, referente a um programa de computador que ilustra o uso de uma função recursiva.

```
programa mestrado2013;
variáveis
    num: YZ, NR, PC;
função PIAUÍ(XW:inteira):inteira;
início
    adicionar 1 a PC;
    se XW < 2 então atribuir 4 a PIAUÍ
    senão atribuir PC+PIAUÍ(XW-1) a PIAUÍ;
fim_função
início
    atribuir 3 a NR;
    atribuir 0 a PC;
    atribuir PIAUÍ(NR) a YZ;
    mostrar(YZ,PC);
fim_do_programa
```

Após a execução, a quantidade de vezes que a função foi executada e o valor final para YZ serão, respectivamente:

- a) 4 e 6
 - b) 4 e 7
 - c) 3 e 5
 - d) 3 e 6
 - e) 3 e 7
20. Sobre recursividade na construção de algoritmos, considere o algoritmo apresentado a seguir.

```
função resultado(n)
{
    se (n <= 1)
        retorne 1;
    senão
        retorne n * resultado(n-1);
    fim se;
}
```

Julgue os itens abaixo e, em seguida, assinale a opção correta.

- I. Se a chamada resultado(6) for executada, o retorno será 720.
- II. Se o algoritmo fosse modificado para não utilizar a recursividade, bastaria adicionar mais uma cláusula de condição para que a função não tivesse seu resultado alterado.
- III. A utilização de funções recursivas torna o código mais conciso. Além disso, simplifica a solução de alguns problemas.
- IV. A recursividade, usada incorretamente, poderá acarretar estouro da pilha durante a execução da função.

Estão corretas as opções:

- a) I, II e IV
- b) I, III e IV
- c) Apenas a III

- d) Apenas a IV
- e) Todas estão corretas

21. Qual a estrutura de dados utilizada para implementar a chamada recursiva em funções?

- a) Filas
- b) Árvores
- c) Hash
- d) Pilhas
- e) Listas

22. As operações mais comuns em pilha são: size(), isEmpty(), top(), push(), pop(). Qual a ordem de complexidade dessas operações?

- a) $O(1)$, $O(1)$, $O(1)$, $O(1)$, $O(1)$
- b) $O(n)$, $O(n)$, $O(n)$, $O(n)$, $O(n)$
- c) $O(1)$, $O(1)$, $O(1)$, $O(1)$, $O(n)$
- d) $O(1)$, $O(1)$, $O(n)$, $O(1)$, $O(1)$
- e) Nenhuma das respostas acima

23. A remoção de uma lista inteira, em uma linguagem de programação que utilize garbage collector é de que ordem de complexidade?

- a) $O(n)$
- b) $O(n*n)$
- c) $O(1)$
- d) $O(\log(n))$
- e) Nenhuma das respostas acima.

24. Suponha a execução das seguintes operações em uma fila inicialmente vazia:

enqueue(5), enqueue(3), dequeue(), enqueue(7), size(), dequeue(), front(), dequeue(), isEmpty()

Qual é o resultado apresentado pelas operações size(), front() e isEmpty() na descrição acima?

- a) 2, 3, true
- b) 2, 7, true
- c) 2, 7, false
- d) 1, 7, true
- e) Nenhuma das respostas acima.

25. Assinale a alternativa **CORRETA**.

- I. Listas encadeadas e arrays são exemplos de estruturas de dados que podem ser usadas para implementar FILAS.
- II. Listas duplamente encadeadas não podem implementar o comportamento FILA.
- III. Uma estrutura de dados auxiliar, como uma Hashtable, pode reduzir a complexidade de algumas operações, mas aumenta o consumo de memória.
- IV. Todas as possíveis implementações para o comportamento FILA apresentam complexidade $O(n)$ para a operação de busca.

Sobre as afirmações, é correto afirmar que:

- a) Apenas I e II são verdadeiras
- b) Apenas I e III são verdadeiras
- c) Apenas I e IV são verdadeiras
- d) Apenas II e IV são verdadeiras
- e) Todas são verdadeiras

26. Considere as seguintes afirmações sobre PILHAS:

- I. PILHAS só podem ser implementadas com alocação dinâmica de memória
- II. Árvores não podem implementar o comportamento PILHA
- III. A implementação de PILHA usando um lista duplamente encadeada apresenta complexidade $O(\log-n)$ para busca e remoção
- IV. Arrays possuem a menor complexidade na operação de busca quando usados para implementar uma PILHA

São incorretas as afirmações:

- a) Apenas I e II são verdadeiras
- b) Apenas I e III são verdadeiras
- c) Apenas I e IV são verdadeiras
- d) Apenas II e IV são verdadeiras
- e) Todas são falsas

27. Dada as seguintes afirmativas sobre vetores e matrizes, analise-as e marque a opção **CORRETA**.

- I. Pode-se realizar operações com vetores numéricos da mesma forma como se opera com variáveis numéricas comuns.
- II. Uma matriz de m linhas e n colunas contém $(m * n)$ dados.
- III. Uma matriz é uma coleção de variáveis de tipos distintos, acessíveis com um único nome e armazenados contiguamente na memória.
- IV. Vetor é uma matriz unidimensional
- V. O produto de duas matrizes de n linhas e n colunas resulta em uma matriz de n linhas e n colunas.

- a) Apenas I e II estão corretas.
- b) Apenas II e V estão incorretas.
- c) Apenas I, III e IV estão corretas.
- d) Apenas III está incorreta.
- e) Apenas I e II estão incorreta.

28. Quanto vale k no fim da execução do seguinte trecho de código?

```
k = 0;
for (i=1; i <= n; i++)
  for (j = i; j <= n; j++)
    k = k + 1;
```

- a) $n - 1$
- b) n
- c) $(n^2 - n)/2$
- d) $n(n + 1)/2$
- e) n^3

29. Dada as seguintes afirmativas sobre vetores e matrizes, analise-as e marque a opção **CORRETA**.

- I. Uma matriz é uma coleção de variáveis de um mesmo tipo que é referenciada por um nome comum.
- II. Um vetor é uma coleção de variáveis de um mesmo tipo que é referenciada por um nome comum.
- III. O número de posições do vetor corresponde ao tamanho do vetor, isto é, o número de valores de um tipo que ele armazena.
- IV. Os índices de vetores e matrizes podem ser quaisquer valores inteiros.
- V. Vetores são usados nos casos em que um conjunto de dados do mesmo tipo precisa ser armazenado em uma mesma estrutura

- a) Todas as alternativas estão corretas
- b) Apenas II e V estão incorretas.
- c) Apenas I, III e IV estão corretas.
- d) Apenas IV está incorreta.
- e) Apenas I e II estão incorreta.

30. O que imprime o programa escrito em "C" abaixo?

```
#include <stdio.h>

float f (int n, int c, float a[][2]) {
float t=0;
int i;
int j;

for (i=0; i< n; i++)
    for (j=0; j< c; j++)
        t=t+ a[i][0];
return t/(n*c);
}

float mat [3] [2] = {{5.0, 10.0},{20.0, 25.0},{35.0, 40.0}};

int main() {
    printf ("%f\n", f(3, 2, mat));
}
```

- a) 45
- b) 135
- c) 27
- d) 70
- e) 24