

Prova de Seleção

Programa de Pós-Graduação em Informática (PPGI)
Processo seletivo de 2025/1

Instruções para a prova

- A prova contém **20 questões** objetivas com cinco alternativas cada. Para cada questão, **existe apenas uma alternativa correta**.
- O tempo total para a realização desta prova é de **2 horas**. Um aviso será dado 30 minutos antes do término.
- Você receberá um cartão de respostas oficial para marcar suas respostas. Preencha-o utilizando **caneta esferográfica preta ou azul**.
- Marque apenas **uma resposta por questão**. Caso haja marcações múltiplas, a questão será anulada.
- É **proibido** o uso de celulares, smartwatches ou qualquer outro dispositivo eletrônico durante a prova.
- Guarde **todos** os dispositivos **desligados** e fora do alcance (por exemplo, em sua mochila ou bolsa).
- Apenas canetas, lápis, borracha, documento de identidade e o material fornecido pela organização da prova são permitidos sobre a mesa.
- Caso tenha dúvidas durante a prova, levante a mão e aguarde a presença de um fiscal.
- Os fiscais não estão autorizados a esclarecer questões, apenas dúvidas relacionadas ao processo.
- Ao finalizar, entregue o cartão de respostas ao fiscal. Confira se preencheu corretamente todos os campos obrigatórios no cartão de resposta antes de entregá-lo. Lembre-se também de assinar a lista de presença.
- Mantenha silêncio durante toda a prova para não atrapalhar os colegas.
- Qualquer tentativa de comunicação ou comportamento inadequado poderá resultar na desclassificação.

Boa prova!

Questão 1: Qual das opções abaixo representa o enunciado logicamente equivalente ao enunciado. "Pelo menos uma criança gosta de todas as frutas".

- a) Algumas crianças gostam de algumas frutas
- b) Existe criança que não gosta de nenhuma fruta
- c) Não é o caso que para toda criança, há uma fruta da qual ela não gosta
- d) Algumas crianças não gostam de todas as frutas
- e) Nenhuma das alternativas anteriores

Questão 2: Considerando como verdadeiras as informações:

Nenhum ator gosta de todos os produtores de cinema e Alguns atores gostam de todos os diretores de cinema.

Podemos concluir que:

- a) Não há um produtor de cinema que não seja diretor de cinema
- b) Todo produtor de cinema é diretor de cinema
- c) Há um produtor de cinema que é diretor de cinema
- d) Alguns atores gostam de alguns produtores de cinema
- e) Há pelo menos um produtor de cinema que não é diretor de cinema

Questão 3: Tom, Marcos e João são membros do clube dos Alpes. Cada membro do clube dos Alpes ou é um alpinista, ou é um esquiador, ou ambos. Nenhum alpinista gosta de chuva e todos os esquiadores gostam de neve. Tudo que Tom gosta, Marcos não gosta e Marcos gosta de tudo o que Tom não gosta. Tom gosta de chuva e neve.

Considerando verdadeiras as informações dadas acima, podemos concluir que:

- a) Nenhum membro do clube dos Alpes é esquiador
- b) Todos os membros do clube dos Alpes são esquiadores
- c) Marcos gosta de chuva
- d) Há um membro do clube dos Alpes que é alpinista, mas que não é esquiador
- e) Nenhuma das alternativas anteriores

Questão 4: Considerando como verdadeiras as afirmativas:

- I. Quem canta rock não é estressado
- II. Pessoas calvas não pescam de tarrafa
- III. Quem não pesca de tarrafa é estressado

A sentença que é consequência lógica das anteriores é:

- a) Quem não pesca de tarrafa é calvo
- b) Pessoas calvas não cantam rock
- c) Quem pesca de tarrafa não é estressado
- d) Que não canta rock é estressado
- e) Nenhuma das alternativas anteriores

Questão 5: Considere as premissas a seguir:

- Se Paulo trabalha duro, então ele conseguirá uma promoção
- Paulo trabalha duro e faz cursos de aperfeiçoamento

Nessas condições e considerando as regras de inferência, assinale a alternativa que apresenta a conclusão correta:

- a) Paulo não faz cursos de aperfeiçoamento
- b) Paulo conseguirá uma promoção
- c) Paulo não conseguirá uma promoção e faz cursos de aperfeiçoamento
- d) Paulo não conseguirá uma promoção
- e) Nenhuma das respostas anteriores

Questão 6: Considere o seguinte trecho de código em C-like:

```
int i, j, c;  
c = 1;  
for (i = 1; i < n; i = i*2) {  
    for (j = 1; j <= n; j++) {  
        c = c + 1;  
    }  
}
```

Assumindo que a instrução $c = c + 1$ é $O(1)$, qual a expressão que melhor define a ordem de complexidade do trecho de código acima?

- a) $O(n \log n)$
- b) $O(\log n)$
- c) $O(n)$
- d) $O(n^2)$
- e) $O(n^3)$

Questão 7: Suponha que você esteja trabalhando com a estrutura de dados Pilha, na qual existe uma função para adicionar um tipo de dado na pilha (`push()`) e outra para remover o dado da pilha (`pop()`). Considere as seguintes instruções:

1. `push (A)`
2. `push (B)`
3. `pop ()`
4. `push (C)`
5. `pop ()`

Quais elementos são removidos nas instruções 3 e 5, respectivamente?

- a) B e C
- b) A e B
- c) B e A
- d) A e C
- e) Nenhuma das anteriores

Questão 8: Considere as seguintes afirmações sobre tabelas de dispersão (ou tabela *hash*) e colisões:

- I. O tratamento de colisões é necessário apenas quando a tabela está cheia e se necessita inserir mais uma chave
- II. O tratamento de colisões é necessário para determinar se uma tabela *hash* atingiu o seu limite máximo
- III. O tratamento de colisões é necessário quando a tabela está vazia, pois não é possível calcular o endereço diretamente nesse caso
- IV. O tratamento de colisões é necessário, pois o *hashing* pode gerar repetição de endereço para diferentes chaves

Qual das alternativas a seguir representa a(s) afirmação(ões) correta(s)?

- a) Somente I e II
- b) Somente I, II e IV
- c) Somente II e III
- d) Somente a IV
- e) Nenhuma das alternativas

Questão 9: Estruturas de dados como fila, pilha e lista são muito utilizadas para resolver problemas computacionais. Sobre as características dessas estruturas de dados, atribua V (verdadeiro) ou F (falso) para as afirmativas a seguir.

- () Em uma pilha, o último elemento a entrar é o primeiro a sair
- () Em uma fila, o primeiro elemento a entrar é o último a sair
- () Uma lista permite que as inserções possam ser feitas em qualquer lugar (posição), mas as remoções, não
- () Em uma lista circular com encadeamento simples, o primeiro elemento aponta para o segundo e para o último

() Para remover um elemento de uma lista duplamente encadeada, deve-se alterar o encadeamento dos elementos anterior e próximo ao elemento removido

Assinale a alternativa que contém, de cima para baixo, a sequência correta.

- a. V, F, V, F, V
- b. V, F, F, V, F
- c. V, F, F, F, V
- d. F, V, V, F, F
- e. F, F, V, V, V

Questão 10: Sobre os conceitos de tipos de dados e verificação de tipos de uma linguagem de programação, considere as afirmativas a seguir.

- I. A verificação de tipos em tempo de compilação é menos eficiente do que a verificação em tempo de execução, pois não leva em consideração as condições dinâmicas que podem ocorrer durante a execução do programa
- II. Em uma linguagem de programação com verificação de tipos dinâmica, a correção dos tipos de dados é realizada em tempo de compilação, antes da execução do programa
- III. Em linguagens com tipagem estática, o compilador pode realizar otimizações com base nos tipos conhecidos durante a compilação, enquanto linguagens com tipagem dinâmica não têm essa capacidade.
- IV. Se uma linguagem de programação permite que uma variável armazene qualquer tipo de dado e altere o tipo de dado durante a execução, isso significa que a linguagem não realiza verificação de tipos

Assinale a alternativa correta.

- a) Somente a II está correta
- b) Somente a III está correta
- c) Somente a I e II estão corretas
- d) Somente a IV está correta
- e) Nenhuma afirmativa está correta

Questão 11: Considere o trecho de código em linguagem de programação C a seguir.

```
int myCount = 0;  
  
while (myCount < 10) {  
    printf("%d", myCount+1);  
}  
  
printf("Fim: %d", myCount+1);
```

Assinale a alternativa que apresenta, corretamente, o que esse trecho de código fará ao ser executado.

- a. Escreverá na tela myCount por 10 vezes
- b. Mostrará na tela os valores inteiros de 1 até 10
- c. Entrará em loop infinito
- d. Escreverá na tela 0 por 10 vezes
- e. Mostrará na tela os valores inteiros de 0 até 9

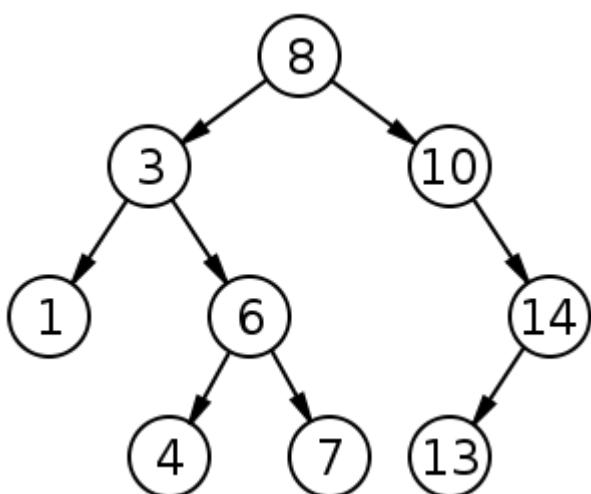
Questão 12: Considere o trecho de código em linguagem de programação C a seguir.

```
void sort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int min_idx = i;
        for (int j = i + 1; j < n; j++) {
            if (arr[j] < arr[min_idx]) {
                min_idx = j;
            }
        }
        int temp = arr[i];
        arr[i] = arr[min_idx];
        arr[min_idx] = temp;
    }
}
```

Assinale a alternativa que apresenta, corretamente, o que esse trecho de código fará ao ser executado.

- a) Ordenará o vetor de entrada, de forma crescente, em ordem quadrática
- b) Ordenará o vetor de entrada, de forma decrescente, em ordem quadrática
- c) Ordenará o vetor de entrada, de forma crescente, em ordem linear
- d) Preencherá o vetor todo com o elemento da primeira posição
- e) Entrará em loop infinito

Questão 13: Considere a estrutura de dados ilustrada abaixo:



Sobre a estrutura:

- a) Trata-se de uma árvore B
- b) Trata-se de uma árvore binária de busca

- c) Trata-se de uma matriz de adjacência
- d) Trata-se de uma lista encadeada
- e) Nenhuma das anteriores

Questão 14: Considere as equações que descrevem a complexidade de tempo dos algoritmos Alg1, Alg2 e Alg3, respectivamente, para entradas de tamanho n:

- $T_1(n) = 100 \cdot n + 15$
- $T_2(n) = 10 \cdot n^2 + 2 \cdot n$
- $T_3(n) = 0,5 \cdot n^3 + n^2 + 3$

A respeito da ordem de complexidade desses algoritmos, pode-se concluir que:

- a. As complexidades assintóticas de Alg1, Alg2 e Alg3 estão, respectivamente, em $O(100)$, $O(10)$ e $O(0,5)$
- b. As complexidades assintóticas de Alg1, Alg2 e Alg3 estão, respectivamente, em $O(n)$, $O(n^2)$ e $O(n^3)$
- c. As complexidades assintóticas de Alg1, Alg2 e Alg3 estão, respectivamente, em $O(n)$, $O(n^2)$ e $O(n^3)$
- d. Alg2 e Alg3 pertencem às mesmas classes de complexidade assintótica
- e. Alg1 e Alg2 pertencem às mesmas classes de complexidade assintótica

Questão 15: Sobre o algoritmo de ordenação MergeSort, descrito no trecho de código a seguir:

```
void merge(int *arr, int lo, int mid, int hi) {
    int i = lo; int j = mid + 1;
    while (i <= mid && j <= hi) {
        if (arr[i] <= arr[j]) {
            i++;
        } else {
            int value = arr[j];
            int index = j;
            while (index != i) {
                arr[index] = arr[index - 1];
                index--;
            }
            arr[i] = value;
            i++; mid++; j++;
        }
    }
}
```

```
void merge_sort(Item *a, Item *aux, int lo, int hi) {  
    if (hi <= lo) return;  
    int mid = lo + (hi - lo) / 2;  
    merge_sort(a, aux, lo, mid);  
    merge_sort(a, aux, mid+1, hi);  
    merge(a, aux, lo, mid, hi);  
}
```

Assinale a alternativa correta sobre este algoritmo de ordenação:

- a. Na implementação acima, a árvore de recursão somente cresce para a parcela esquerda do vetor, uma vez que só a primeira metade é ordenada de forma recursiva
- b. O vetor é ordenado primeiro na parte direita, e depois na parte esquerda, uma vez que a recursão segue este padrão de chamadas para as partes direita e depois esquerda do vetor de entrada
- c. O comportamento do Merge Sort, para qualquer entrada, é quadrático
- d. A implementação apresentada trata-se do Merge Sort que, para qualquer entrada, ordena o vetor em ordem $O(n \log n)$
- e. Por ser recursivo, o comportamento do Merge Sort é sempre exponencial

Questão 16: Sobre o algoritmo apresentado a seguir:

```
int funcao(int x, int n, int v[]){  
    int k;  
    k = n-1;  
    while (k >= 0 && v[k] != x)  
        k -= 1;  
    return k;  
}
```

Assinale a alternativa correta sobre o funcionamento do algoritmo:

- a. Trata-se de uma função que verifica a frequência de valores 0 (zero) na entrada
- b. A função avalia espaços vazios dentro do vetor de entrada
- c. A função preenche o vetor com o valor x passado como parâmetro da função, do fim para o começo
- d. A função remove o elemento x do vetor
- e. A função verifica a ocorrência do valor x, no vetor, varrendo-o do fim para o começo

Questão 17: Sobre o conceito de Programação Orientada a Objetos (POO), analise as afirmativas a seguir.

- I. O encapsulamento é o princípio que permite a exposição irrestrita dos atributos e métodos de um objeto para facilitar a interação entre diferentes objetos dentro de um sistema
- II. A herança é um mecanismo que possibilita a reutilização de código ao permitir que uma classe derive de outra, herdando seus atributos e métodos
- III. O polimorfismo permite que diferentes classes utilizem a mesma interface para operações semelhantes, mas que sejam implementadas de formas distintas, dependendo da classe específica
- IV. A abstração é um processo de ocultar detalhes complexos de implementação, destacando apenas as características essenciais relevantes para o contexto em que o objeto será utilizado

Está correto o que se afirma apenas em

- a. II, III e IV.
- b. II e III.
- c. I, III e IV.
- d. I, II e IV.
- e. II.

Questão 18: Com base no conceito de “encapsulamento” e “polimorfismo” na Programação Orientada a Objetos, assinale a alternativa INCORRETA.

- a. O encapsulamento permite que os atributos de uma classe sejam acessados apenas através de métodos específicos, proporcionando um controle maior sobre como esses atributos são manipulados
- b. O encapsulamento, ao isolar as partes de um programa, dificulta a implementação de novas funcionalidades, já que a modificação de uma parte pode afetar outras partes do código.
- c. O polimorfismo permite que métodos com o mesmo nome, mas implementações diferentes, possam ser chamados em diferentes classes, dependendo do contexto da chamada
- d. O uso de modificadores de acesso restritivos no encapsulamento não permite que atributos de uma classe sejam modificados diretamente de fora da classe, mantendo a integridade do estado do objeto
- e. O padrão de nomenclatura com `get<nomeDoAtributo>` e `set<nomeDoAtributo>` é uma prática comum para acessar e modificar atributos encapsulados de uma classe

Questão 19: Considere os conceitos de herança na Programação Orientada a Objetos (POO). Analise as afirmativas a seguir:

- I. A herança permite que uma classe (chamada de classe derivada ou subclasse) reutilize os atributos e métodos de outra classe (chamada de classe base ou superclasse)
- II. Uma subclasse pode sobrescrever métodos da superclasse, inclusive os métodos definidos como privados na superclasse
- III. É possível que uma subclasse tenha novos atributos e métodos que não existem na superclasse

- IV. Em algumas linguagens, como Java e C++, uma classe pode herdar de várias superclasses diretamente

Qual das alternativas a seguir está correta?

- a. Apenas as afirmativas I e III.
- b. Apenas as afirmativas I, II e IV.
- c. Apenas as afirmativas I e IV.
- d. Apenas a afirmativa I.
- e. Todas as afirmativas estão corretas.

Questão 20: Quais destes algoritmos de ordenação têm a classe de complexidade assintótica, no pior caso, em $O(n \log n)$?

- a. QuickSort, MergeSort, e HeapSort
- b. QuickSort e SelectionSort
- c. QuickSort e BubbleSort
- d. MergeSort e HeapSort
- e. QuickSort, MergeSort e SelectionSort