Universidade Federal do Espírito Santo

Prova de Seleção

Programa de Pós-Graduação em Informática (PPGI)
Processo seletivo de 2026/2

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 caderno de questões e 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 entrega.
- Mantenha silêncio durante toda a prova para não atrapalhar os colegas.
- Qualquer tentativa de comunicação ou comportamento inadequado dentro, ou fora do local de prova poderá resultar na desclassificação.

Boa prova!

Questão 1: Qual das opções a seguir representa o enunciado logicamente equivalente ao enunciado: *"Nem todo artista ganhou algum prêmio".*

a) Alguns artistas ganharam ao menos um prêmio.
b) Ao menos um artista ganhou todos os prêmios.
c) Ao menos um artista não ganhou nenhum prêmio.
d) Nenhum artista ganhou todos os prêmios.
e) Nenhuma das alternativas anteriores.

Questão 2: Considere como verdadeiras as seguintes informações:

- Nenhum estudante gosta de todo professor de Arqueologia.
- Ao menos um estudante gosta de todo doutor em Arqueologia.

O enunciado, que é consequência lógica dos enunciados anteriores, é:

a) Não há professor de Arqueologia que não seja doutor em Arqueologia;
b) Todo professor de Arqueologia é doutor em Arqueologia;
c) Há um professor de Arqueologia que é doutor em Arqueologia;
d) Alguns estudantes gostam de alguns professores de Arqueologia;
e) Há pelo menos um professor de Arqueologia que não é doutor em Arqueologia.

Questão 3: Considere como verdadeiras as seguintes afirmativas:

- Quem canta rock não é estressado;
- Quem é romântico não gosta de filme de terror;
- Quem não gosta de filme de terror é estressado.

Escolha o enunciado que é consequência lógica dos anteriores:

a) Quem gosta de filme de terror é romântico;
b) Quem é romântico não canta rock;
c) Há quem não goste de filme de terror e não seja estressado;
d) Há quem seja romântico e não seja estressado;
e) Nenhuma das alternativas anteriores.

Questão 4: Considere que são simultaneamente verdadeiros os enunciados a seguir:

- $p \rightarrow (q \vee r)$
- $\sim (\sim q \wedge \sim r) \rightarrow s$
- $\sim (\sim p \wedge \sim t)$
- $t \rightarrow u$
- $\sim (u \wedge \sim x)$

O enunciado, que é consequência lógica dos enunciados anteriores, é:

a) $\sim(\sim s \rightarrow t)$
b) $p \wedge \sim s$
c) $\sim(\sim t \vee u)$
d) $s \vee x$
e) Nenhuma das respostas anteriores.

Questão 5: Considere como verdadeiras as premissas a seguir:

- Se Tom atuar na peça, então conhecerá Londres ou não atuará no filme.
- Se Tom não atuar no filme, então não concorrerá ao Oscar de melhor ator.
- Se Tom fretar um jatinho ou terminar o namoro, então não conhecerá Londres.
- Não é verdade que Tom não atuará na peça ou que ele não fretará um jatinho.

Logo, _____.

Determine qual das alternativas a seguir, quando considerada como conclusão das premissas anteriores, forma um argumento dedutivamente válido.

a) Tom não concorrerá ao Oscar de melhor ator e atuará na peça;
b) Tom não fretará um jatinho e não terminará o namoro;
c) Tom conhecerá Londres ou atuará no filme;
d) Tom concorrerá ao Oscar e terminará o namoro;
e) Nenhuma das respostas anteriores.

Questão 6: Considere as afirmativas a seguir sobre estruturas de dados e algoritmos de ordenação:

I. Em uma fila (Queue), o primeiro elemento a entrar é o último a sair.
II. O algoritmo de ordenação Merge Sort é classificado como estável e apresenta complexidade de tempo de ordem O(n log n) no pior caso.
III. Uma Árvore de Busca Binária (BST) pode se degenerar em uma estrutura comparável a uma lista encadeada, passando a ter comportamento linear (O(n)) para operações de busca e inserção.
IV. Em tabelas de dispersão (hash tables), o tratamento de colisões é necessário apenas quando a tabela está cheia e necessita-se inserir mais uma chave.

Assinale a alternativa que contém apenas as afirmativas corretas:

a) Somente as afirmativas I e IV estão corretas.
b) Somente as afirmativas II e III estão corretas.
c) Somente as afirmativas I, II e III estão corretas.
d) Somente as afirmativas II, III e IV estão corretas.
e) Todas as afirmativas estão corretas.

Questão 7: Sobre os conceitos de tipos de dados e verificação de tipos em linguagens de programação, analise as afirmativas a seguir.

I. Em linguagens com tipagem estática, erros de incompatibilidade de tipos podem ser detectados durante a compilação.
II. Em linguagens com tipagem dinâmica, uma variável pode armazenar valores de tipos diferentes em momentos distintos da execução.
III. Em linguagens fortemente tipadas, conversões implícitas entre tipos incompatíveis são sempre permitidas para aumentar a flexibilidade da linguagem.
IV. O conhecimento prévio dos tipos de dados em linguagens com tipagem estática pode permitir que compiladores realizem otimizações durante a compilação.

Assinale a alternativa correta.

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

Questão 8: Sobre estruturas de dados lineares e não lineares, analise as afirmativas a seguir.

I. Em uma lista simplesmente encadeada, a inserção de um novo elemento no início da lista pode ser realizada em tempo constante O(1).
II. Em uma fila implementada por meio de lista encadeada com ponteiros para início e fim, as operações de inserção e remoção podem ser realizadas em tempo constante O(1).
III. É possível implementar as operações de busca, inserção e remoção com complexidade O(log n) em árvores binárias de busca balanceadas que contenham n elementos.
IV. Em uma pilha implementada com vetor, a operação de acesso ao elemento do topo possui complexidade O(1).

Assinale a alternativa correta.

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

Questão 9: Considere o trecho de código em linguagem C a seguir:

```
int *criaVetor() {
    int *v = (int *) malloc(10 * sizeof(int));
    int contador = 0;
    for (int i = 0; i < 10; i++) {
        v[i] = i;
    }
    return v;
}

int main() {
    int *dados;
    dados = criaVetor();
    free(dados);
    return 0;
}
```

Sobre o código apresentado e os conceitos de gerenciamento de memória em C, analise as afirmativas a seguir:

I. A variável contador é armazenada na heap e deixa de existir quando a função criaVetor() termina sua execução.
II. O vetor apontado por v é alocado dinamicamente na heap e continua existindo após o término da função criaVetor().
III. Se a função retornasse o endereço da variável contador em vez do ponteiro v, o programa apresentaria comportamento indefinido.
IV. Após a execução de free(dados), a memória do vetor é liberada, mas a variável dados continua existindo na stack até o final da execução da função main().

Assinale a alternativa correta.

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

Questão 10: Considere o trecho de código em linguagem C a seguir:

```
void* menorPixelImagem(tImagem *img)
{
    void *menor;
    if (img->tipo == INT)
    {
        menor = malloc(sizeof(int));
        *((int*)menor) = 100000000;
        for(int i=0; i<img->colunas*img->linhas; i++)
            if(*((int*)img->dados + i) < *((int*)menor))
                *((int*)menor) = *((int*)img->dados + i);

    }
    else if (img->tipo == FLOAT)
        {
            menor = malloc(sizeof(float));
            *((float*)menor) = 100000000;
            for(int i=0; i<img->colunas*img->linhas; i++)
                if(*((float*)img->dados + i) < *((float*)menor))
                    *((float*)menor) = *((float*)img->dados + i);
        }
    return menor;
}
```

Analise as afirmações a seguir e marque **V** para verdadeiro ou **F** para falso.

I. Quando `img->tipo == INT`, a função aloca memória suficiente para armazenar um valor inteiro.
II. A expressão `img->colunas * img->linhas` representa a quantidade total de elementos (pixels) percorridos pela função.
III. A memória alocada por `malloc()` dentro da função é liberada automaticamente quando a função retorna.
IV. A função percorre `img->linhas * img->colunas` posições em memória, na busca pelo menor elemento encontrado, podendo este ser do tipo INT ou FLOAT.

Assinale a alternativa correta.

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

Questão 11: Considere os trechos de código a seguir, que implementam um tipo abstrato de dado `Array` utilizando ponteiros genéricos e funções de callback:

```
typedef void (*func_ptr_imprimeDado)(void *dado);
typedef void (*func_ptr_liberaDado)(void *dado);

struct Array
{
    void *dado;
    func_ptr_imprimeDado imprimeDado;
    func_ptr_liberaDado liberaDado;
};

void imprimeArray(tArray **a, int tam)
{
    printf("Valores do array: ");
    for(int i=0; i<tam; i++)
    {
        a[i]->imprimeDado(a[i]->dado);
    }
}
```

Analise as afirmações a seguir.

I. O campo `dado` foi declarado como `void *`, permitindo armazenar dados de diferentes tipos por meio de um ponteiro genérico.
II. O parâmetro `tArray **a` indica que a função recebe um vetor cujos elementos são ponteiros para estruturas do tipo `tArray`.
III. O ponteiro para função `liberaDado` permite que cada tipo de dado armazenado possua sua própria estratégia de liberação de memória, tornando a estrutura mais genérica e reutilizável.

Assinale a alternativa correta.

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

Questão 12: Sobre o conceito de polimorfismo na Programação Orientada a Objetos (POO), analise as afirmativas a seguir.

I. O polimorfismo permite que objetos de diferentes classes respondam de maneiras distintas à mesma chamada de método.
II. O polimorfismo possibilita que uma referência de uma classe base aponte para objetos de suas subclasses.
III. A sobrescrita (override) de métodos é um mecanismo frequentemente utilizado para implementar polimorfismo.

IV. O uso de polimorfismo elimina a necessidade de herança, uma vez que qualquer classe pode assumir o comportamento de qualquer outra classe.

Assinale a alternativa correta.

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

Questão 13: Considere a função a seguir:

```
int func(int n) {
    if (n <= 1)
        return 1;

    return func(n - 1) + func(n - 1);
}
```

Sobre o funcionamento e a complexidade de tempo do código implementado na função, analise as afirmativas a seguir.

I. Para cada chamada com `n > 1`, a função realiza duas chamadas recursivas para `func(n - 1)`.
II. O número de chamadas recursivas cresce exponencialmente com o valor de `n`.
III. A complexidade de tempo do algoritmo é `O(n)`, pois o parâmetro é reduzido em uma unidade a cada chamada recursiva.
IV. A árvore de recursão gerada pelo algoritmo possui aproximadamente 2^n nós.

Assinale a alternativa correta.

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

Questão 14: Sobre os algoritmos de busca em grafos BFS (Busca em Largura) e DFS (Busca em Profundidade), analise as afirmativas a seguir.

I. O algoritmo BFS utiliza uma pilha para controlar os vértices que serão visitados.
II. O algoritmo DFS sempre encontra o menor caminho entre dois vértices de um grafo não ponderado.

III. Em um grafo conexo, BFS e DFS visitam necessariamente os vértices na mesma ordem.
IV. O algoritmo DFS explora inicialmente todos os vértices vizinhos ao vértice de origem antes de avançar para níveis mais profundos.

Assinale a alternativa correta.

a) Apenas as afirmativas I e II estão corretas.
b) Apenas as afirmativas II e III estão corretas.
c) Apenas as afirmativas III e IV estão corretas.
d) Apenas as afirmativas I, II e IV estão corretas.
e) Nenhuma das afirmativas está correta.

Questão 15: Considere o trecho de código em linguagem C a seguir.

```
void main() {
    int x = 10;
    int y = 20;
    int *p;
    p = &x;
    *p = *p + 5;
    p = &y;
    *p = *p - 3;
    printf("%d %d\n", x, y);
}
```

Assinale a alternativa que apresenta corretamente a saída produzida pelo programa.

a) 10 20
b) 15 20
c) 10 17
d) 15 17
e) 25 17

Questão 16: Considere o seguinte trecho de pseudocódigo:

```
Classe Animal:
    método emitirSom():
        exibir("Som genérico")


Classe Cachorro herda de Animal:
    método emitirSom():
        exibir("Au Au")


Programa Principal:
Animal a = novo Cachorro()
a.emitirSom()
```

Sobre o programa apresentado e os conceitos de Programação Orientada a Objetos, analise as afirmativas a seguir.

I. A classe Cachorro herda os atributos e métodos da classe Animal.
II. A variável a pode referenciar um objeto da classe Cachorro porque Cachorro é uma subclasse de Animal.
III. Em linguagens que implementam polimorfismo dinâmico (late binding), A execução de `a.emitirSom()` exibirá a mensagem "Som genérico", pois a variável a foi declarada como sendo do tipo Animal.

Assinale a alternativa correta.

a) Apenas as afirmativas I e II estão corretas.
b) Apenas as afirmativas II e III estão corretas.
c) Apenas a afirmativa I está correta.
d) Apenas as afirmativas I e III estão corretas.
e) Todas as afirmativas estão corretas.

Questão 17: Considere as afirmativas a seguir sobre tabelas hash e operações de busca.

I. Em uma tabela hash com uma boa função de dispersão e fator de carga controlado, a complexidade média da operação de busca é O(1).
II. No pior caso, quando muitas colisões ocorrem, a busca em uma tabela hash pode apresentar complexidade O(n).
III. O uso de tratamento de colisões elimina completamente a possibilidade de degradação do desempenho da busca.

Assinale a alternativa correta.

a) Apenas a afirmativa I está correta.
b) Apenas as afirmativas II e III estão corretas.
c) Apenas as afirmativas I e III estão corretas.
d) Apenas as afirmativas I e II estão corretas.
e) Todas as afirmativas estão corretas.

Questão 18: Considere o trecho de código em linguagem C a seguir. Suponha que existam uma pilha P e uma fila F independentes. As funções push() e pop() implementam, respectivamente, as operações de inserção e remoção da pilha P, enquanto enqueue() e dequeue() implementam as operações de inserção e remoção da fila F.

```
Stack P;
Queue F;

push(P, 10);
push(P, 20);
push(P, 30);

enqueue(F, 10);
enqueue(F, 20);
enqueue(F, 30);

int a1 = pop(P);
int b1 = dequeue(F);
int a2 = pop(P);
int b2 = dequeue(F);
```

Sobre a execução do código, analise as afirmativas a seguir.

I. Ao final da execução, a variável `a1` armazenará o valor 30.
II. Ao final da execução, a variável `b1` armazenará o valor 10.
III. Ao final da execução, as variáveis `a2` e `b2` armazenam o mesmo valor.
IV. Ao final da execução, as estruturas pilha e fila conterão os mesmos elementos.

Assinale a alternativa correta.

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

Questão 19: Sobre as instruções de repetição e controle de fluxo em linguagens de programação, analise as afirmativas a seguir.

I. Em uma estrutura `while`, a condição de parada é avaliada antes da execução do corpo do laço.
II. Em uma estrutura `do...while`, o corpo do laço é executado pelo menos uma vez, independentemente do valor inicial da condição.
III. A instrução `break`, quando executada dentro de um laço de repetição, interrompe imediatamente a execução do laço.
IV. Em uma estrutura `for`, a condição de repetição é sempre avaliada após a execução do corpo do laço.

Assinale a alternativa correta.

a) Apenas as afirmativas I e II estão corretas.
b) Apenas as afirmativas II, III e IV estão corretas.
c) Apenas as afirmativas I, II e III estão corretas.

d) Apenas as afirmativas I e IV estão corretas.
e) Todas as afirmativas estão corretas.

Questão 20: O Problema do Caixeiro Viajante (*Travelling Salesman Problem* - TSP) consiste em determinar uma rota de menor custo que visite um conjunto de cidades exatamente uma vez e retorne à cidade de origem. Uma instância do problema pode ser representada por um grafo ponderado, em que os vértices representam cidades e as arestas representam os custos de deslocamento entre pares de cidades.

Sobre o Problema do Caixeiro Viajante, analise as afirmativas a seguir.

I. Uma solução válida para o problema corresponde a um ciclo que visita todos os vértices exatamente uma vez antes de retornar ao vértice inicial.
II. Em uma abordagem de força bruta, o número de rotas possíveis cresce factorialmente com o número de cidades.
III. O algoritmo de Dijkstra pode ser utilizado para encontrar o menor caminho entre dois vértices de um grafo ponderado, mas não resolve o Problema do Caixeiro Viajante de forma ótima.
IV. A complexidade de tempo de um algoritmo de força bruta para o TSP é $O(n^2)$, uma vez que é necessário verificar apenas as conexões entre pares de cidades.

Assinale a alternativa correta.

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