CONCURSO ITA 2025

EDITAL: 02/ITA/2025

CARGO: PESQUISADOR

PERFIL: PQ-11

CADERNO DE QUESTÕES

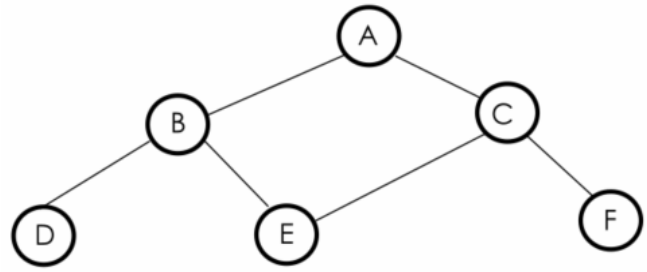
1. Esta prova tem duração de **4 (quatro) horas**.
2. Você poderá usar **apenas** caneta esferográfica de corpo transparente com tinta preta, lápis ou lapiseira, borracha, régua transparente simples e compasso. **É proibido portar qualquer outro material escolar ou equipamento eletrônico.**
3. Você recebeu este **caderno de questões e um caderno de respostas** que deverão ser devolvidos ao final do exame.
4. O caderno de questões é composto por **7 questões dissertativas**.
5. As **questões dissertativas devem ser respondidas exclusivamente no caderno de respostas**. Responda sequencialmente as questões, usando caneta preta.
6. **É obrigatória a devolução do caderno de questões e do caderno de respostas**, sob pena de desclassificação do candidato.
7. **Aguarde o aviso para iniciar a prova. Ao terminá-la, avise o fiscal e aguarde-o no seu lugar**.

Questão 1. Considere a seguinte recorrência, que modela o tempo de execução de um algoritmo de divisão e conquista que requer tempo O(n) para combinar os resultados de dois subproblemas de tamanho n/2:

$$T(n) = 2T\left(\frac{n}{2}\right) + n$$

Determine a **ordem de complexidade assintótica** de T(n). Sugere-se utilizar algum teorema conhecido para análise de complexidade de algoritmos.

Questão 2. Para o seguinte grafo não direcionado, considerando que os nós adjacentes são explorados na ordem alfabética:



a) Qual seria a ordem de visita dos vértices se você utilizasse **busca em largura** (breadth-first search - BFS) partindo do vértice A?
b) Qual seria a ordem de visita dos vértices se você utilizasse **busca em profundidade** (depth-first search - DFS) partindo do vértice A?
c) Quais estruturas de dados auxiliares são utilizadas respectivamente na execução das buscas BFS e DFS? Justifique sua resposta.

Questão 3. Como pesquisador, você deve convencer sua equipe a utilizar estruturas de **árvore binária de busca (ABB)** que sejam balanceadas.

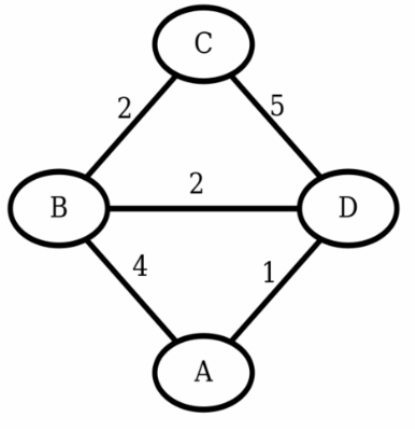
a) Explique o impacto do **desequilíbrio** em uma ABB sobre a complexidade da busca.
b) Escolha um tipo de ABB balanceada (por exemplo, AVL ou rubro-negra) e descreva qual é a propriedade que essa árvore procura manter após cada operação de inserção ou remoção de elemento a fim de garantir o seu balanceamento.

Questão 4. Sobre conceito e aplicação de grafos e árvores:

a) Defina o que é um grafo e explique em que condições um grafo **é** uma árvore.

b) Dê um exemplo de aplicação prática em que uma **árvore** é adequada e outro em que um **grafo (não árvore)** é mais apropriado, justificando o motivo em cada caso.

Questão 5 Considere o grafo ponderado não direcionado abaixo (com pesos positivos):



Aplique o **algoritmo de Dijkstra** a partir do vértice **A**. Preencha a tabela abaixo passo a passo, indicando em cada iteração:

- o **vértice selecionado**,
- as **distâncias atuais** até cada vértice,
- e os **vértices já visitados**.

.

Iteração	Vértice selecionado	Distâncias (A,B,C,D)	Visitados
Inicial	—	(0, ∞, ∞, ∞)	**Ø**
1			
2			
3			
4			

Questão 6 Considere o seguinte algoritmo recursivo para calcular o máximo divisor comum (MDC) entre dois números inteiros não negativos:

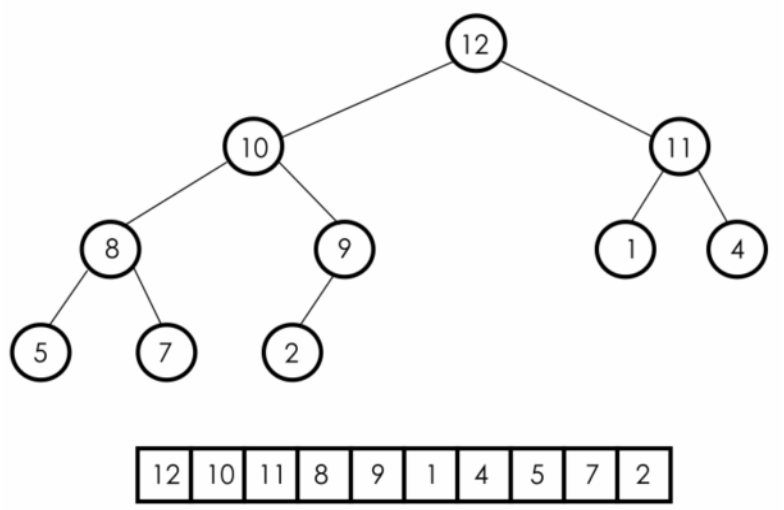
```
ALGORITMO MDC(a, b)
    SE b = 0 ENTÃO
        RETORNE a
    SENÃO
        RETORNE MDC(b, a MOD b) // MOD = resto da divisão
    FIM SE
FIM ALGORITMO
```

a) Simule passo a passo a execução de MDC(48,18) indicando os valores de a e b em cada chamada recursiva até o retorno do resultado final.

b) Reescreva o algoritmo de forma iterativa, ou seja, com uma malha de repetição (for ou while, por exemplo) e sem a chamada recursiva.

Questão 7 Um heap binário é uma aplicação da estrutura de árvore, sendo que cada nó corresponde a um elemento que segue uma regra específica:

- A propriedade de heap máximo (max-heap) especifica que um nó filho (no código calculado pelas funções esquerda e direita) tem sempre armazenado um valor menor ou igual ao seu pai.



A heap pode ser organizada na forma de um vetor como ilustrado na figura.Essa estrutura de dados possibilita a consulta ou extração de forma eficiente do maior elemento.

Considerando a implementação a seguir, heapificar é um procedimento auxiliar para reorganizar o vetor (garantindo a propriedade de heap máximo em uma determinada posição do vetor) e construirHeap é um procedimento que usa heapificar para reorganizar todas as posições do vetor (garantindo a propriedade de heap máximo para todos os elementos).

```
função esquerda(i): retorne (2 * i + 1)
função direita(i): retorne (2 * i + 2)

/* a - vetor, n - número de elementos, i - posição do elemento que deve respeitar a
propriedade de heap */
procedimento heapificar(a, n, i):
    esq ← esquerda(i)
    dir ← direita(i)
    max ← i

    se (esq < n) e (a[esq] > a[i]) então
        max ← esq

    se (dir < n) e (a[dir] > a[max]) então
        max ← dir

    se (max ≠ i) então
        aux ← a[i]
        a[i] ← a[max]
        a[max] ← aux
        heapificar(a, n, max)

/* a - vetor, n - número de elementos */
procedimento construirHeap(a, n)
    para i de ((n - 1) // 2) até 0 passo -1 faça
        heapificar(a, n, i)
```

CONTINUA →

Continuação - Questão 7

De acordo com as informações apresentadas, faça o que se pede nos itens a seguir.

a) Como ficará o vetor a = [2, 5, 8 ,13, 21, 1, 3, 34] após a execução do procedimento construirHeap(a, 8).

b) Apresente a complexidade de tempo no pior caso para o procedimento heapificar, justifique.

RASCUNHO

RASCUNHO

RASCUNHO

RASCUNHO

RASCUNHO

RASCUNHO

RASCUNHO

RASCUNHO

RASCUNHO