

**ATIVIDADE 2 - ESOFT - ESTRUTURA DE DADOS II - 52/2023****Período:**29/05/2023 08:00 a 16/06/2023 23:59 (Horário de Brasília)**Status:**ENCERRADO**Nota máxima:**0,50**Gabarito:**Gabarito será liberado no dia 17/06/2023 00:00 (Horário de Brasília)**Nota obtida:**0,40**1ª QUESTÃO**

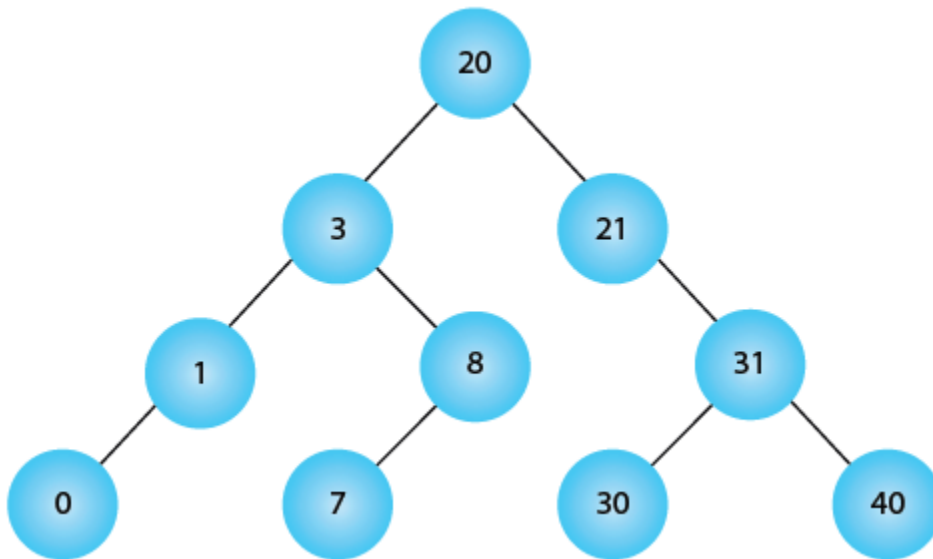
Uma árvore binária é uma estrutura que ou está vazia ou é formada por três conjuntos disjuntos: raiz, sub-árvore esquerda e sub-árvore direita. Cada uma dessas sub-árvores também é uma árvore binária e segue as mesmas regras da árvore principal.

OLIVEIRA, Pietro Martins de; LEON, Rogério de. Estrutura de Dados II. Maringá-PR, Unicesumar, 2019.

Para acessar os dados de uma árvore como essa, existem algoritmos de caminhamento como o que você pode ver a seguir:

```
void bar(NO* r){  
    if(r){  
        bar(r->esq);  
        bar(r->dir);  
        printf("%d \t", r->dado); //visita o nó atual  
    }  
}
```

A função bar representa o algoritmo de caminhamento pós-ordem para árvores binárias. Assim sendo, observe a árvore a seguir:



Considere a execução da invocação da função bar(), passando como parâmetro a referência para o nó raiz com valor igual a 20 (árvore ilustrada acima). Qual alternativa representa corretamente a mensagem que seria impressa na tela do usuário?

## ALTERNATIVAS

- ☐ 20, 3, 21, 1, 8, 31, 0, 7, 30, 40
- ☒ 0, 1, 7, 8, 3, 30, 40, 31, 21, 20
- ☐ 20, 3, 1, 0, 8, 7, 21, 31, 30, 40
- ☐ 0, 1, 3, 7, 8, 20, 30, 31, 40, 21
- ☐ 20, 21, 31, 40, 30, 3, 8, 7, 1, 0

**Atenção!** Questão anulada.

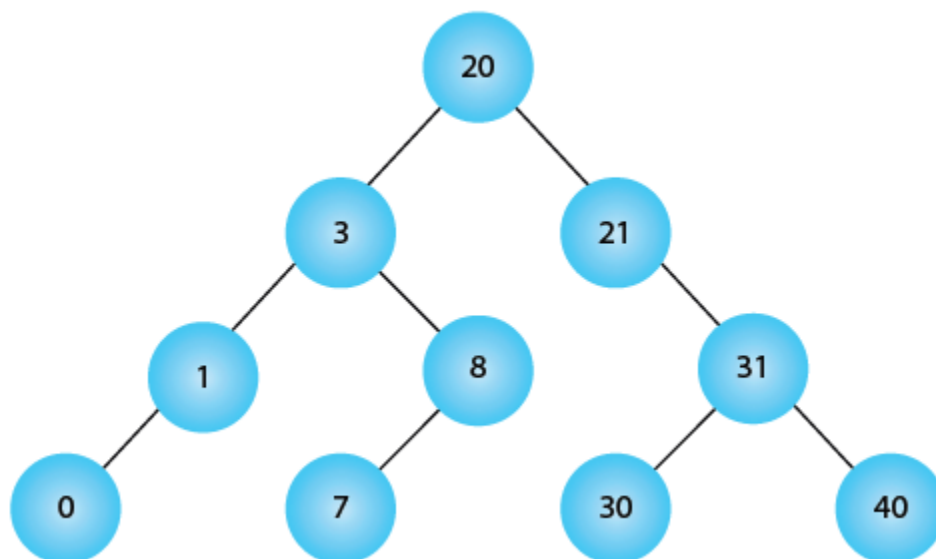
## ALTERNATIVAS

- ☐ I, apenas.
- ☐ I e II, apenas.
- ☐ II e III, apenas.
- ☐ II e III, apenas.
- ☐ I, II e III.

## 3ª QUESTÃO

Para resolver problemas de maneira eficiente, pode-se tentar eliminar soluções inviáveis, encurtando a resolução. A busca binária e a busca em árvore binária são dois exemplos de algoritmos que "podam" soluções errôneas, para encurtar o caminho até uma solução.

OLIVEIRA, Pietro Martins de; LEON, Rogério de. Estrutura de Dados II. Maringá-PR, Unicesumar, 2019.



Considerando uma busca em árvore binária ordenada que se inicie em 20, caso o usuário decida por buscar o valor alvo igual a 25, qual seria a ordem de visitação dos nós?

## ALTERNATIVAS

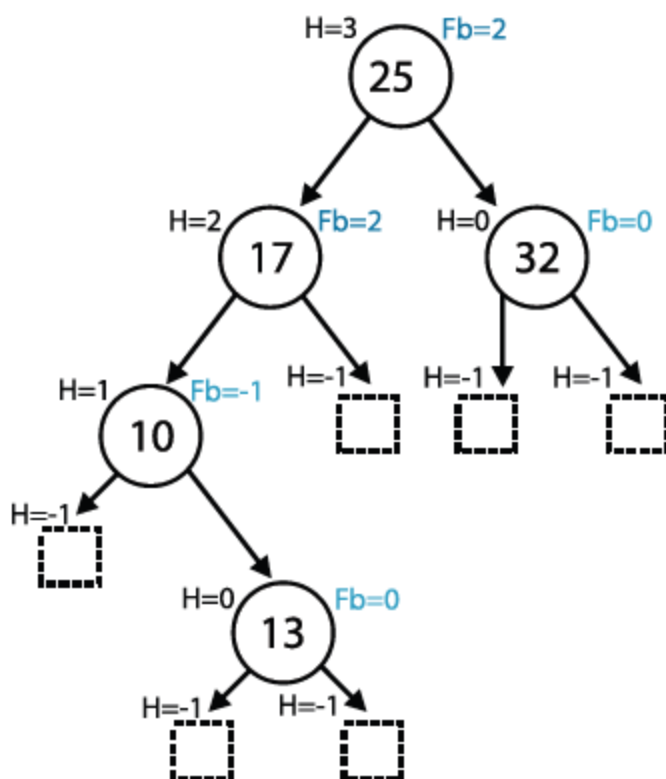
- ☐ 20, 21, 31, 30
- ☐ 20, 21, 31
- ☐ 20, 3, 8, 7
- ☒ 20, 21, 31, 40
- ☐ 20, 21

## 4ª QUESTÃO

Para resolver o problema do desbalanceamento de árvores binárias de busca, os pesquisadores Adelson-Velskii e Landis, em 1962, criaram um algoritmo que leva as iniciais de seus nomes. As árvores AVL são, nesse sentido, árvores nas quais todos os nós encontram-se balanceados.

OLIVEIRA, Pietro Martins de; LEON, Rogério de. Estrutura de Dados II. Maringá-PR, Unicesumar, 2019.

Observe a árvore binária AVL a seguir:



Com base na árvore representada na ilustração acima, avalie as afirmações que se seguem:

- I. Pode-se dizer que a altura do nó 25 é igual a 2.
- II. Não podemos dizer que é uma árvore estritamente binária.
- III. Para balancear essa árvore, é preciso realizar rotação dupla.

É correto o que se afirma em:

**ALTERNATIVAS**

- ☐ I, apenas.
- ☐ I e II, apenas.
- ☐ I e III, apenas.
- ☒ II e III, apenas.
- ☐ I, II e III.

**5ª QUESTÃO**

Uma das características mais importantes de árvores binárias é que elas podem ser utilizadas para servir de base para algoritmos que percorrem seus nós para executar funções como, por exemplo, algoritmos de busca e caminhamento.

(Fonte: OLIVEIRA, Pietro Martins de; LEON, Rogério de. Estrutura de Dados II. Maringá-PR, Unicesumar, 2019.)

```
void foo(struct NO* r){  
    if(r){  
        printf("%d \t", r->val); //visita o nó atual  
        foo(r->esquerda);  
        foo(r->direita);  
    }  
}
```

Assinale a alternativa que corresponde ao nome correto do algoritmo representado no código-fonte acima.

**ALTERNATIVAS**

- ☐ Busca em largura.
- ☐ Busca em profundidade.
- ☐ Caminhamento em-ordem.
- ☐ Caminhamento pós-ordem.
- ☒ Caminhamento pré-ordem.