

Disciplina: Estrutura de Dados I

Modalidade: Presencial

Professor: André Lucio de Oliveira

TRABALHO TEÓRICO (A1)

Data de Entrega: **06/10/2022**

Possibilidade de fazer a atividade em trio

Pontuação máxima: 10,0 (dez)

Forma de entrega: por e-mail (andre.oliveira@uva.br).

OBS: O campo "Assunto" do e-mail deve ter o seguinte padrão:

- ED1-TRABALHO_TEÓRICO_A1_NOME_INTEGRANTES_GRUPO

Especificação da atividade: Este trabalho consiste na criação de questões objetivas que abordem assuntos já vistos em sala de aula, tais como: Vetores dinâmicos, structs, vetores de structs, ponteiros, recursividade, etc. **Vocês devem enviar a questão + o gabarito.**

- Para quem está fazendo individual: 1 questão de análise de código.
- Para quem está fazendo dupla ou trio: 1 questão de análise de código + 1 questão teórica.

OBS-1: questões devem seguir o padrão de avaliação do [ENADE \(Exame Nacional de Desempenho dos Estudantes\)](#).

OBS-2: A questão de análise de código pode seguir a seguinte ideia:

Questão 1 - Em Ciência da computação, uma árvore binária de busca (ou árvore binária de pesquisa) é uma estrutura de dados de árvore binária baseada em nós, onde todos os nós da subárvore esquerda possuem um valor numérico inferior ao nó raiz e todos os nós da subárvore direita possuem um valor superior ao nó raiz (esta é a forma padrão, podendo as subárvores serem invertidas, dependendo da aplicação). O objetivo desta árvore é estruturar os dados de forma a permitir busca binária. O trecho de código a seguir foi extraído de um programa que implementa as funcionalidades de uma árvore binária de busca.

```

6  ...
7  typedef struct arvore{
8      int num;
9      struct arvore* sad;
10     struct arvore* sae;
11 } TABB;

27 ...
28 /* Função que cria o nó da árvore */
29 TABB* cria(int num, TABB* sae, TABB* sad){
30     TABB* a = (TABB*) malloc(sizeof(TABB));
31     a->num = num;
32     a->sae = sae;
33     a->sad = sad;
34     return a;
35 }
36 /* Função que insere um dado na árvore */
37 TABB* inserir_arv(TABB* t, int x){
38     if(!t) return cria(x, NULL, NULL);
39     if(t->num < x)
40         t->sad = inserir_arv(t->sad, x);
41     else if(t->num > x)
42         t->sae = inserir_arv(t->sae, x);
43     return t;
44 }
45 ...

```

Analisando este código é INCORRETO afirmar que:

- a) A linha 38 trata a inserção de um nó folha, sendo a condição de parada da árvore de recursão.
- b) As linhas 40 e 42 tratam a “descida” na árvore de recursão, com o objetivo de descobrir o local correto de inserção na árvore binária de busca. Na linha 40 descemos para à direita e na linha 42 descemos à esquerda.
- c) As linhas 7 a 11 definem um tipo redefinido como TABB, o qual representa a estrutura de um nó na árvore binária de busca, onde “sae” e “sad” são os ponteiros para as subárvores esquerda e direita, respectivamente.
- d) A linha 43 retorna sempre o ponteiro para raiz principal da árvore, isto é, aquela que hierarquicamente é o primeiro nó desta árvore (mais ao topo).
- e) Na linha 38, “!t” significa que a árvore de recursão encontrou uma subárvore vazia, isto é, apontando para NULL.

OBS-3: A questão teórica pode seguir a ideia:

Questão 2 - Árvore, no contexto da programação, engenharia de software e ciência da computação, é uma das mais importantes estruturas de dados não lineares. Este tipo de estrutura herda as características das topologia em árvore.

Em uma definição mais formal, uma árvore é uma estrutura que contém um conjunto finito de um ou mais nós, sendo que um dos nós é especialmente designado como o nó raiz e os demais nós são particionados em 0 ou mais conjuntos disjuntos onde cada um desses conjuntos é em si uma árvore, que recebe o nome de sub-árvore.

Sobre a estrutura de dados "Árvore", analise seguintes as afirmativas:

I. A árvore é uma estrutura linear que permite representar uma relação de hierarquia. Ela possui um nó raiz e sub-árvores não vazias.

II. Na árvore binária de busca, um dos percursos permite que os nós sejam obtidos de forma ordenada.

III. O processo de balanceamento (estático ou dinâmico) otimiza a busca em árvores binárias, minimizando sua altura.

IV. Uma árvore-B não pode ser usada para armazenamento de dados em disco, pois necessita de um número maior de nós (maior altura) quando comparada a uma árvore binária.

Está correto o que se afirma em:

- a) Todas as asserções são proposições verdadeiras.
- b) Apenas as asserções II e III são proposições verdadeiras.
- c) As asserções I e II são proposições verdadeiras, e as demais são falsas.
- d) As asserções I e II são proposições falsas e a III e IV são verdadeiras.
- e) Apenas a asserção II é uma proposição verdadeira.

OBS-4: Vocês estão liberados para adicionar imagens, tabelas, fluxos, etc nas questões.

ATENÇÃO: Não se limitem aos formatos de questões apresentados acima. Vocês podem criar questões com diferentes formatos, exemplo: aqueles que colocam duas sentenças separadas por um "PORQUE", análise de asserções V/F, etc.