

ALGORITMOS E ESTRUTURA DE DADOS AVANÇADO TRABALHO PRÁTICO 01

Orientações para o a atividade avaliativa:

- A atividade pode ser realizada individualmente ou em dupla (2 pessoas);
- Nota: valor máximo de 500 pontos;
- Data de entrega: 27/10/2024;
- O trabalho deverá ser enviado para o e-mail noiza.waltrick@gmail.com.
- Assunto do e-mail: [EDA] Trabalho Prático 01
- Data de apresentação: A partir de 29/10/2024.

O que deve ser entregue:

- Um projeto com as funções e implementações seguindo a descrição das funções especificadas neste trabalho.

Comentários Gerais:

- Os grupos terão que fazer uma entrevista sobre o trabalho. O participante ou os participantes do grupo deverão saber responder a TODAS as perguntas feitas.
- A nota do trabalho será obtida somente mediante entrevista.
- Alunos que não enviarem o trabalho no prazo estipulado não poderão realizar a entrevista.
- Alunos que enviarem o trabalho, mas não estiverem presentes na data da entrevista terão nota zero.
- Trabalhos copiados serão penalizados com nota zero.

DESCRIÇÃO

Trabalho de Implementação de Funções para Árvores Binárias

Objetivo:

Neste trabalho, vocês deverão implementar quatro funções para manipulação e consulta de uma **árvore binária de busca** (ABB). O foco será em operações de consulta e percursos (em diferentes ordens) em uma árvore binária, além da implementação de uma função que imprime os elementos da árvore por nível.

Funções a serem implementadas:

1. Função: `int consulta_ArvBin(ArvBin *raiz, int valor);`

Objetivo: Esta função deve procurar um valor na árvore binária de busca.

Descrição: A função recebe como parâmetros o ponteiro para a raiz da árvore e o valor que será buscado. Ela deve percorrer a árvore de maneira eficiente (de acordo com as propriedades da ABB) e retornar 1 se o valor for encontrado ou 0 caso o valor não exista na árvore.

2. Função: void preOrdem_ArvBin(ArvBin *raiz);

Objetivo: Esta função deve realizar o percurso **pré-ordem** na árvore e imprimir os valores dos nós.

Descrição: No percurso pré-ordem, a visitação dos nós segue a ordem:

1. Visita o nó atual (raiz da subárvore).
2. Percorre a subárvore esquerda.
3. Percorre a subárvore direita.

3. Função: void posOrdem_ArvBin(ArvBin *raiz);

Objetivo: Esta função deve realizar o percurso **pós-ordem** na árvore e imprimir os valores dos nós.

Descrição: No percurso pós-ordem, a visitação dos nós segue a ordem:

1. Percorre a subárvore esquerda.
2. Percorre a subárvore direita.
3. Visita o nó atual (raiz da subárvore).

4. Função: void imprimePorNivel_ArvBin(ArvBin *raiz);

Objetivo: Esta função deve imprimir os valores dos nós **por nível** na árvore binária.

Descrição: A ideia é visitar a árvore nível por nível, começando pela raiz. Este é um exemplo de percurso em **largura** (ou *breadth-first*).

A função deve imprimir os nós da árvore organizados por nível. Por exemplo, primeiro o nó raiz, depois os filhos da raiz, depois os netos, e assim por diante.

Dicas:

- Para implementar esse percurso, considere utilizar uma fila (ou estrutura similar) para armazenar temporariamente os nós enquanto percorre a árvore.
- Primeiro, insira a raiz na fila, depois retire a raiz e insira seus filhos (esquerdo e direito), e continue esse processo até que todos os nós sejam visitados.

Instruções:

- Cada uma dessas funções deve ser implementada de maneira independente e testada.
- Vocês devem garantir que a árvore binária está devidamente estruturada para permitir o funcionamento correto das funções.