

Lista de exercícios-ED-06 Árvores Binárias

Estes exercícios devem ser entregues no Google Classroom. O objetivo desta lista é implementar as principais operações em árvores binárias na linguagem C.

Questões:

1. **Implementação da árvore.** Inicialmente crie os seguintes arquivos fonte:
 - **principal.c:** função main
 - **arv.h:** declarações de estruturas, tipo e funções da pilha
 - **arv.c:** implementação das funções da pilha
2. **Estruturas e tipos de dados.** Iremos representar um nó da árvore binária por meio de uma estrutura em C contendo a informação armazenada no nó e dois ponteiros para sub-árvores à esquerda e à direita conforme o código apresentado abaixo. Neste exercício, cada nó da árvore armazena um dado do tipo caracter:

```
struct arv {  
    char info;  
    struct arv* esq;  
    struct arv* dir;  
};  
  
typedef struct arv Arv;
```

3. **Interface do tipo abstrato Árvore Binária.** A seguir temos os protótipos das funções a serem implementadas. Os protótipos devem estar declarados no arquivo `art.h`, e as respectivas implementações devem estar no arquivo `arv.c`:

Função	Descrição
<code>Arv* arv_criavazia(void);</code>	instancia uma nova árvore binária vazia. Esta função simplesmente deve retornar <code>NULL</code>
<code>Arv* arv_cria(char c, Arv* e, Arv* d);</code>	instancia uma nova árvore. A função recebe ponteiros para as suas duas sub-árvores e a informação (char) a ser armazenada no nó raiz. A função deve retornar o endereço do nov nó raiz
<code>Arv* arv_libera(Arv* a);</code>	libera toda a memória ocupada pela árvore (em outras palavras, “apaga” a árvore
<code>int arv_vazia(Arv* a);</code>	retorna 1 ou 0 caso a árvore esteja vazia ou não, respectivamente
<code>int arv_pertence(Arv* a, char c);</code>	retorna 1 caso o caracter c estiver contido na árvore, e retorna 0 caso contrário
<code>void arv_imprime_pre(Arv* a);</code>	imprime a árvore em saída de texto em pré-ordem

Função	Descrição
<code>void arv_imprime_sim(Arv* a);</code>	imprime a árvore em saída de texto em ordem simétrica
<code>void arv_imprime_pos(Arv* a);</code>	imprime a árvore em saída de texto em pós-ordem
<code>int arv_folhas(Arv* a);</code>	retorna a quantidade de nós folha da árvore
<code>int arv_altura(Arv* a);</code>	retorna a altura da árvore (0 para árvore vazia)
<code>int arv_igual(Arv* a1, Arv* a2);</code>	retorna 1 se as árvores a1 e a2 são iguais. Retorna 0 caso contrário. As duas árvores são iguais se seus nós armazenarem informações iguais nas mesmas posições
<code>Arv* arv_copia(Arv* a);</code>	retorna uma cópia da árvore a. A nova árvore deve ter novos nós instanciados

4. **Demonstração do funcionamento da estrutura.** No arquivo principal.c, implemente uma demonstração de todas as funções implementadas.