



## Algoritmos e Estruturas de Dados II

### Exercícios Complementares - Árvores de Busca: revisão

Pense num algoritmo e elabore um programa em **C** para cada um dos problemas a seguir. Na resolução de cada exercício edite o programa, compile, execute e **teste suas soluções**.

1. Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária de busca e retorna o menor valor armazenado na árvore. Caso a árvore seja vazia, a função deve retornar NULL. O protótipo da função é:  
`Arv* menor_valor(Arv* a);`
2. Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária de busca e um valor  $x$ , e retorna o número de nós que armazenam valores maiores do que  $x$ . O protótipo da função é:  
`int maiores_que_x(Arv* a, int x);`
3. Considere árvores binárias de busca que armazenam valores inteiros, de forma que, para qualquer nó, o valor a ele associado é sempre maior que aqueles associados aos nós da sub-árvore da esquerda e menor ou igual àqueles associados aos nós da sub-árvore da direita, e cada nó é do tipo `NoArv`, descrito abaixo:
4. Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária de busca que armazena apenas valores positivos e retorna o maior valor armazenado na árvore. Caso a árvore seja vazia, a função deve retornar -1. O protótipo da função é: `int maior_valor(Arv* a);`
5. Escreva uma função que recebe como parâmetro o ponteiro para a raiz de uma árvore binária e retorna 1, se esta for uma árvore binária de busca, ou 0, caso contrário. O protótipo da função é: `int verifica_abb(Arv* a);`
6. Duas árvores binárias são similares se elas são vazias ou se ambas armazenam o mesmo valor em seu nó raiz, suas sub-árvores da esquerda são similares e suas sub-árvores da direita também são similares. Escreva uma função para determinar se duas árvores binárias são similares. Protótipo: `int similares(Arv* a1, Arv* a2);`
7. Escreva uma função recursiva que recebe como parâmetros o ponteiro para o primeiro elemento de um vetor de inteiros ordenado e seu tamanho  $n$  (onde  $n = 2m - 1$  para  $m > 0$ ) e retorne o ponteiro para uma árvore binária de busca completa, construída a partir dos valores contidos no vetor. A função deve ser executada em tempo linear, ou seja, deve ter complexidade  $O(n)$ . O protótipo da função é:  
`Arv* cria_arv(int* v, int n);`



8. Escreva uma função em C que, dada uma árvore binária de busca qualquer, retire todos os elementos ímpares da árvore original.
9. Escreva uma função em C que, dada uma árvore binária de busca qualquer, retorne, num vetor, todos os elementos menores que N.