

LISTA EXERCÍCIOS SOBRE ÁRVORES

Para os exercícios abaixo, monte dois programas: um para árvores não binárias e outro para árvores binárias com as funções dadas em sala de aula – REPRESENTAÇÃO ALOCAÇÃO DINÂMICA.

No material de aula, você encontrará funções básicas para construir uma árvore exemplo e testar as funções dos exercícios abaixo.

ÁRVORES GERAIS (NÃO BINÁRIAS):

1. Considere uma **árvore qualquer**, cujo nó raiz está armazenado na variável **raiz**. Construir uma função que recebe o endereço do nó raiz e conta quantos nós a árvore possui. Retorna pelo return número de nós.
2. Considere uma árvore qualquer, cujo nó raiz está armazenado na variável **raiz**. Construir uma função que verifica quantos nós folha a árvore possui. O total deve retornar pelo return.
3. Considere uma árvore qualquer, cujo nó raiz está armazenado na variável **raiz**. Construir uma função que recebe o endereço do nó raiz e conta quantos nós dessa árvore possui apenas um descendente. Retorna pelo return número de nós.
4. Considere uma árvore qualquer, cujo nó raiz está armazenado na variável **raiz**. Construir uma função que recebe o endereço do nó raiz e conta quantos nós dessa árvore possui todos os descendentes. Isto é, nenhum dos seus links é NULL. Retorna pelo return número de nós.

ÁRVORES BINÁRIAS :

1. Considere uma **árvore binária qualquer**, cujo nó raiz está armazenado na variável **raiz**. Construir uma função que recebe o endereço do nó raiz e conta quantos nós a árvore possui. Retorna pelo return número de nós.
2. Dadas duas **árvores binárias de pesquisa** A e B, uma apontada por **raizA** e outra **raizB**. Construir uma função que verifica se a árvore B está contida na A. A árvore B é uma árvore com apenas dois níveis compondo uma família nodal completa. Isto é, B é uma árvore composta pelo nó raiz, descendente esquerdo e descendente direito. Retornar pelo return 1, se sim e 0 se não.
3. Considere **uma árvore binária qualquer**, cujo nó raiz está armazenado na variável **raiz**. Construir uma função que recebe o

endereço do nó raiz e localiza um elemento V lido na main e passado na lista de parâmetros. Se existir na árvore retorne pelo return o endereço de localização de V na árvore. Caso não exista retorne NULL.

4. Considere uma **árvore binária de pesquisa**, cujo nó raiz está armazenado na variável **raiz**. Construir uma função RECURSIVA, que recebe o endereço do nó raiz e localiza um elemento V lido na main e passado pela lista de parâmetros. Se existir na árvore retorne pelo return o endereço de localização de V na árvore. Caso não exista retorne NULL.
5. Considere **uma árvore binária qualquer**, cujo nó raiz está armazenado na variável **raiz**. Construir uma função que verifica quantos nós folha a árvore possui.
6. Considere **uma árvore binária qualquer**, cujo nó raiz está armazenado na variável **raiz**. Construir uma função que verifica quantos nós possuem os dois descendentes (isto é, grau 2).
7. Considere **uma árvore binária qualquer**, cujo nó raiz está armazenado na variável **raiz**. Construir uma função que verifica quantos nós dessa árvore possuem apenas um dos descendentes.