

## Trabalho: 08-arvore-busca

Linguagens: C

Data de abertura: 2016/10/10 14:00:00

Data limite para envio: 2016/10/17 12:00:00 (encerrado)

Número máximo de envios: 25

Casos-de-teste abertos: [casos-de-teste.tgz](#)

## Árvore binária de busca

Este trabalho consiste em escrever um programa com funções para manipular uma árvore de busca binária para inteiros de acordo com as instruções a seguir. Inicialmente o programa deve criar uma árvore vazia e em seguida deve processar a entrada-padrão. A entrada contém comandos que podem ser da seguinte forma:

- inserir x

Ao ler o comando inserir x, onde x é um inteiro, o programa deve inserir a chave x na árvore de busca binária. A árvore não deve ter elementos repetidos, então se x já pertence à árvore ele não deve ser inserido de novo. Se não houver memória para essa operação, o programa deve imprimir "memoria insuficiente" em uma linha e deve continuar a execução.

- excluir x

Ao ler o comando excluir x, onde x é um inteiro, o programa deve remover a chave x da árvore binária de busca. Se x não estiver na árvore o programa não deve fazer nada. O programa deve usar o sucessor de um nó para a substituição de um nó que tenha dois filhos.

- buscar x

Ao ler o comando buscar x, onde x é um inteiro, o programa deve buscar a chave na árvore e imprimir "pertence" ou "nao pertence" em uma linha.

- minimo  
maximo

Ao ler desses comandos o programa deve imprimir a menor chave ou a maior chave na árvore. Se a árvore estiver vazia então o programa deve imprimir "vazia" em uma linha.

- pos-ordem  
em-ordem  
pre-ordem  
largura

Ao ler um desses comandos o programa deve imprimir as chaves na ordem em que forem visitadas por um percurso em profundidade em pós-ordem, em-ordem, em pré-ordem ou por um percurso em largura. As chaves devem ser impressas seguidas por um espaço, em uma única linha. Se a árvore estiver vazia então o programa deve imprimir "vazia" em uma linha.

- terminar

Ao ler esse comando o programa deve desalocar a árvore e todos os nós dela e terminar.

## Exemplo

### Entrada:

```
inserir 50
inserir 30
inserir 40
inserir 10
inserir 20
pre-ordem
excluir 50
pre-ordem
terminar
```

### Saída:

```
50 30 10 20 40
30 10 20 40
```

## Observações

- Cada operação deve ser implementada por uma função.
-