

Atividade de Programação 7

Árvore Binária de Busca

Algoritmos e Estruturas de Dados I

Tempo limite: 1s

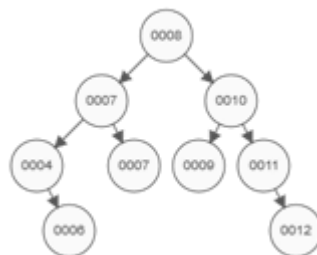
O estudo de Árvores Binárias tem diversas aplicações. Pode-se citar algumas como: Busca eficiente de informações em banco de dados, particionamento de polígonos em jogos digitais 3D, representação de expressões em compiladores, criação de números aleatórios, compressão de imagens, etc.

Dentre essas aplicações pode-se destacar a estrutura da árvore binária de busca.

Uma árvore binária é considerada de busca se para cada nó com valor **X** que a compõe, possui as seguintes propriedades:

- Todos os elementos da subárvore esquerda têm valor **menor** que **X**;
- Todos os elementos da subárvore direita têm valor **maior ou igual** à **X**.

Por exemplo:



Deve-se escrever um programa que construa uma árvore binária de busca com base em uma sequência de números positivos. Logo após deve-se encontrar a sequência percorrida para encontrar um item e altura da árvore construída.

Entrada

A entrada para cada teste se inicia informando a quantidade de elementos inseridos na árvore **N** ($1 \leq N \leq 1024$), o valor **V** ($V \geq 0$) do elemento que será encontrado. Na segunda linha são fornecidos os elementos (**E**₁, **E**₂, ..., **E**_n) que serão inseridos na árvore. Onde, $0 \leq E_i \leq 1024$, com $i = 1, \dots, n$.

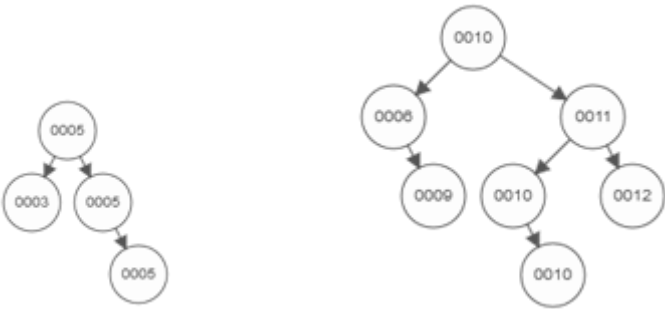
Saída

A saída do programa é exibida em cinco linhas:

- 1. 1ª linha: Exibe o percorrimto da árvore em Pré-ordem;
- 2. 2ª linha: Exibe o percorrimto da árvore em Em-ordem;
- 3. 3ª linha: Exibe o percorrimto da árvore em Pós-ordem;
- 4. 4ª linha: Exibe a sequência percorrida para encontrar o valor (incluindo no final o valor busca). Caso o valor não seja encontrado deve-se exibir o valor: -1 (menos um);
- 5. 5ª linha: Exibe a altura da árvore.

Dicas:

- A árvore proposta, deve aceitar elementos repetidos. Nesse caso os elementos iguais serão alocados na subárvore direita do elemento encontrado. Por exemplo:



Restrições:

- O programa deve ser escrito em C;
- Deve-se ter no código as funções de: inicialização, inserção, percorrimto em pré-ordem, em-ordem, pós-ordem e busca;
- Deve-se utilizar uma TAD Árvore Binária.

Exemplos:

Exemplo de Entrada	Exemplo de Saída
10 8 4 10 5 6 8 9 3 2 64 0	4 3 2 0 10 5 6 8 9 64 0 2 3 4 5 6 8 9 10 64 0 2 3 9 8 6 5 64 10 4 4 10 5 6 8 5

Exemplo de Entrada	Exemplo de Saída
12 20 17 7 4 16 17 1 8 7 13 9 14 13	17 7 4 1 16 8 7 13 9 14 13 17 1 4 7 7 8 9 13 13 14 16 17 17 1 4 7 9 13 14 13 8 16 7 17 17 -1 6

Exemplo de Entrada	Exemplo de Saída
15 30 0 30 24 4 14 1 22 20 2 15 17 5 11 23 8	0 30 24 4 1 2 14 5 11 8 22 20 15 17 23 0 1 2 4 5 8 11 14 15 17 20 22 23 24 30 2 1 8 11 5 17 15 20 23 22 14 4 24 30 0 0 30 8