

Algoritmos e Estruturas de Dados 1

Atividade F5– Usada para computar frequência

Prazo final de entrega - 08/6/2021 (Terça)

ENPE 2 - DC-UFSCar

1. Introdução

Nesta atividade vamos desenvolver o percurso de uma árvore. O programa entregue deve seguir os itens abaixo:

- Esta atividade deverá ser feita individualmente e plágio não será tolerado;
- A função pedida deve seguir a assinatura indicada na sua descrição para fins de correção;
- Outras funções auxiliares podem ser criadas para facilitar o desenvolvimento da recursão, desde que as funções pedidas no enunciado estejam presentes e se comportem de acordo com o que foi pedido;
- O estudante deverá usar um arquivo **f5.c** como base para o desenvolvimento. Este arquivo possui definições de tipos, algumas funções implementadas, o esqueleto das funções que devem ser implementadas pelo estudante, bem como testes no corpo da função main.
- A atividade **F5** deve ser entregue no run codes (<https://run.codes>) em um arquivo f5.c com as funções pedidas e um cabeçalho com as informações do estudante (nome, curso, RA);
- Cada estudante deve se cadastrar no run codes (<https://run.codes>) informando Nome Completo, escolhendo “UFSCar - Universidade Federal de São Carlos” no campo Universidade e colocando seu RA no campo Número da Matrícula. Depois de cadastrado, basta logar no run codes e se matricular na disciplina “1001502 - Algoritmos e Estruturas de Dados 1”, Professores: Marcela Xavier Ribeiro, Turma: **AED-ENPE2-2021**. Usando o código da Matrícula: **SLF1**.
- Compile o seu trabalho usando o compilador GCC. Sugestão de ambiente Codeblocks.

2 Tarefa

Considere as declarações de uma árvore binária de busca:

```
typedef int Chave;

typedef struct no{
    Chave chave;

    int tam;

    struct no *esq, *dir, *pai;
} No;

typedef struct arvore{
    No * raiz; } Arvore;
```

Dado o arquivo F5.c com a função main e a implementação de inserção e destruição de uma árvore binária de busca, **implemente as três funções a seguir, completando o código:**

Imprime os nós da árvore no percurso em ordem:

void em_ordem(No * raiz);

Imprime os nós da árvore no percurso pré ordem:

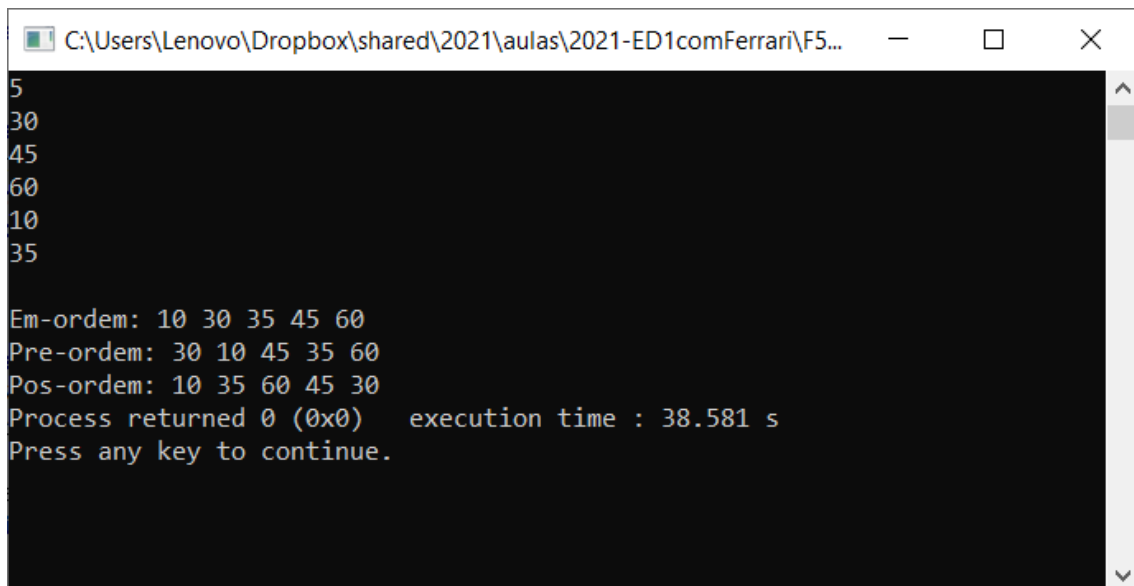
void pre_ordem(No * raiz);

Imprime os nós da árvore no percurso pós ordem:

void pos_ordem(No * raiz);

O parâmetro raiz é um ponteiro para a raiz da árvore cujas chaves dos nós serão impressas.

O programa main já é fornecido no código e sua execução é a inserção de inteiros. O primeiro inteiro fornecido é o número de nós que seja inserido na árvore binária a ser criada. Os demais números já são os valores das chaves dos nós inseridos na árvore. Exemplo de execução:



```
C:\Users\Lenovo\Dropbox\shared\2021\aulas\2021-ED1comFerrari\F5...
5
30
45
60
10
35

Em-ordem: 10 30 35 45 60
Pre-ordem: 30 10 45 35 60
Pos-ordem: 10 35 60 45 30
Process returned 0 (0x0)   execution time : 38.581 s
Press any key to continue.
```

O usuário digitou 5 para dizer que será construída uma árvore binária de busca com 5 valores e depois forneceu os 5 valores 30,45,60,10 e 35. O programa retornou os nós impressos nas árvores usando os percursos em-ordem, pre-ordem e pos-ordem