

Algoritmos e Estruturas de Dados 1

Atividade F5– Usada para computar frequência

Prazo final de entrega - 18/6/2021

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 **f6.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 **F6** 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 (para que a saída impressa já esteja no formato dos casos de testes cadastrados run.codes), implemente as seguintes operações:

a) Encontrar o maior o nó com a maior chave da árvore enraizada em raiz e retorna um ponteiro para ele

No * maior(No *raiz);

b) Encontrar o menor nó com a maior chave da árvore enraizada em raiz e retorna um ponteiro para ele

No * menor(No *raiz);

c) **Inserere recursivamente ou iterativamente um novo nó criado na árvore enraizada por raiz retornando um ponteiro para ele.**

No * insererec(No * raiz, No * novo);

A função main já é fornecida. Um exemplo de execução :

Entrada (primeiro número é o número de nós a ser inserido na árvore, e os demais são as chaves dos nós inseridos na árvore):

5

50

20

70

10

25

Saída:

Altura da arvore = 2

Maior elemento da arvore = 70

Menor elemento da arvore = 10