

## 2º Trabalho

Curso: Engenharia da Computação  
Disciplina: Estruturas de Dados  
Prof. Jarbas Joaci de Mesquita Sá Junior  
Universidade Federal do Ceará – UFC/Sobral

Entrega: 25/01/2021 via e-mail para

### Obs.:

- O trabalho é individual e não será recebido após a data mencionada;
- Preferencialmente fazer o trabalho usando a IDE Dev-C++;
- Enviar **todos** os arquivos do projeto, exceto os executáveis (.exe). Organizar os arquivos nas pastas q1 e q2.
- O uso da diretiva `#include` sem um header file (.h) implicará nota zero no código. Por exemplo, **não** usar `#include "nomearquivo.c"`

1. Implemente a TAD “arvb.h” (Árvore Binária de Busca) e acrescente as seguintes funções:

a) função que retorne a quantidade de folhas que possuem o campo `info` com número primo. Essa função deve obedecer ao protótipo:

```
int folhas_primo(ArvB* a);
```

b) função que retorne a quantidade de nós que possuem campos `dir` e `esq` diferentes de `NULL`. Essa função deve obedecer ao protótipo:

```
int subarvs_difnull(ArvB* a);
```

c) função que retorne a quantidade de nós cujas subárvores esquerda e direita tenham a mesma altura. Essa função deve obedecer ao protótipo:

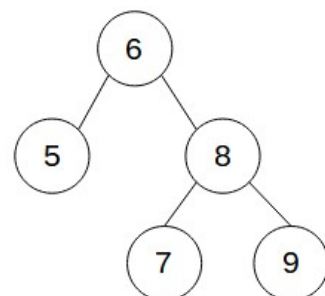
```
int subarvs_ig_alt(ArvB* a);
```

d) função que imprima os elementos por níveis. Essa função deve obedecer ao protótipo:

```
void impressao_arv_niveis(ArvB* a);
```

Por exemplo, na árvore da figura abaixo, a impressão deve ser:

6 – nível 0  
5, 8 – nível 1  
7, 9 – nível 2



A seguir, execute o seguinte programa.

```
#include <stdio.h>
#include <stdlib.h>
#include "arvb.h"

int main(void) {

    Arv* arvA = arvb_cria_vazia();
    arvA=arvb_inserte(arvA,43);
    arvA=arvb_inserte(arvA,51);
    arvA=arvb_inserte(arvA,21);
    arvA=arvb_inserte(arvA,4);
    arvA=arvb_inserte(arvA,45);
    arvA=arvb_inserte(arvA,29);
    arvA=arvb_inserte(arvA,3);
    arvA=arvb_inserte(arvA,23);
    arvA=arvb_remove(arvA,4);
    printf("'Nós subarvs dif null %d\n'",subarvs_difnull(arvA));
    printf("'Nós subarvs igual alt.  %d\n'",subarvs_ig_alt(arvA));
    printf("'Folhas com val. primos %d\n'",folhas_primo(arvA));

    printf("'-----\n');
    impressao_arv_niveis(arvA);
    printf("'-----\n');

    arvb_libera(arvA);

    system("'PAUSE'");
    return 0;
}
```

2. Implemente os algoritmos **BubbleSort**, **InsertionSort**, **QuickSort** e **HeapSort** e calcule o tempo de cada um para ordenar vetores de tamanho  $10^3$ ,  $10^4$ ,  $10^5$  e  $10^6$  com os elementos dispostos de três formas: crescente, decrescente, e aleatória. Elabore um pequeno relatório com os dados obtidos (no máximo, com três páginas – formato PDF).

**Obs.:** O tempo deve ser dado em milissegundos.