

Trabalho prático 1 - Árvore AVL

Allan Cedric G. B. Alves da Silva - GRR20190351

15 de dezembro de 2020

1 Introdução

Este é um relatório realizado para o primeiro trabalho da disciplina de **Algoritmos e Estruturas de Dados III - (CI1057) - ERE2**, do curso de **Ciência da Computação, UFPR**.

Em um primeiro momento será discorrido brevemente sobre a implementação da árvore AVL para a resolução do problema do trabalho. E no final, será tratado sobre a resolução propriamente dita.

2 Implementação da Árvore AVL

Para a implementação, foi reciclado toda uma biblioteca já realizada no passado de uma **árvore BST**, com alterações nas funções de inserção e remoção para adaptá-las ao paradigma de balanceamento de uma árvore AVL. A implementação como um todo foi feita na linguagem **C**.

A biblioteca da **árvore AVL** se encontra no diretório chamado *ArvoreAVL*, mais precisamente o conjunto de arquivos: *ArvoreAVL/avl.h* e *ArvoreAVL/avl.c*. A estrutura de um nodo da árvore AVL se encontra a seguir:

```
/*
| Estrutura de dado de um nodo (p/ AVL) com chave simples => int
*/
typedef struct node_t
{
| struct node_t *left;
| struct node_t *right;
| struct node_t *parent;
| int height;
| int key;
} AVL;
```

Figura 1: Estrutura de um nodo para a árvore AVL (Arquivo *ArvoreAVL/avl.h*)

Foi aderido fazer o máximo que pudesse de técnicas recursivas para facilitar a implementação, no mais os comentários no código da biblioteca em questão completam este relatório. Vale ressaltar que foi feito um *setup* para o algoritmo de remoção da árvore AVL, o qual pode ser escolhido a tática com antecessor ou sucessor, como mostra a figura seguir:

```
/*
 * Setup para o algoritmo de remoção, pode escolher pelo formato com sucessor ou antecessor
 */

/* Comente essa linha imediatamente abaixo para usar o sucessor, por default está configurado para antecessor */
#define _PREDECESSOR_

#ifndef _PREDECESSOR_
#define _SUCESSOR_
#endif

#ifndef _PREDECESSOR_
#ifndef _SUCESSOR_
#define _PREDECESSOR_
#endif
#endif
```

Figura 2: Setup do algoritmo de remoção da árvore AVL (Arquivo *Arvore-AVL/avl.h*)

3 Problema proposto

O objetivo principal deste trabalho é implementar a inserção e remoção de uma árvore AVL, e realizar um teste simples de impressão. O arquivo fonte de teste, *myavl.c*, possui a implementação para a impressão 'emOrdem' das chaves da árvore AVL, com os respectivos níveis onde se encontram.

O *output* esperado foi realizado de forma extremamente simples com apenas a chamada *printInOrder(rootAVL, 0)* no programa principal. O algoritmo se encontra a seguir:

```
void printInOrder(AVL *root, int level)
{
    if (root)
    {
        printInOrder(root->left, level + 1);
        fprintf(stdout, "%i,%i\n", root->key, level);
        printInOrder(root->right, level + 1);
    }
}
```

Figura 3: Algoritmo do output esperado (Arquivo *myavl.c*)