

Módulo 1 - Estruturas básicas - Lista #4

Data de entrega: 23 de abril de 2017

Modifique a sua implementação da Avl para permitir a remoção. Renomeie os arquivos para **mod1_lista4.h** e **mod1_lista4.cpp**, contendo a classe COMPLETA. Aproveite para completar e corrigir o que ficou faltando da primeira parte da Avl. Envie os arquivos zipados no arquivo numero_matricula.zip por email para **profs-eda@tecgraf.puc-rio.br**, com o assunto **[EDA] Lista 4**. Atenção: Crie um arquivo contendo a função main do seu programa para testar suas implementações, mas envie SOMENTE os arquivos e as classes solicitadas.

A assinatura da função de remoção deve ser:

```
void remove(int key);
```

E assim a estrutura da classe completa passa a ser:

```
#ifndef MOD1_LISTA4
#define MOD1_LISTA4

#include <string>

struct AvlNode
{
    int _key;

    //fator de balanceamento = hd-he
    int _balance_factor;

    AvlNode* _up;
    AvlNode* _left;
    AvlNode* _right;
};

class Avl
{
public:
    //Cria uma arvore vazia
    Avl();
```

```

//Cria uma arvore com um primeiro elemento
Avl(int key);

//Cria uma arvore a partir de outra
Avl(const Avl& orig);

//Destroi a arvore
virtual ~Avl();

//Retorna a altura da arvore
int height();

//Insere um elemento na arvore
void insert(int key);

//Remove um elemento da arvore
void remove(int key);

//Exibe a arvore no formato
//<raiz<sub-arvore esquerda><sub-arvore direita>>
void show(const std::string& title);

/** Funcoes para percorrer a arvore em ordem: */

//Posiciona o cursor no primeiro elemento
bool first();

//Posiciona o cursor no ultimo elemento
bool last();

//Atualiza o no cursor para o proximo elemento
bool next();

//Atualiza o no cursor para o elemento anterior
bool prev();

//Exibe o valor do no cursor atual
int value();

private:

/* Funcoes privadas (recursivas ou nao)
 * que julgar necessarias */

//Funcao recursiva para exibir a arvore
void show_rec(AvlNode* node);

```

```
private:
    AvlNode* _root;
    AvlNode* _cursor;
};

#endif
```