RELATÓRIO TRABALHO CI057. ÁRVORE AVL
Murilo Santos Ferreira (GRR20186317)

Particularidades principais:
- Faz remoção BST pelo antecessor
- O fator de balanceamento é guardado no nodo
- Duplicatas nao sao permitidas
- Maioria das funções modularizadas

- **ARQUIVO "myavl.c"**
  - main(): Em um while loop, lê o stdin até que o scanf() falhe em coletar o formato desejado (instrução chave), ainda no loop executa as operações na árvore e após o final do laço irá imprimir a estrutura.

- **ARQUIVO "AVLStruct.h":** Simplesmente a header para o arquivo AVLStruct.c
  - struct nodo_t: Na minha implementação, o tipo nodo tem 5 campos:
    - ptr filho esquerdo (nodo_t *esq)
    - ptr filho direito (nodo_t *dir)
    - fator de balanceamento (int fator)
    - altura (int altura)
    - chave (int chave)

  Por facilidade, preferi guardar o fator de balanceamento no struct

- **ARQUIVO "AVLStruct.c"**
  - **imprimeArvore(nodo, altura)**: Faz uso da recursão para imprimir as chaves da estrutura EM ORDEM. Quando chamado, deve-se passar a raiz da árvore e o int zero para funcionamento correto. (Essa função não usa a altura dentro da struct).
  - **insereAVL(raiz, chave)**: Esta função faz primeiramente a inserção BST ignorando chaves que se repetem. Após o novo nodo ser inserido, as alturas/fatores de balanceamento são atualizados de baixo para cima com ajuda da recursão E nodos rotacionados se necessário.
  - **removeAVL(raiz, chave)**: Após percorrer árvore com a propriedade BST, o nodo é removido se for encontrado, do contrário nada acontece. Primeiro, é executado a remoção BST usando o método do antecessor (os casos de teste falharam com o mtd. do sucessor). Após isto, é feito a atualização das alturas/fatores de baixo para cima e também o balanceamento, assim como na inserção.
  - **balanceamentoAVL(nodo)**: Essa função verifica o fator de balanceamento do nodo e faz as operações necessárias. Se este estiver balanceado, o mesmo é retornado de maneira intacta. Caso contrário, então existem 4 casos de rotação para cada tipo de desbalanceamento. A fórmula do fator é dada por $altura(filho_{esquerda}) - altura(filho_{direita})$, então números maiores que 1 indicam desbalanceamento na subárvore esquerda, os menores que -1 indicam o mesmo na subárvore direita. *(Para o funcionamento correto da função, ela deve ser usada de baixo para cima na estrutura).* Lista de casos:

        - Dupla-Esquerda => rotacaoDir()
        - Esquerda-Direita => rotcaoEsquerdaDireita()
        - Dupla-Direita => rotacaoEsquerda()
        - Direita-Esquerda => rotocaoDireitaEsquerda()
    - **novoNodo(chave)**: simplesmente retorna um ponteiro para um novo nodo, sendo este alocado dinamicamente. Com exceção da chave, todos os campos da estrutura são nulos.
    - **atualizaAltura(nodo)**: com informação dos filhos, o fator de balanceamento e altura do nodo são atualizados. *(Para o funcionamento correto da função, ela deve ser usada de baixo para cima na estrutura)*
        - Altura: $max(altura(filho_{esquerda}),\ altura(filho_{direita})) + 1$
        - Fator: $altura(filho_{esquerda}) - altura(filho_{direita})$