

Struct do nó:

```
struct no:  
    int chave;  
    no_t *esq, *dir, *pai;
```

Obs.: A estrutura do node possui sua chave(um valor int), e ponteiros com seus vizinhos(filho da esquerda, da direita e seu pai).

criaNo(): Função que serve para alocar e iniciar a raiz de um árvore com sua chave, e retorna o seu ponteiro. Os valores iniciam como NULL, a não ser a chave.

altura(): percorre a árvore a partir de um node calculando recursivamente a altura da esquerda e da direita.

fatorBalanceamento(): Calcula o fator de balanceamento de um node utilizando a função de altura citada acima

Rotações: realizam as rotações para poder balancear. São elas: Esquerda, Direita, Esquerda Dupla e Direita Dupla.

Balancear e BalancearAux: As funções em conjunto verificam e se necessário realizar o balanceamento da árvore, verificando de um node enviado até a raiz da árvore.

Inclui e IncluiAux: As funções realizam a inserção de um elemento na árvore, e chamam a função de balanceamento, enviando o node do local em que ocorreu a inserção para verificar e balancear daquele local até a raiz.

Busca(): Passa pela árvore procurando um node, e o retorna se encontrar, se não retorna NULL.

min(): retorna o valor mais a esquerda a partir de um node.

max(): retorna o valor mais a direita a partir de um node.

ajustaNoPai(): Função que de fato realiza a remoção de um valor, atualizando o pai do valor removido com seu respectivo filho, e substituindo por um novo.

sucessor(): retorna valor sucessor de um node

antecessor(): retorna valor antecessor de um node

exclui(): Exclui um valor, e balanceia a árvore adequadamente após sua exclusão.

nivel(): calcula e retorna o nível de um node.

printNivel e printNivelAux(): As funções servem para realizar a saída imprimindo a árvore EmOrdem junto com os níveis dos nodes.

main(): A main que cuida de ler o teclado, criar a árvore e realizar a saída de acordo com a especificação do projeto.