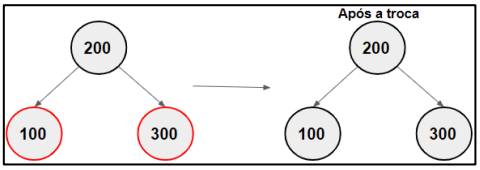
Universidade Federal do Piauí – UFPI
Campus Senador Helvídio Nunes de Barros – CSHNBCurso de Sistemas de Informação Bloco: IV
Disciplina: Estruturas de Dados II
Professora: Juliana Oliveira de Carvalho
Acadêmico: Dayan Ramos Gomes Matrícula: 20209001285

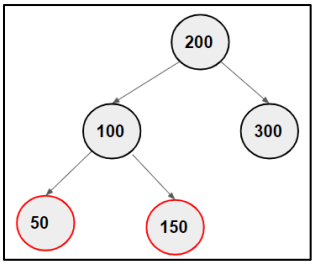
ATIVIDADE DE FIXAÇÃO 7

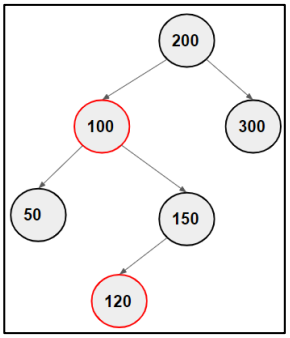
1) Descreva passo a passo usando texto e desenho e o anexo I para inserir elementos em uma árvore Vermelha-Preta para as seguintes sequências:

Sequência: 200, 100, 300, 50, 150, 120

Na inserção do 200, como a raiz é NULL o 200 é inserido com a cor RED, mas quando a função retorna no fim de tudo a Raiz SEMPRE recebe a cor BLACK na linha 118 do código, em seguida o valor 100 entra na linha cem do código e nessa chamada recursiva é inserido à esquerda da Raiz, o valor 300 entra na linha 102 do código e nessa chamada recursiva é inserido à direita da Raiz, na volta da chamada o algoritmo entra na linha 110 que troca a cor dos nós, onde a Raiz fica RED e seus dois filhos BLACK, mas no fim a Raiz fica BLACK novamente na linha 118 do código. Para inserir o 50 é feita duas chamadas recursivas na linha 100 do código e o 50 é inserido na cor RED à esquerda do 100. Na inserção do 150 é feita duas chamadas recursivas nas linhas 101 e 103, respectivamente, assim, ele é inserido à direita do 100 com a cor RED. Para inserir o 120 o algoritmo faz três chamadas recursivas na linha 101, 103 e 101 respectivamente, assim o 120 é inserido à esquerda do 150, quando o algoritmo volta na primeira chama recursiva, ele entra na linha 110 e troca a cor 100 para RED e de seus filhos da esquerda e direita para BLACK





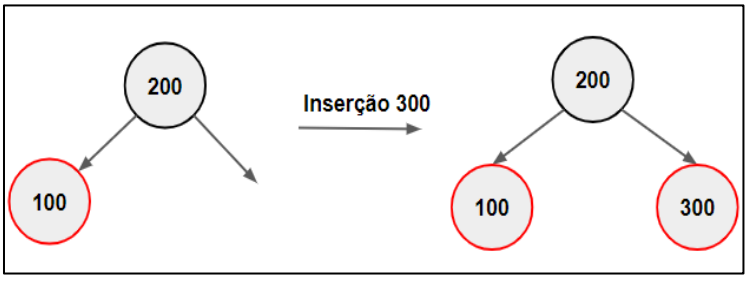



```
83    if (H == NULL){
84        struct NO *novo;
85        novo = (struct NO *)malloc(sizeof(struct NO));
86        if (novo == NULL){
87            *resp = 0;
88            return NULL;
89        }
90        novo->info = valor;
91        novo->cores = RED;
92        novo->dir = NULL;
93        novo->esq = NULL;
94        *resp = 1;
95        return novo;
96    }
97    if (valor == H->info)
98        *resp = 0; // Valor duplicado
99    else{
100       if (valor < H->info)
101           H->esq = insereNO(H->esq, valor, resp);
102       else
103           H->dir = insereNO(H->dir, valor, resp);
104   }
105   if (cor(H->dir) == RED && cor(H->esq) == BLACK)
106       H = rotacionaEsquerda(H);
107   if (cor(H->esq) == RED && cor(H->esq->esq) == RED)
108       H = rotacionaDireita(H);
109   if (cor(H->esq) == RED && cor(H->dir) == RED)
110       trocaCor(H);
```

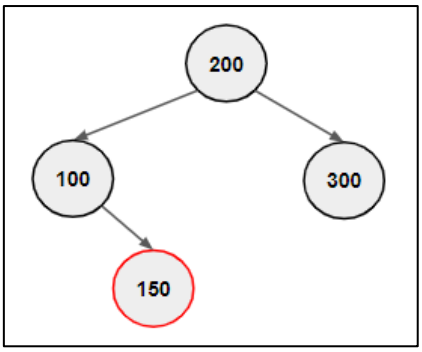
```
114 int insere_ArvLLRB(ArvLLRB *raiz, int valor){
115     int resp;
116     *raiz = insereNO(*raiz, valor, &resp);
117     if ((*raiz) != NULL)
118         (*raiz)->cores = BLACK;
119     return resp;
120 }
```

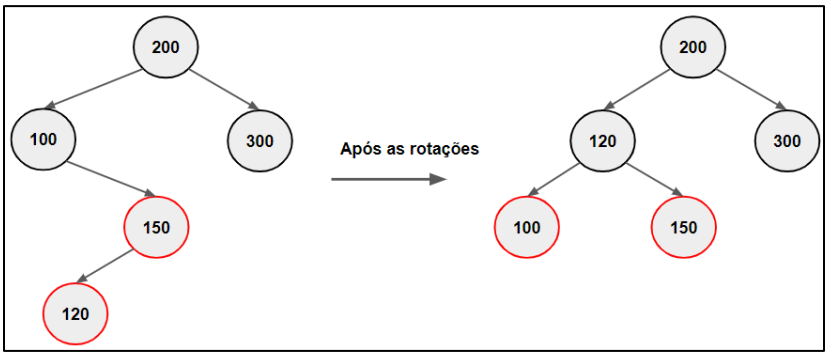
Sequência: 100, 200, 300, 150, 120

Na inserção do 100, como a raiz é NULL o 100 é inserido com a cor RED, mas quando a função retorna no fim de tudo a Raiz SEMPRE recebe a cor BLACK na linha 118 do código, para inserir o 200 o algoritmo faz uma chamada recursiva na linha 102 e insere o 200 na cor RED à direita do 100, mas na volta da chamada a árvore é rotacionada, para inserir o 300 é feita uma chamada recursiva na linha 103 e o 300 é inserido na cor RED à direita do 200. Para inserir o 150 é feita duas chamadas recursivas nas linhas 101 e 103, respectivamente, e o 150 é inserido, na



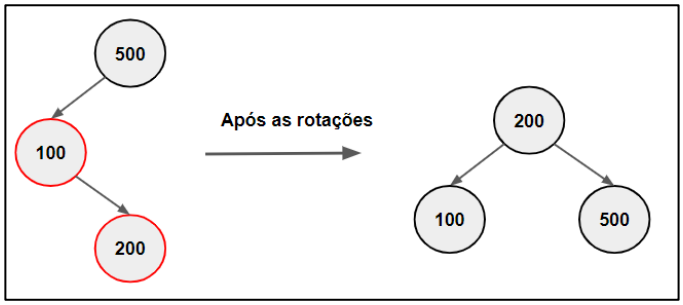
volta da primeira chamada recursiva o algoritmo entra na linha 110 e troca a cor da raiz pra RED e de seus filhos da direita e esquerda para BLACK, no fim a raiz recebe a cor BLACK novamente na linha 118. Para inserir o 120 é feita três chamadas recursivas nas linhas 101, 103 e 101 respectivamente, e o 120 é inserido na cor RED à esquerda do 150, na volta das recursividades são feitas duas rotações que balanceiam a árvore.

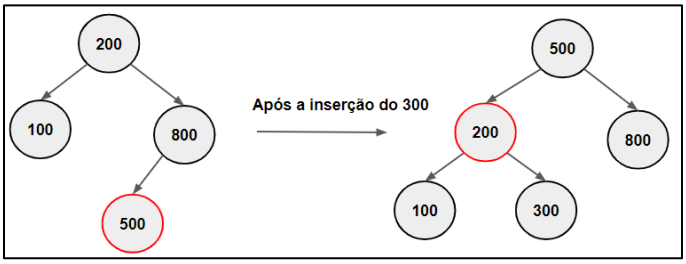


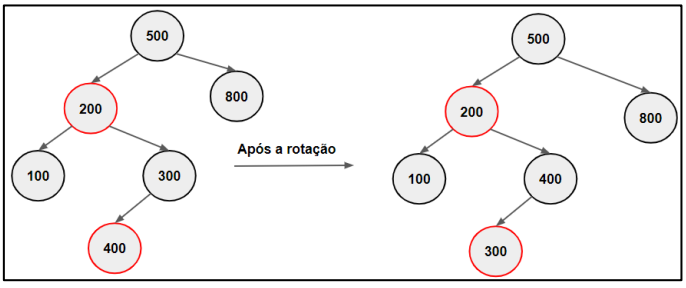


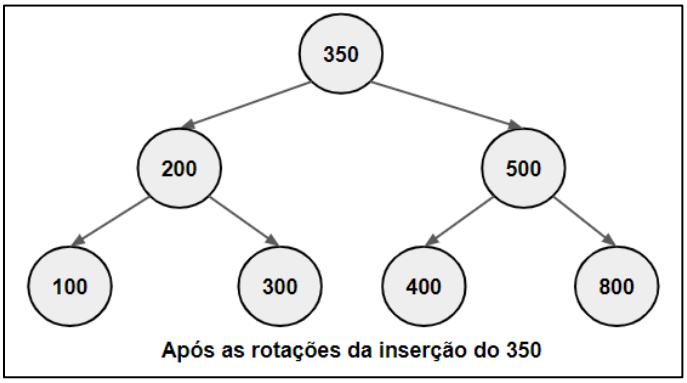
Sequência: 500, 100, 200, 800, 300, 400,350

Na inserção do 500, como a raiz é NULL o 500 é inserido com a cor RED, mas quando a função retorna no fim de tudo a Raiz SEMPRE recebe a cor BLACK na linha 118 do código, para inserir o 100 é feita uma chamada recursiva na linha 101 do código e o 100 é inserido na cor RED à esquerda do 500, para inserir o 200 é feita duas chamadas recursivas nas linhas 101 e 103 respectivamente, e o 200 é inserido na cor RED à direita do 100, na volta das chamadas recursivas, são feitas duas rotações que balanceiam a árvore e também entra na linha 110 no fim e troca a cor da raiz para RED e dos filhos para BLACK, mas no fim a raiz volta a ser BLACK na linha 118. Para inserir o 800 são feitas duas chamadas recursivas na linha 103 e o 800 é inserido à direita do 500 na cor RED, na volta da recursividade é feita uma rotação e o 800 vai pro lugar do 500 e o 500 vira filho da esquerda do 800. Para inserir o 300 são feitas três chamadas recursivas nas linhas 103, 101 e 101 respectivamente e o 300 é inserido à esquerda do 500 na cor RED, mas na volta das recursividades são feitas rotações e a árvore é balanceada. Para inserir o 400, são feitas três chamadas recursivas na linha 101, 103 e 101 respectivamente e o 400 é inserido à esquerda do 300 e na volta da recursividade é feita uma rotação para balancear a árvore. Para inserir o 350 são feitas quatro chamadas recursivas nas linhas 101, 103, 101 e 103 respectivamente e o 350 é inserido à direita do 300 e na volta das chamadas recursivas são feitas rotações para que a árvore continue balanceada.









2) Descreva o funcionamento da remoção de uma árvore vermelha-preta através de exemplo.

Como exemplo de remoção, queremos excluir o valor 15 da árvore, inicia a busca pelo nó a ser removido a partir do nó 50, como o valor é menor, vai para o nó 20, como esse nó tem filho e neto NULL, ou seja da cor BLACK, chama a função *move2EsqRed()* que faz rotações na árvore e deixa o valor 15 como folha, quando volta da chamada da função, continua procurando o valor do numero 21 agora, como é menor, chama uma função recursiva para o 20, como é menor, faz uma chamada recursiva para o nó 15, como o valor é igual ao do nó, libera o nó e volta a recursividade até chegar na raiz novamente, assim, o número foi removido.