

Bacharelado em Ciência da Computação

Algoritmos e Estruturas de Dados - GBC034

Professora: Maria Camila Nardini Barioni

- João Vitor Gonçalves Oliveira – 11921bcc024

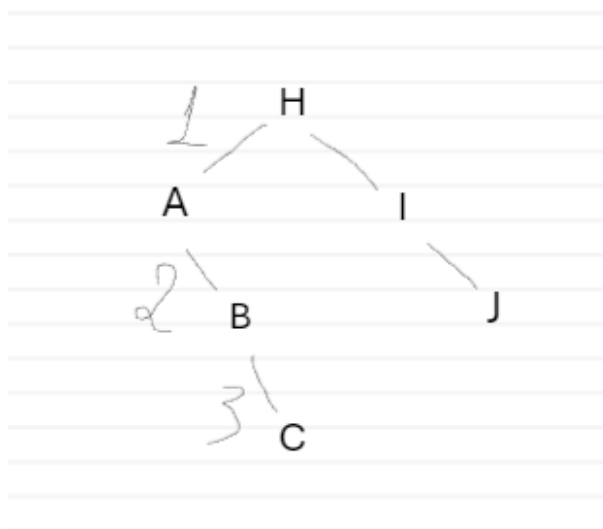
1) Considerando a estrutura de nós apresentado em sala

```
typedef struct {  
    dados;  
    no* esquerda;  
    no* direita} no;
```

Uma função para calcular a quantidade de nós que essa árvore teria é

```
int ContaNo (struct no no_atual){  
    if (no_atual == NULL) retorna 0;  
    else retorna 1 + ContaNo(no_atual->esquerda) + ContaNo(no_atual->direita);
```

2)

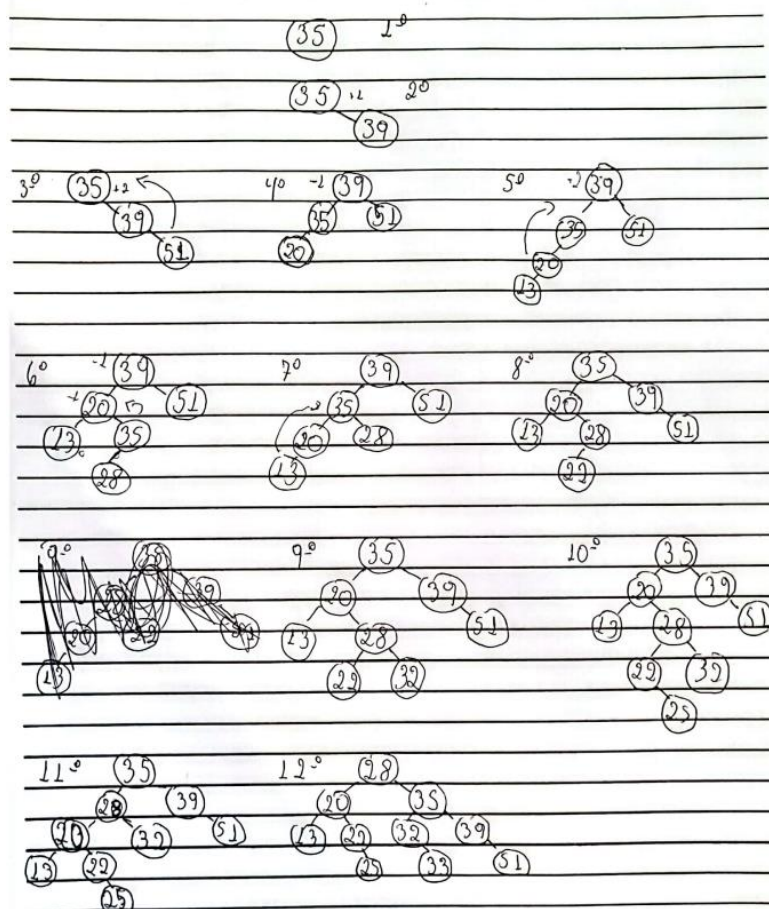


Alternativa D

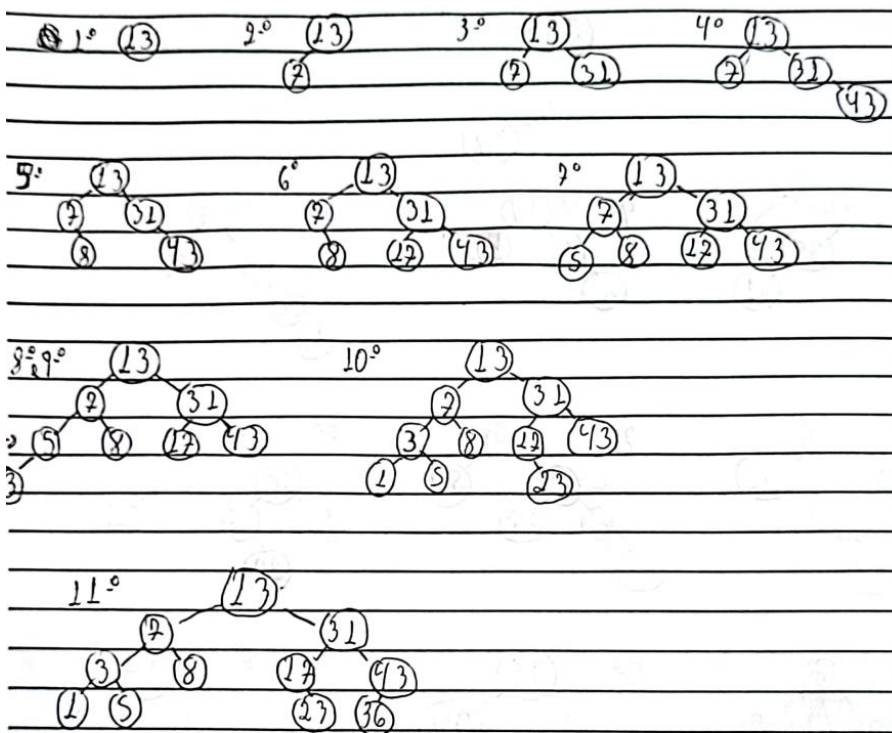
3) Percurso central ou em ordem. Alternativa D

4)

4a) 35, 39, 51, 20, 13, 28, 22, 32, 25, 33



4b) 13, 2, 31, 43, 8, 12, 5, 3, 1, 23, 16, 36



5) A)



Inserindo 62 para ter desbalanceamento direita-direita



Balanceamento:



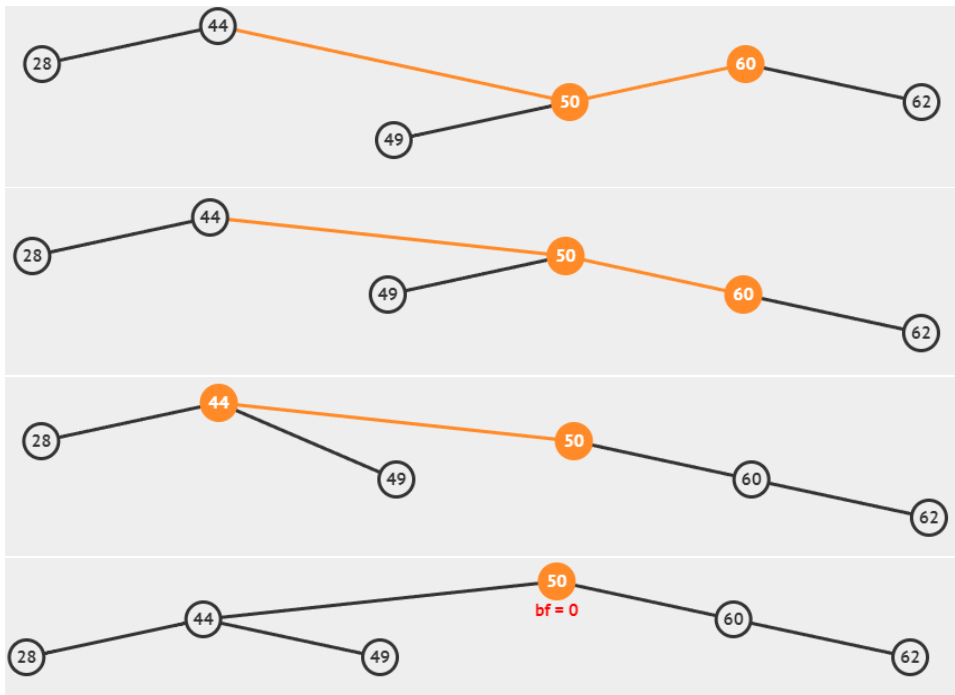
B)



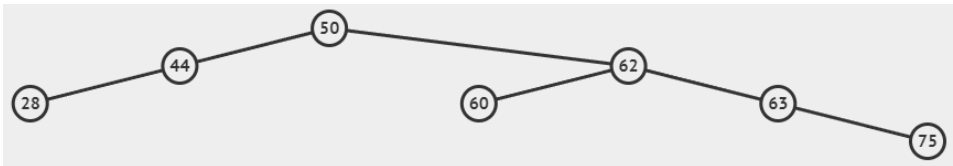
Inserindo 49 para causar desbalanceamento direita-esquerda:



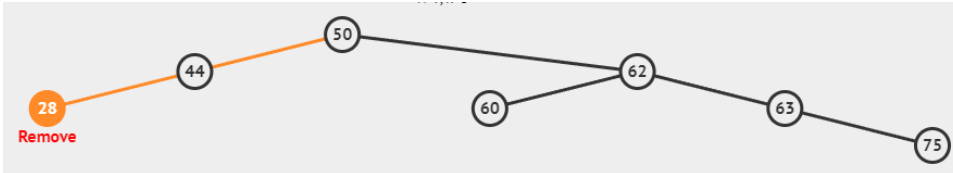
Balanceando:



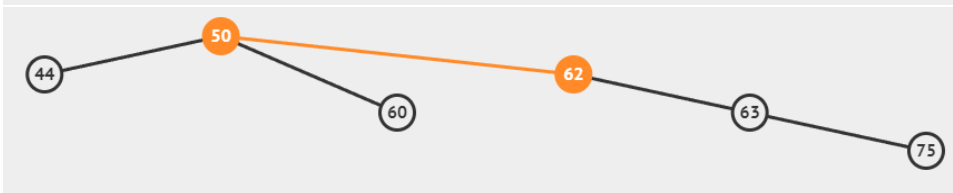
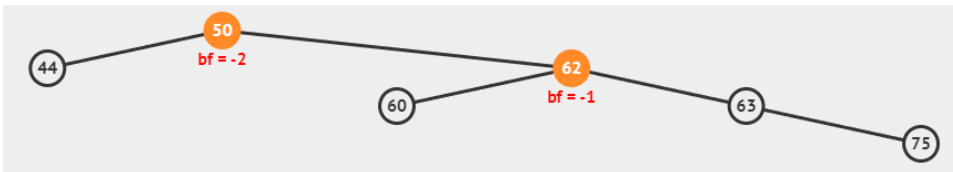
C)



Removendo 28:



Balanceando



6)

6) 42, 38, 31, 12, 19, 8

Número Circulado = Vermelho
Número sem círculo = Preto

Sópis mais Vermelho

1º 41 2º 42 3º 41

4º 38 5º 38 6º 38

7º 31 8º 31 9º 31

10º 12 11º 12 12º 12

13º 19 14º 19 15º 19

16º 8 17º 8 18º 8

Tio Vermelho
Pai e Tio mais Preto e avós Vermelhos

Tio Filho do Vovô / 2 filhos mais Preto

19º 38 20º 38 21º 38

22º 41 23º 41 24º 41

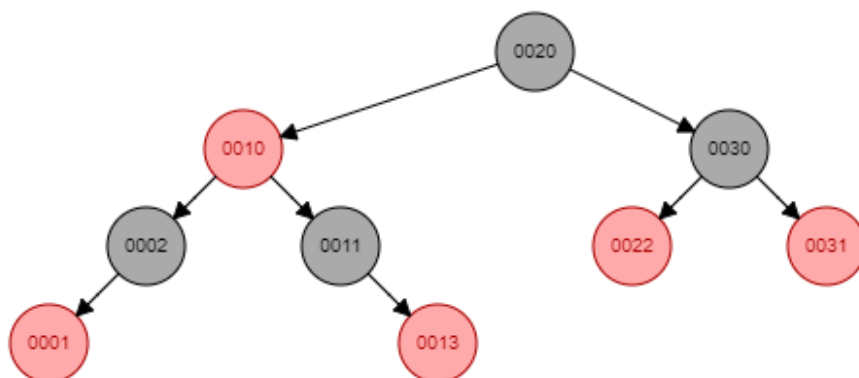
25º 12 26º 12 27º 12

28º 19 29º 19 30º 19

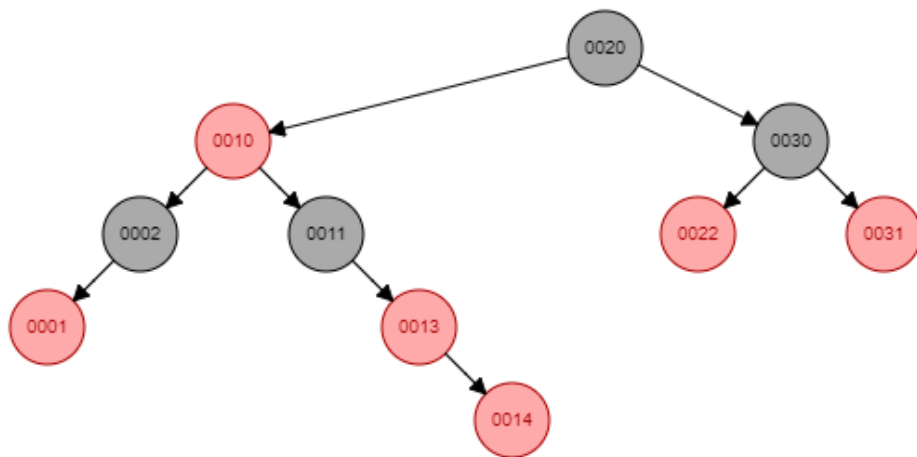
31º 8 32º 8 33º 8

Tio Vermelho
Pai e Tio Preto do Vovô

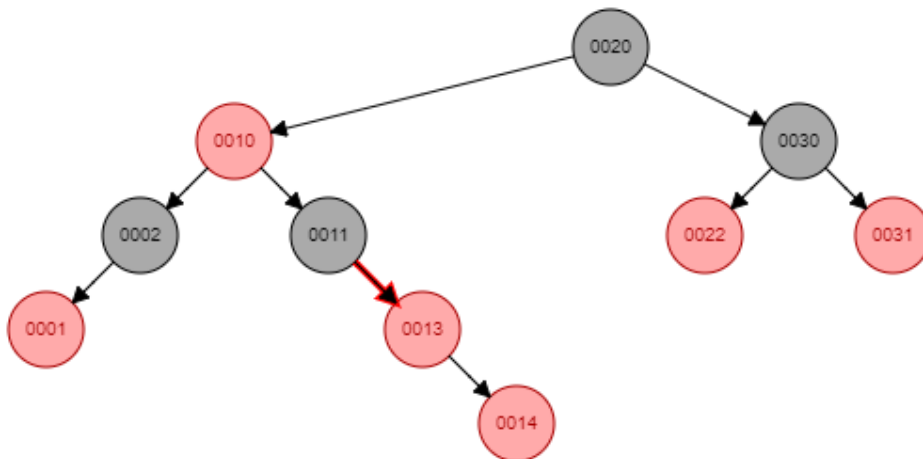
7) Levando em consideração a seguinte árvore



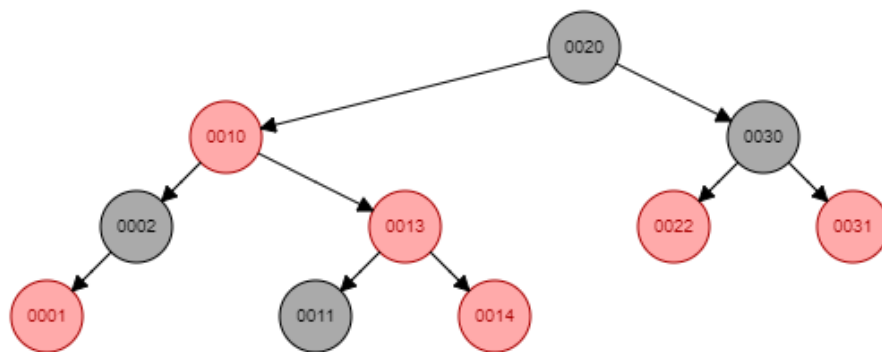
Inserindo 14



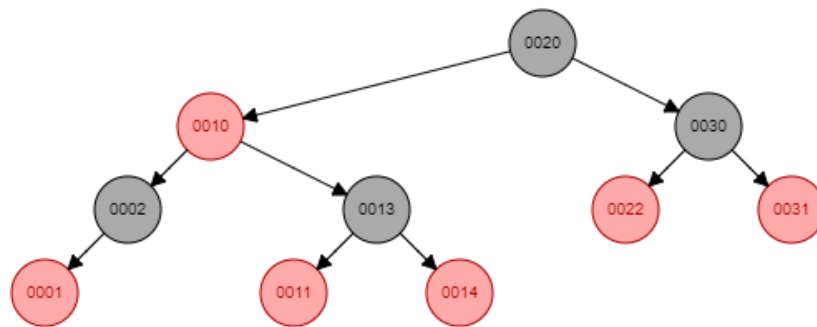
Quebra de regra: Nós vermelhos só podem ter filhos pretos



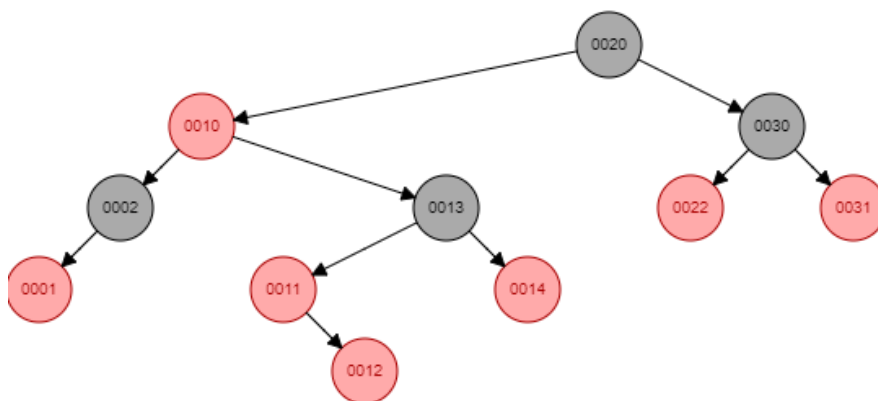
Como não há tios se conserta com uma rotação



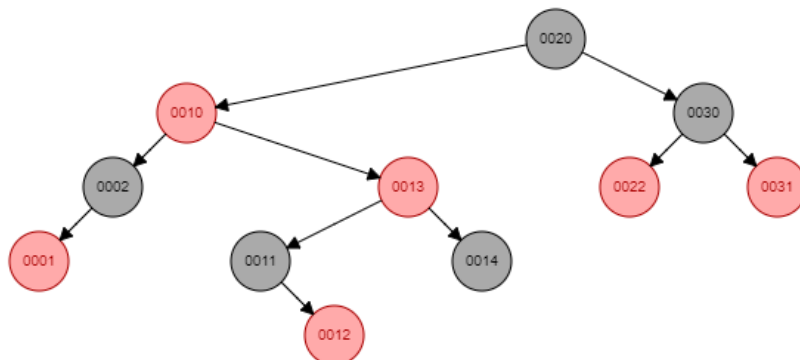
Filho de 10 não pode ser vermelho

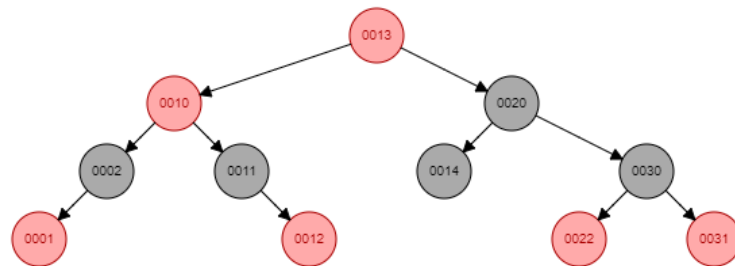
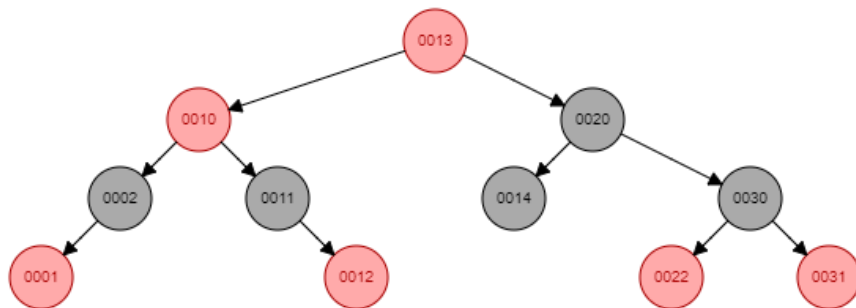
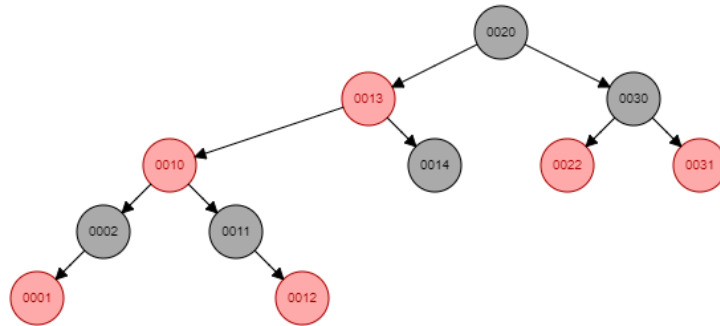


Inserindo 12:

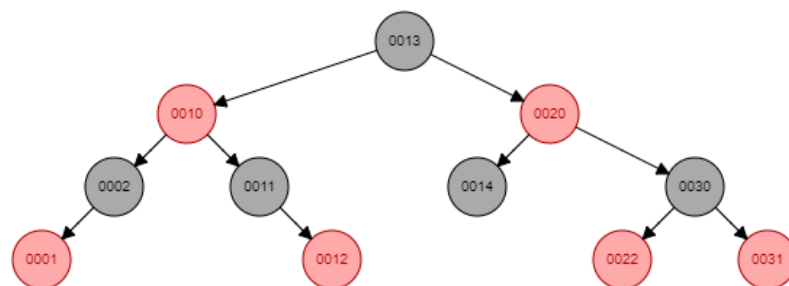


**Todos filhos de vermelho tem que ser pretos.
Tio da nova insert é vermelho, logo pai e filho viram pretos e avo vermelho**

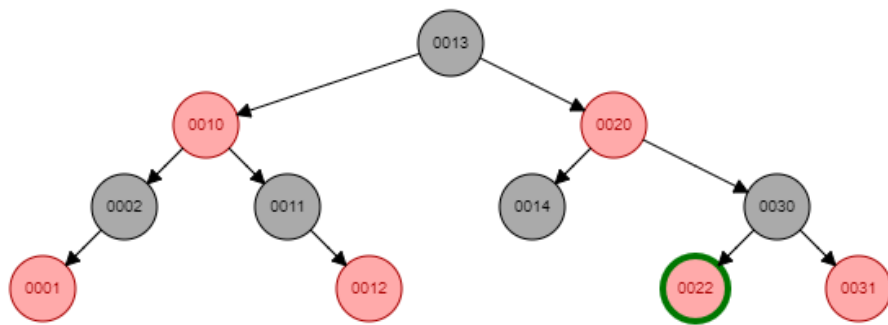




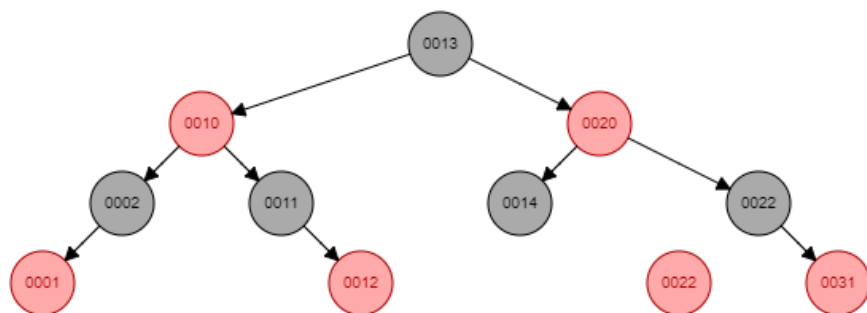
**todos filhos de vermelho devem ser pretos
para não desfazer a árvore 20 se torna vermelho**



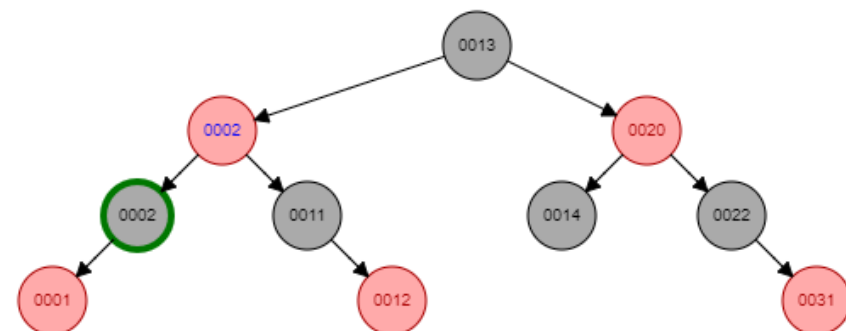
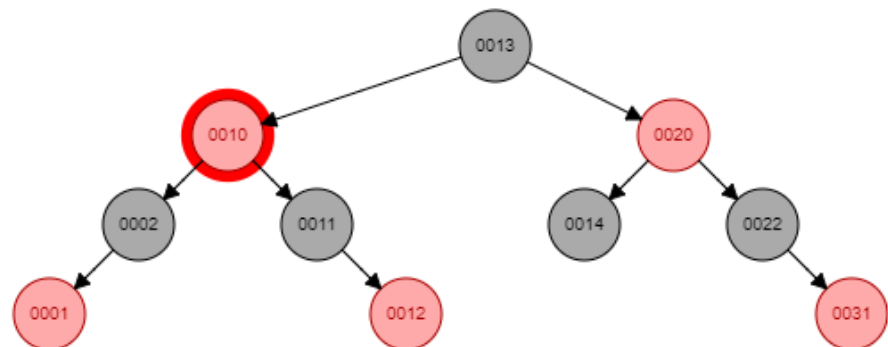
Remoção de 30:



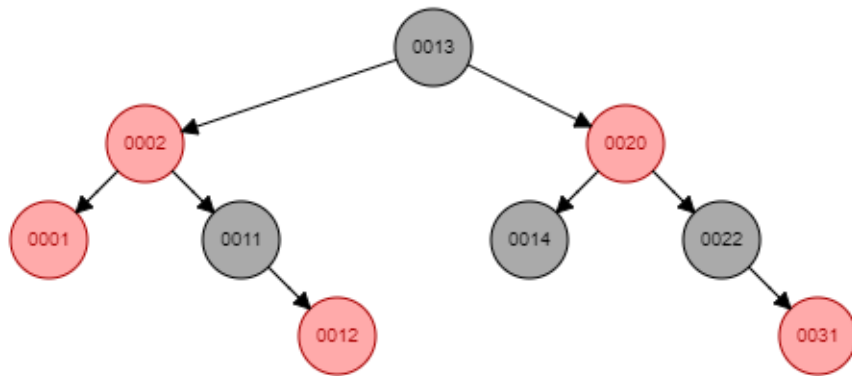
Maior valor na arvore da esquerda fica no lugar de 30



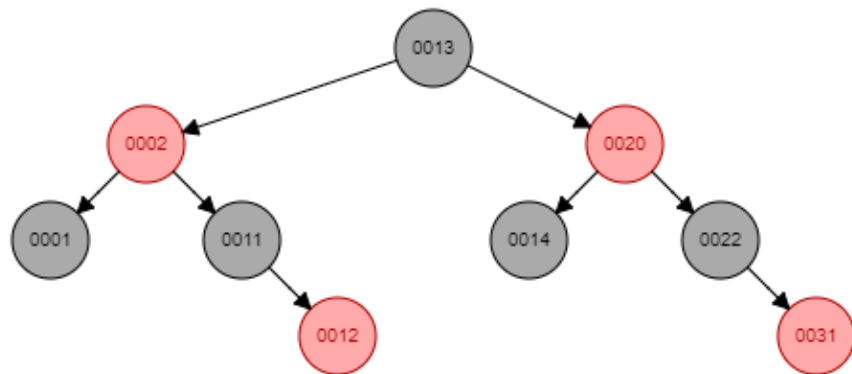
Removendo 10:



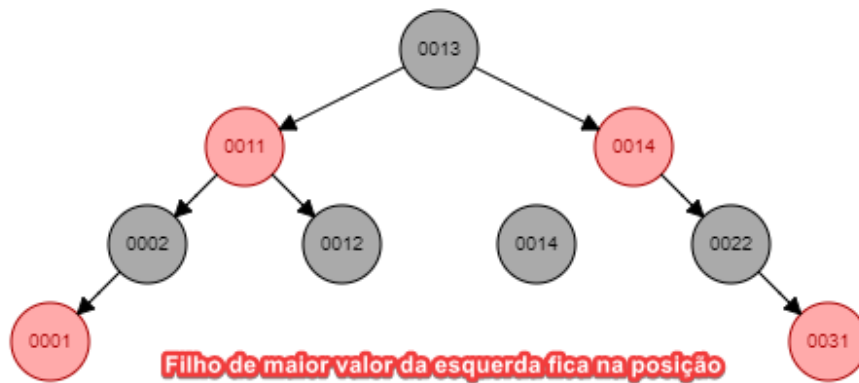
Maior valor dos filhos da esquerda fica no lugar do nó deletado



Quebra de regra dos filhos de 2



Removendo 20:



Filho de maior valor da esquerda fica na posição

Rotação

