



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SUL-RIO-GRANDENSE  
Campus Passo Fundo



# **ESTRUTURA DE DADOS II**

**Prof. Adilso Nunes de Souza**



# ÁRVORE VERMELHO- PRETO

- É um tipo de árvore binária balanceada que, diferentemente da árvore AVL, utiliza um esquema de coloração dos nós para manter o balanceamento da árvore.
- Também conhecida por árvore rubro-negra ou red-black.
- Foi originalmente criada por Rudolf Bayer em 1972, denominada árvore binária simétrica.



# ÁRVORE VERMELHO- PRETO

- Posteriormente em um trabalho de Leonidas J. Guibas e Robert Sedgwick, em 1978 adquiriu o seu nome atual.
- Nesta estrutura cada nó possui, além das informações já conhecidas em uma árvore binária, um atributo de cor, que pode ser vermelho ou preto.



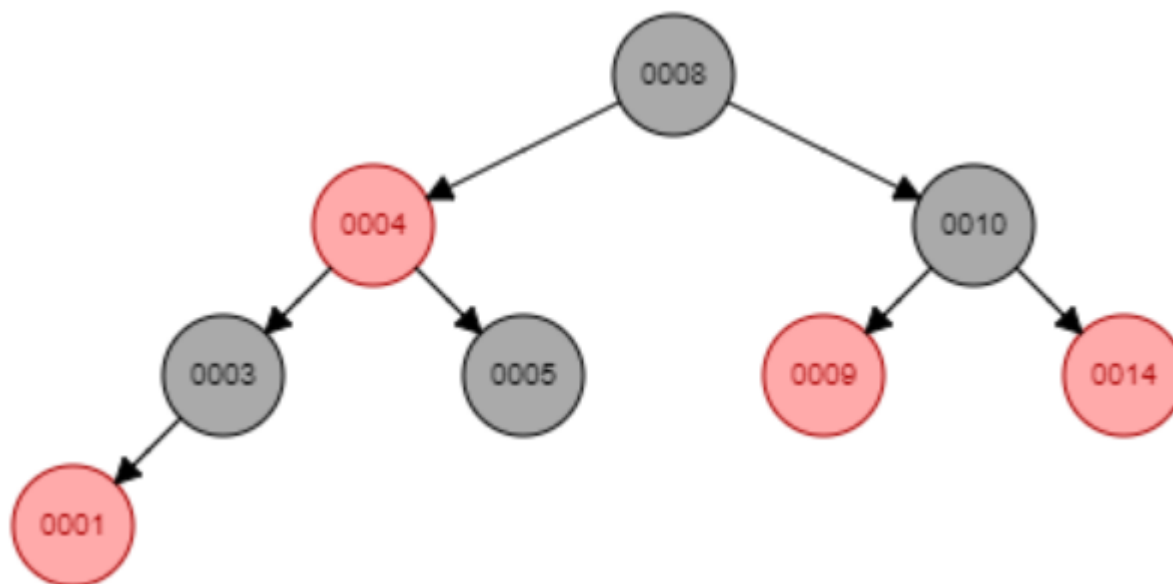
# ÁRVORE VERMELHO- PRETO

- Propriedades desejadas:
  - Todo nó da árvore é vermelho ou preto;
  - A raiz é sempre preta;
  - Se um nó é vermelho, então os seus filhos são pretos não existem nós vermelhos consecutivos;
  - Para cada nó, todos os caminhos deste nó para os nós folhas contêm o mesmo número de nós pretos.
  - Todo nó folha é preferencialmente preto;



# ÁRVORE VERMELHO- PRETO

- Exemplo





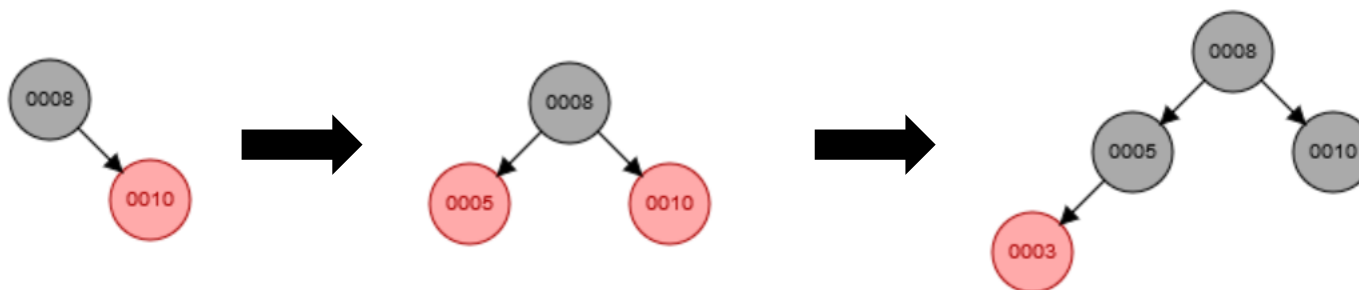
# BALANCEAMENTO

- Para manter o balanceamento, a árvore vermelho-preto faz uso de rotações e ajustes de cores.
- Por meio dessas operações a árvore busca manter-se como uma árvore binária quase completa.
- O objetivo das operações de rotação e de ajuste de cores é garantir que suas propriedades não sejam violadas.



# BALANCEAMENTO

- Durante o balanceamento da árvore, pode ser necessário mudar a cor de um nó e de seus filhos, isso será necessário para garantir as propriedades deste tipo de árvore.





# BALANCEAMENTO

- As operações de rotação na árvore vermelho-preto são similares as rotações da árvore AVL.
- O cuidado principal é administrar a cor de cada nó para manter a integridade das propriedades da árvore vermelho-preto.





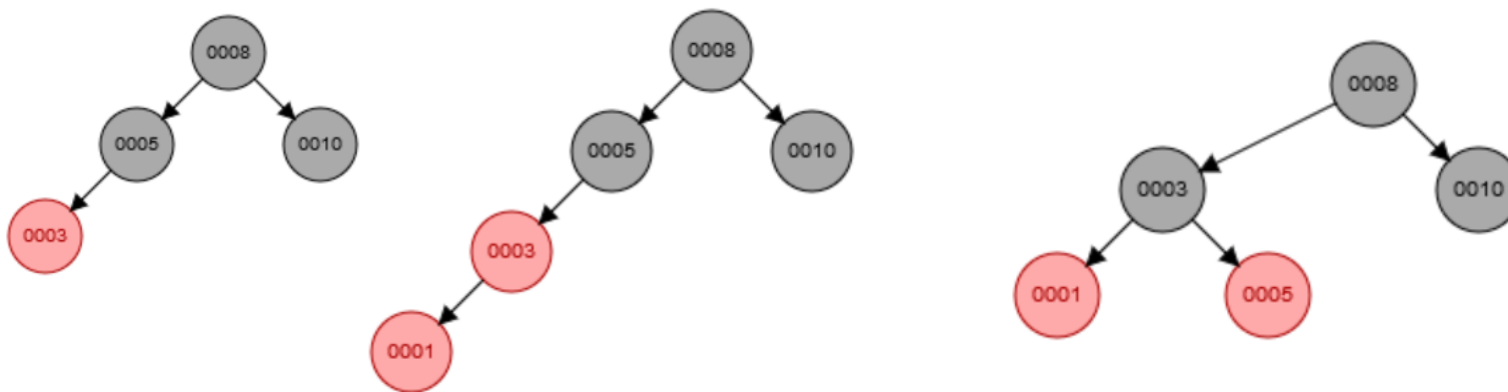
# BALANCEAMENTO

- A rotação é uma operação que, dado um conjunto de três nós, visa deslocar um nó vermelho que esteja à esquerda para a direita, ou vice-versa.
- Dependendo do caso, podemos usar uma rotação à esquerda ou uma rotação à direita.



# BALANCEAMENTO

- Rotação para a direita:

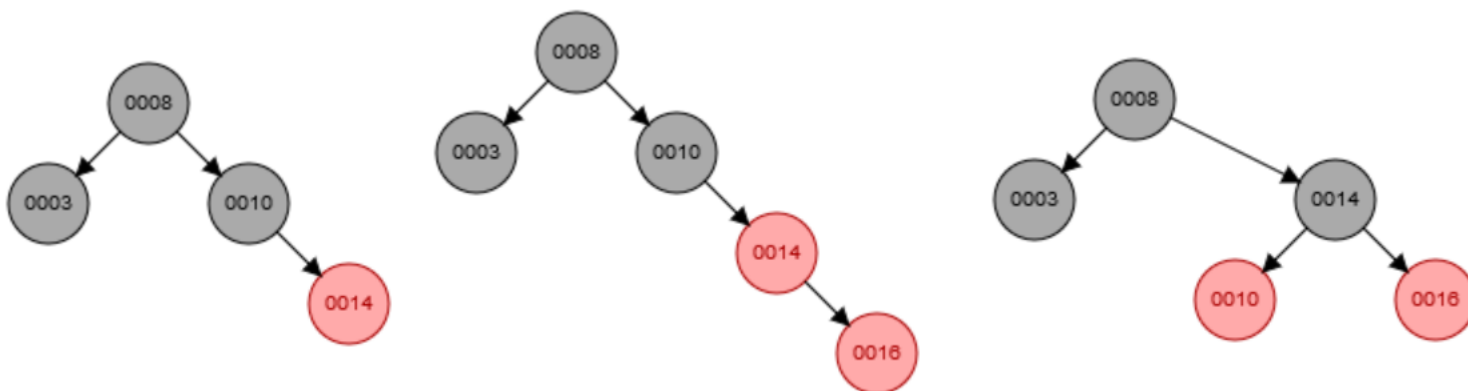


- Se um nó é vermelho, então os seus filhos são pretos não existem nós vermelhos consecutivos;



# BALANCEAMENTO

- Rotação para a esquerda:

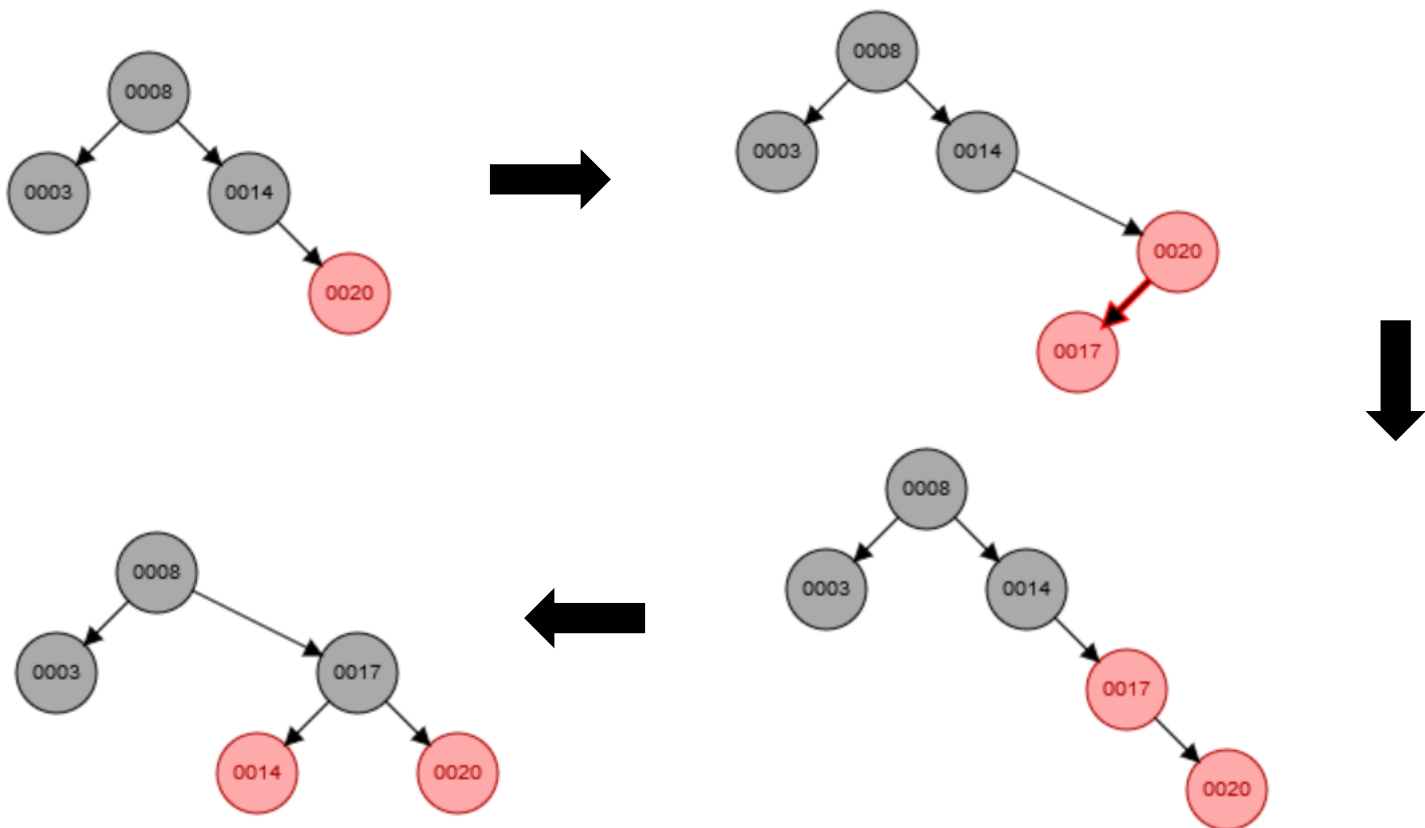


- Se um nó é vermelho, então os seus filhos são pretos não existem nós vermelhos consecutivos;



# BALANCEAMENTO

- Necessidade de duas rotações:





# INSERÇÃO DE UM NÓ

- A inserção de um novo nó é uma tarefa similar à inserção na árvore AVL.
- Primeiramente devemos percorrer um conjunto de nós da árvore até chegar ao nó folha que irá se tornar pai do novo nó.
- Uma vez inserido o nó, devemos verificar se ocorreu alguma violação de alguma das propriedades da árvore e, se necessário, aplicar uma das rotações e/ou mudança de cores.



# REMOÇÃO DE UM NÓ

- Para remover um nó na árvore vermelho-preto, devemos encontrar o nó que será removido, o qual poderá ser um nó folha ou interno, sendo interno poderá ter um ou dois filhos, seguindo a regra das árvores binárias já estudado.



# AVL x VERMELHO- PRETO

- Apesar das duas árvores serem balanceadas, na prática a árvore AVL é mais rápida na operação de busca, e mais lenta nas operações de inserção e remoção.
- Isso se deve ao fato da árvore AVL ser mais balanceada do que as árvores Vermelho-preto, o que acelera a operação de busca.
- No entanto para manter um balanceamento mais rígido exige um custo maior nas operações de inserção e remoção.



# AVL x VERMELHO- PRETO

- Se uma aplicação realiza de forma intensa a operação de busca, recomenda-se a árvore AVL.
- Se uma aplicação necessita um volume maior de operações de inserção e exclusão, adote a árvore vermelho-preto.





# ATIVIDADES

- Execute as operações apresentadas abaixo em uma árvore vermelho-preto:

- Inserir valor 67
- Inserir valor 87
- Inserir valor 23
- Inserir valor 53
- Inserir valor 82
- Inserir valor 51
- Remove valor 87

- Inserir valor 90
- Inserir valor 17
- Inserir valor 31
- Inserir valor 52
- Inserir valor 60
- Remove valor 17
- Remove valor 23

- Após executar os passos da primeira coluna, apresente em ordem crescente os valores constantes nos nós vermelhos



# EXEMPLO

- Inserir valor 67, 87 e 23





# EXEMPLO

- Inserir valor 53

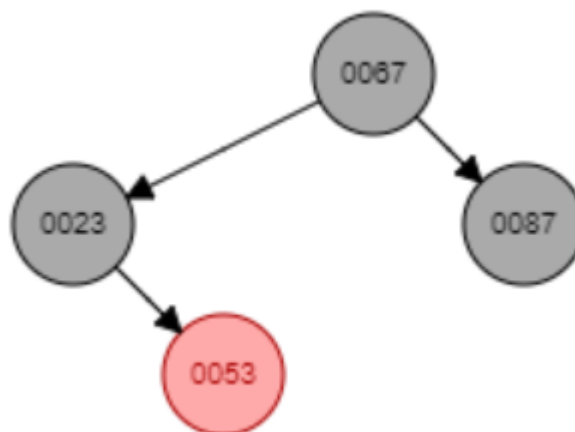


- Se um nó é vermelho, então os seus filhos são pretos  
não existem nós vermelhos consecutivos;
- Para cada nó, todos os caminhos deste nó para os nós  
folhas contêm o mesmo número de nós pretos.



# EXEMPLO

- Inserir valor 53

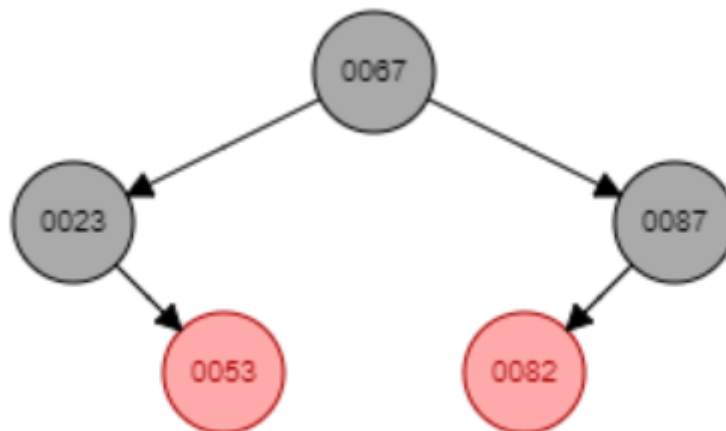


- Altera-se a cor do nó 23 e 87



# EXEMPLO

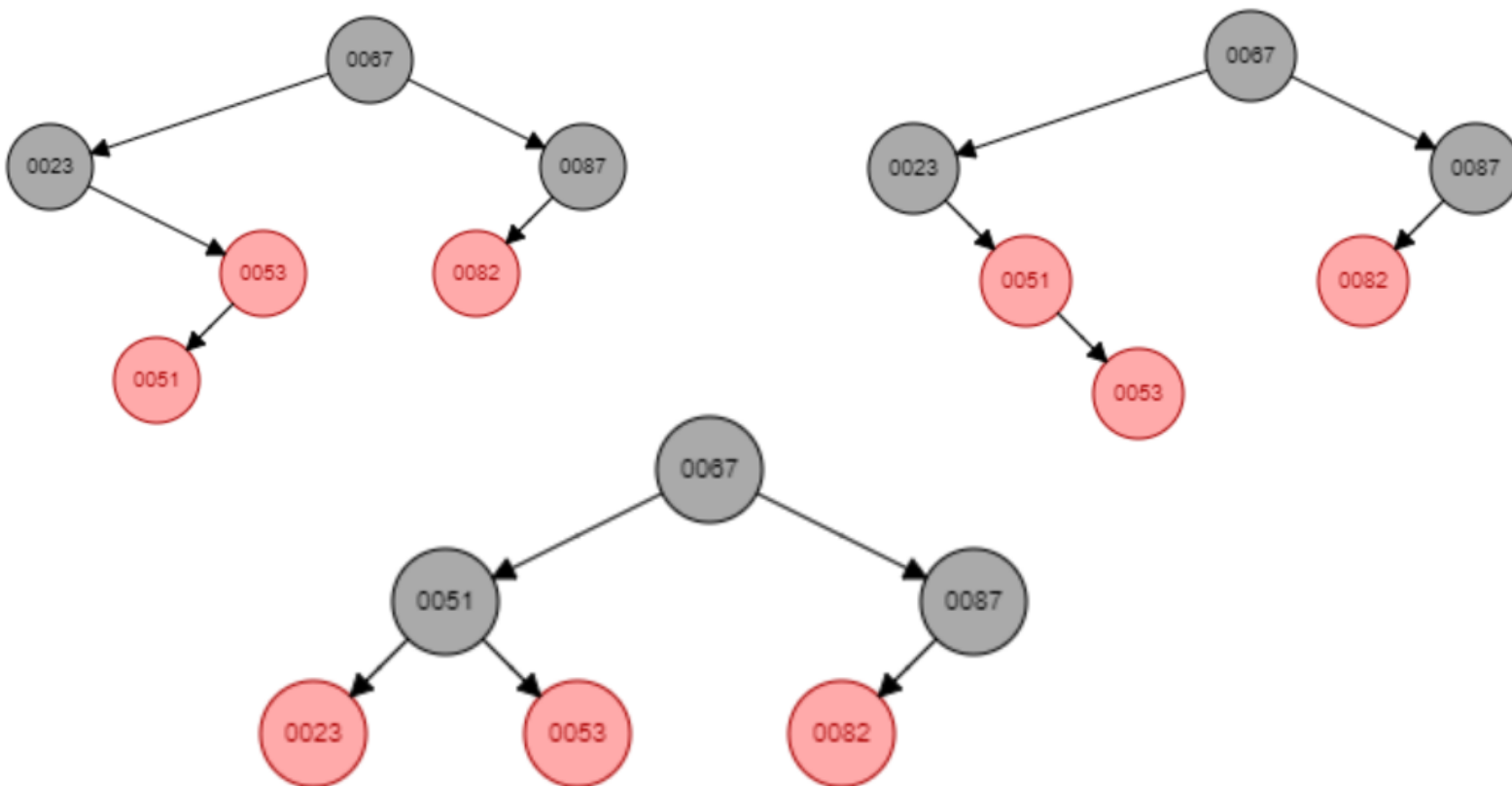
- Inserir valor 82





# EXEMPLO

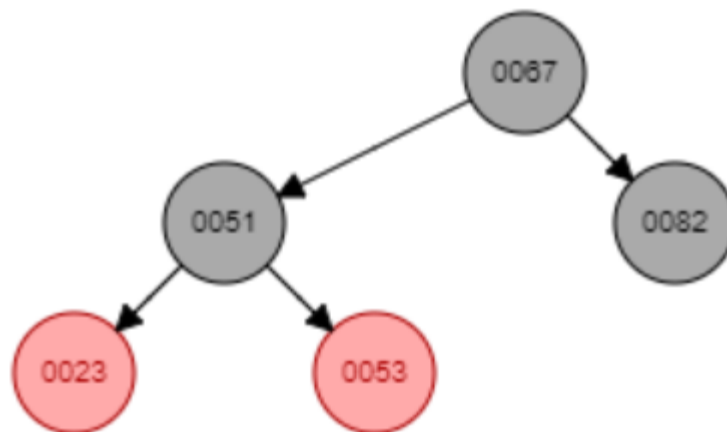
- Inserir valor 51





# EXEMPLO

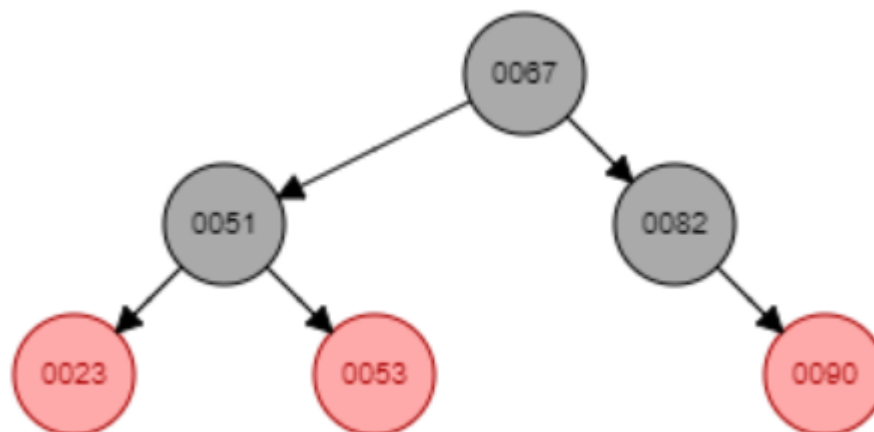
- Remove valor 87





# EXEMPLO

- Inserir valor 90

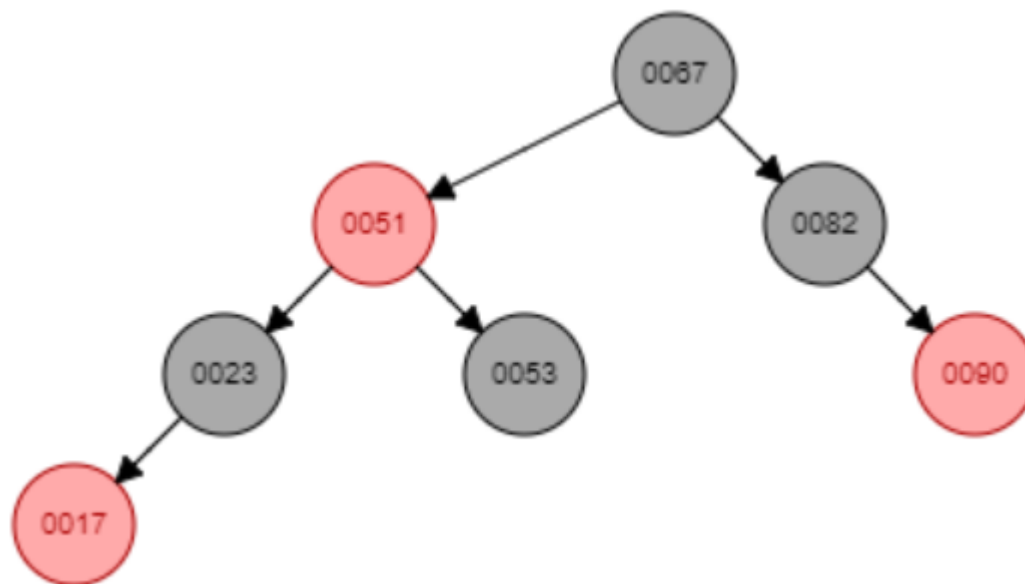






# EXEMPLO

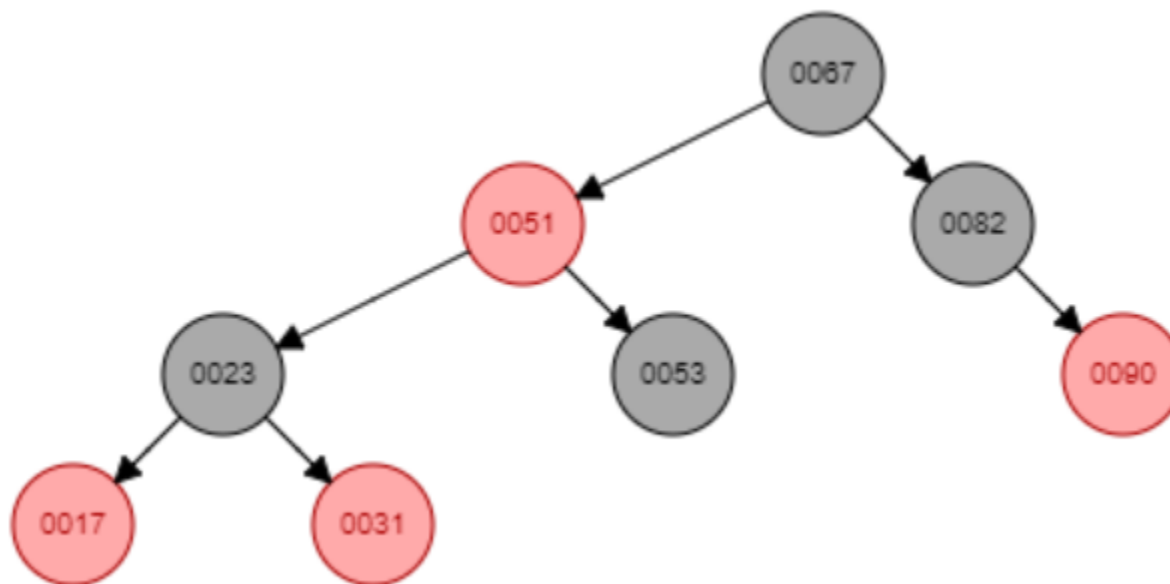
- Inserir valor 17





# EXEMPLO

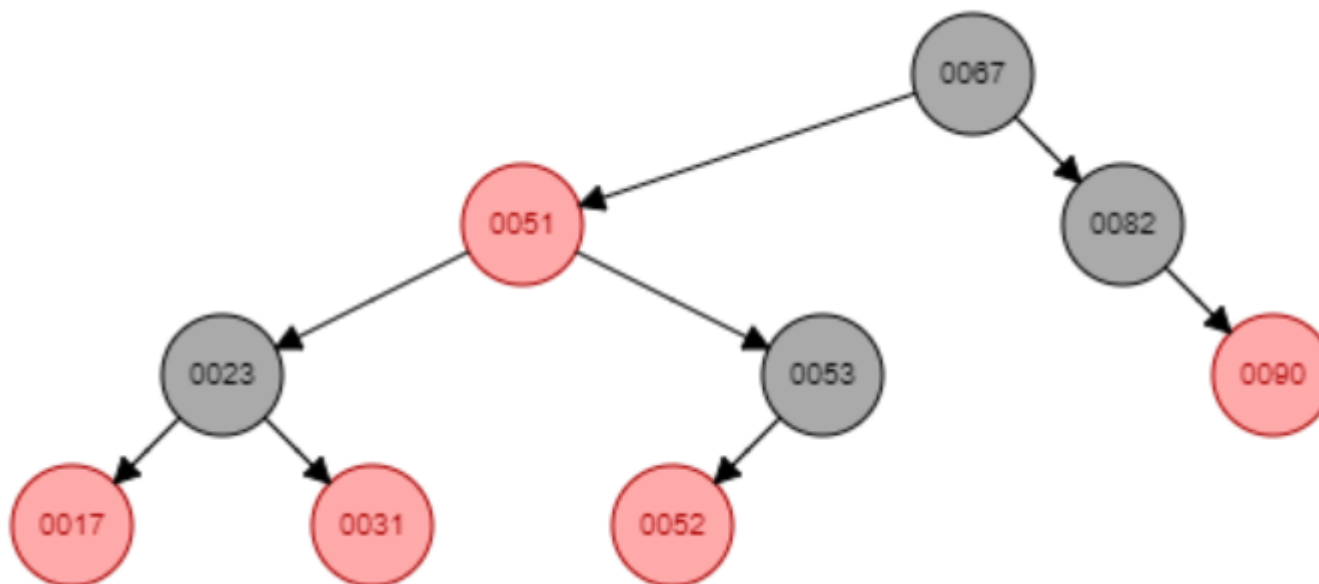
- Inserir valor 31





# EXEMPLO

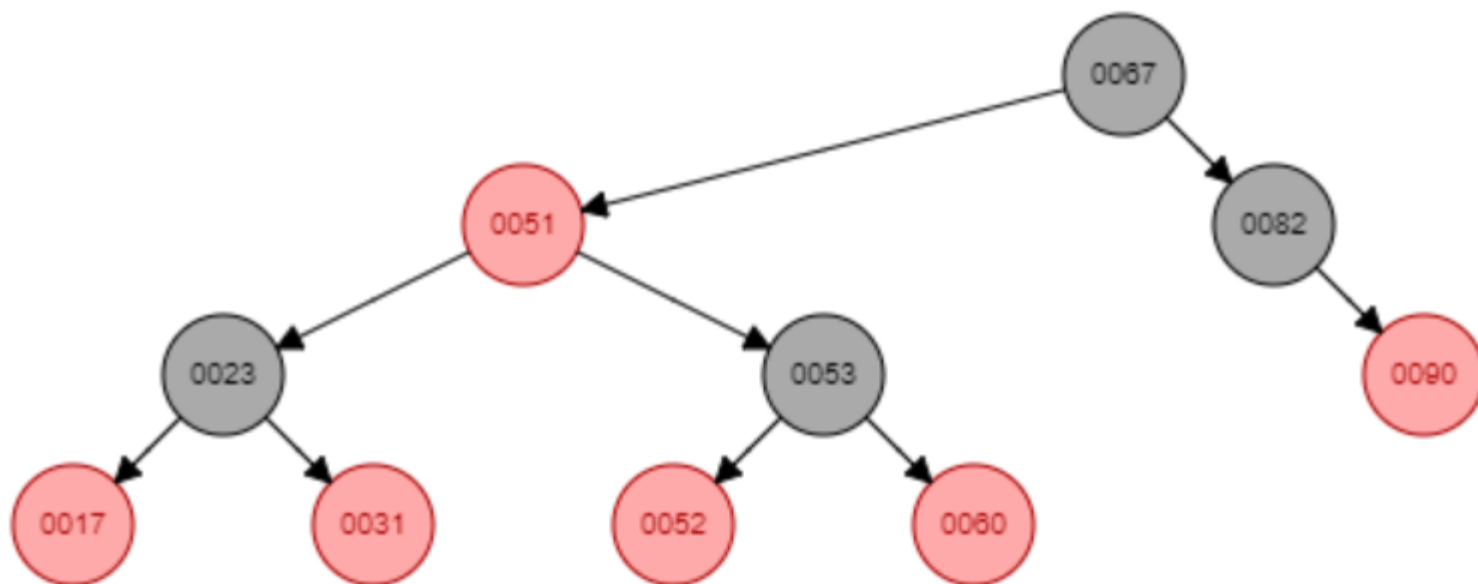
- Inserir valor 52





# EXEMPLO

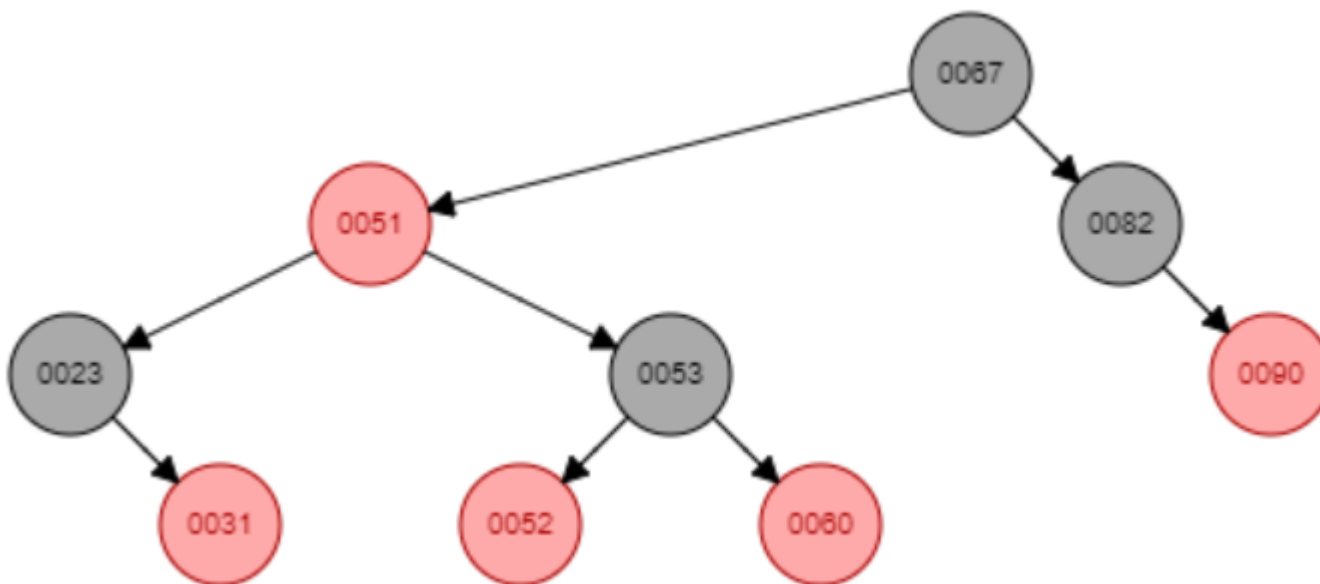
- Inserir valor 60





# EXEMPLO

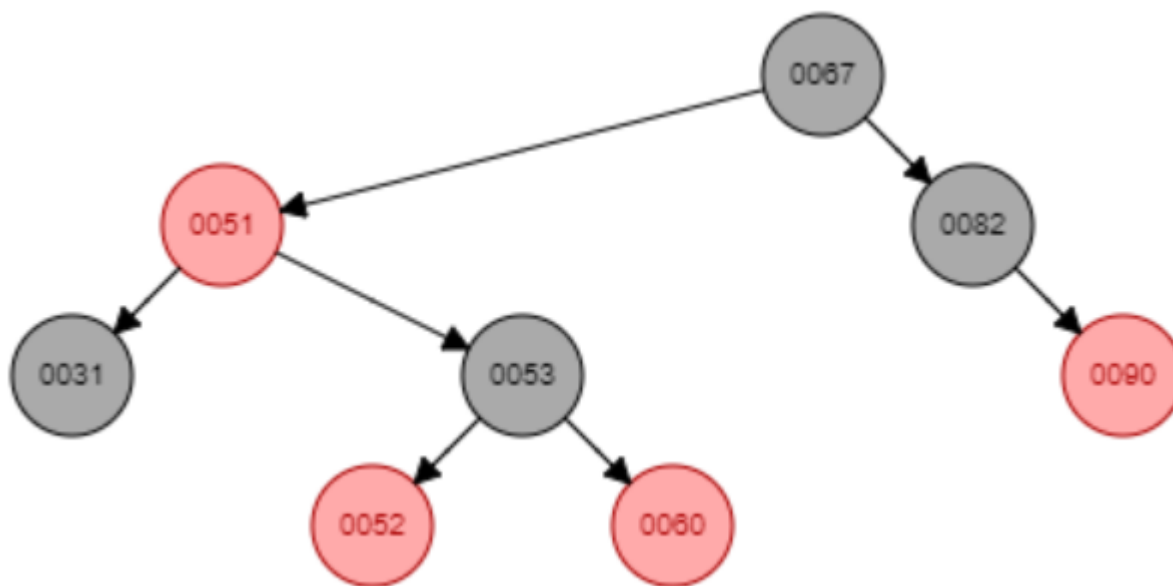
- Remover valor 17





# EXEMPLO

- Remover valor 23





# REFERÊNCIAS

- PEREIRA, Silvio do Lago. Estrutura de Dados Fundamentais: Conceitos e Aplicações, 12. Ed. São Paulo, Érica, 2008.
- BACKES, André Ricardo, Estrutura de dados descomplicada: em linguagem C, 1 Ed. – Rio de Janeiro: Elsevier, 2016.
- SENGER, H., Notas de Aula, Universidade de São Judas Tadeu, 1999.
- WALDEMAR Celes, Renato Cerqueira, José Lucas Rangel, Introdução a Estruturas de Dados, Editora Campus (2004).
- VELOSO, Paulo. SANTOS, Celso dos. AZEVEDO, Paulo. FURTADO, Antonio. Estrutura de dados. Rio de Janeiro: Ed. Elsevier, 1983 27ª reimpressão.
- <https://www.cs.usfca.edu/~galles/visualization/Algorithms.html>