

Algoritmos e Estrutura de Dados II

Prof. Fellipe Guilherme Rey de Souza

Aula 15 –Árvore Rubro-Negra

Agenda

- Definição
- Regras
- Balanceamento
- Inserção

PS: Parte do conteúdo retirado do material do Prof. Flávio B. Gonzaga

Definição

- A árvore Rubro-Negra é uma árvore auto balanceável (semelhante à AVL), mas que permite um grau maior de desbalanceamento.
 - Isto significa que a regra para realizar rotações dentro da árvore Rubro-Negra é mais flexível se comparado à árvore AVL.
 - Com isso, faremos menos rotações. Mas, em contrapartida, a árvore ficará mais desbalanceada.

Definição

- Isso faz com que geralmente a árvore rubro-negra seja mais adequada à situações onde existam muitas operações de inserção e remoção.
- E portanto, a árvore AVL acaba sendo mais indicada em situações onde existam muitas operações de leitura.

Definição

- Desvantagens da árvore AVL em relação à rubro-negra:
 - i. Mais rotações em inserções e deleções
 - ii. Custo maior de manutenção do balanceamento
 - iii. Desempenho inferior em inserções e deleções frequentes
 - iv. Maior complexidade de implementação
 - v. Menor adaptabilidade a operações em lote (bulk operations)

Definição

- Desvantagens da árvore AVL em relação à rubro-negra:
 - i. **Mais rotações em inserções e deleções**
 - A AVL pode exigir mais rotações para reequilibrar a árvore após inserções e, especialmente, deleções.
 - Exemplo: AVL pode fazer até 2 rotações, enquanto rubro-negra geralmente faz no máximo 1.

Definição

- Desvantagens da árvore AVL em relação à rubro-negra:
 - ii. **Custo maior de manutenção do balanceamento**
 - A manutenção do fator de balanceamento (diferença de altura entre subárvores) é mais rigorosa na AVL.
 - Isso significa mais verificações e atualizações durante operações de modificação.

Definição

- Desvantagens da árvore AVL em relação à rubro-negra:
 - iii. Desempenho inferior em inserções e deleções frequentes**
 - Como consequência do balanceamento mais rigoroso, em cenários com muitas atualizações, a árvore rubro-negra tende a ter melhor desempenho.
 - É por isso que estruturas como o TreeMap do Java e o std::map do C++ usam rubro-negras.

Definição

- Desvantagens da árvore AVL em relação à rubro-negra:
 - iv. Maior complexidade de implementação**
 - A lógica de balanceamento da AVL é mais detalhada, exigindo mais casos a serem tratados em código.
 - Isso pode tornar a árvore AVL mais propensa a erros e mais difícil de manter.

Definição

- Desvantagens da árvore AVL em relação à rubro-negra:
 - v. **Menor adaptabilidade a operações em lote (bulk operations)**
 - Em aplicações como bancos de dados ou sistemas de arquivos, onde dados são frequentemente inseridos em grandes quantidades, a flexibilidade da rubro-negra é mais vantajosa.

Definição

- Apesar dessas desvantagens, a AVL é **mais eficiente para buscas**.
 - Especialmente em árvores com muitos nós e menos modificações, pois garante uma altura menor que a rubro-negra!

Regras

- Na Árvore AVL, a necessidade de rebalanceamento se dava quanto algum(ns) do(s) nó(s) da árvore tinham um fator de balanceamento diferente dos três permitidos: -1 , 0 e $+1$.
- Para a Árvore Rubro-Negra, nós não temos um fator de balanceamento. No entanto, temos um conjunto de 6 regras que devem ser seguidas.
 - Seguir essas regras é o que garante o balanceamento da árvore rubro-negra!

Regras

- As regras a serem seguidas da árvore rubro-negra são:
 1. Todo nó é **rubro** (rubro) ou **preto**.
 2. Raiz é sempre **negra**.
 3. Novo nó é sempre **rubro**.
 4. Todo caminho da raiz até algum nó folha terá o mesmo número de nós **negros**.
 5. Nenhum caminho pode ter dois nós **rubros** consecutivos.
 6. NULL é considerado **negro**.

Regras

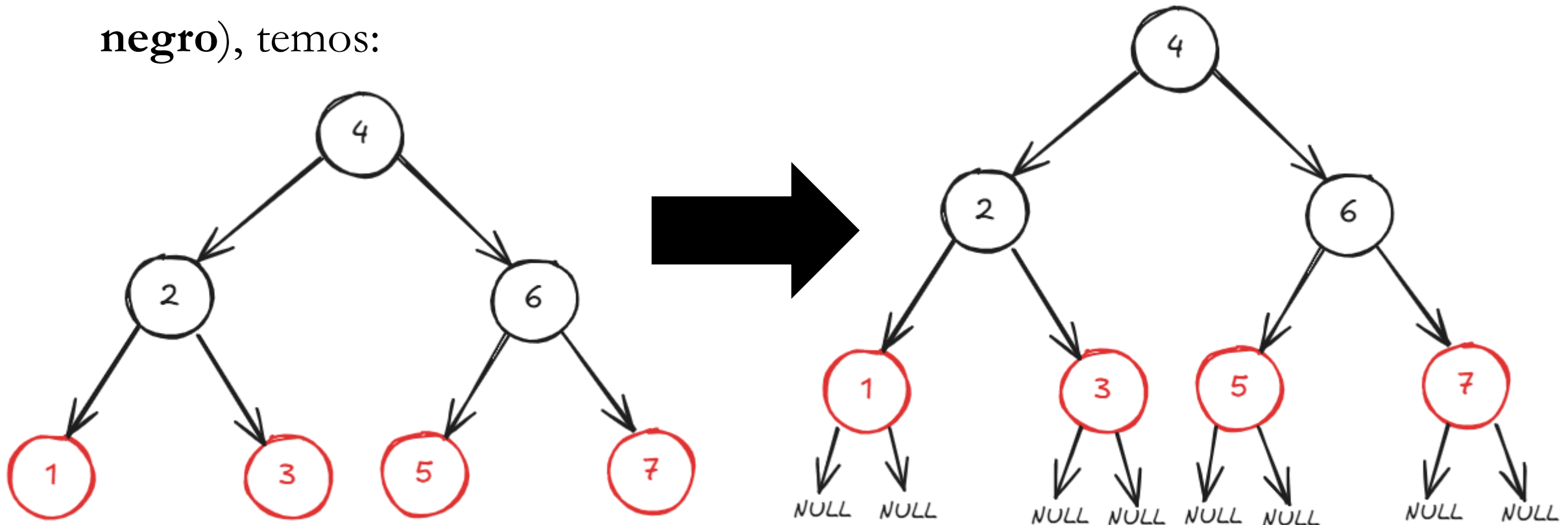
Definição

→ **Regras**

Balanceamento

Inserção

- Sobre as Regras 3 (Novo nó é sempre **rubro**) e 6 (NULL é considerado **negro**), temos:



Regras

- Diferentemente da árvore binária de busca e árvore AVL, na árvore rubro-negra nós precisamos sempre nos lembrar que um nó sem filhos (ou somente com um filho) possui um ou dois filhos que são NULL.
- Por questão de simplicidade na representação, iremos omitir os ponteiros para NULL nos slides.
 - Lembrem-se de que eles serão importantes para a correta aplicação das regras!

Balanceamento

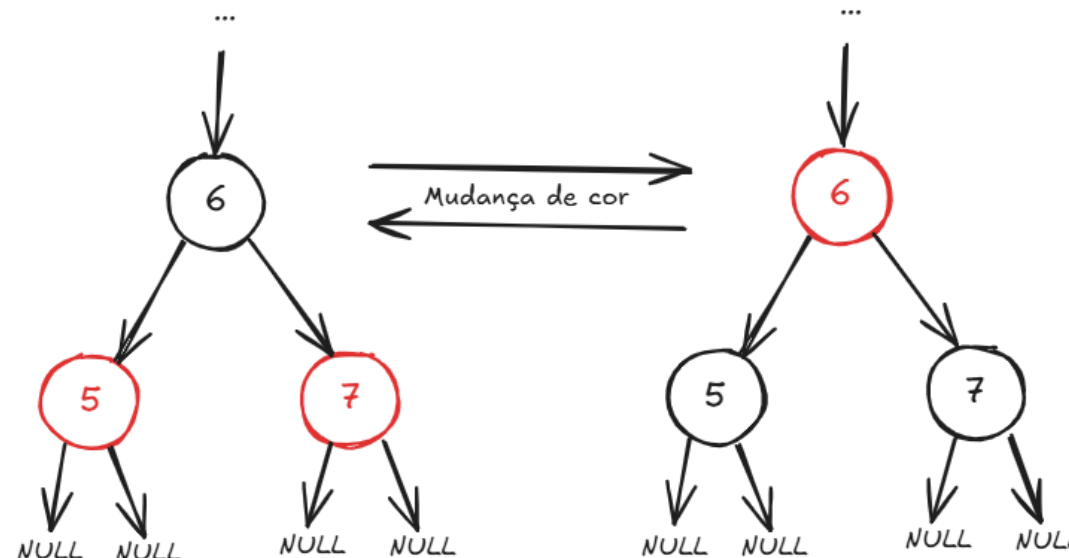
- O balanceamento na árvore Rubro-Negra é feita através de rotações e ajuste de cores a cada inserção ou remoção.
- Os benefícios do balanceamento são:
 - Manter o equilíbrio da árvore
 - Corrigir possíveis violações de suas propriedades
 - Custo máximo de qualquer algoritmo de balanceamento será de $O(\log n)$

Balanceamento

- Durante o balanceamento da árvore, podemos ter a necessidade de mudarmos a cor de um nó e de seus filhos de **rubro** para **negro** (e vice-versa).
- A operação de mudança de cor **não altera** o número de nós pretos da raiz até os nós folhas.

Balanceamento

- No entanto, a operação de mudança de cor **pode introduzir** dois nós rubros de forma consecutiva na árvore.
 - Caso tenhamos esse cenário, ele poderá ser corrigido com outras operações.



Balanceamento

- Enquanto a *Árvore AVL* utiliza quatro funções de rotação para manter a árvore balanceada, a *Árvore Rubro-Negra* utiliza somente duas funções:
 - Rotação à esquerda
 - Rotação à direita
- Dado um conjunto de três nós, visa deslocar um nó **rubro** que esteja à esquerda para a direita (e vice-versa)

Balanceamento

- A Rotação à Esquerda recebe um nó A que tem um outro nó, B , como filho direito.
- A rotação move B para o lugar de A . Assim, A se torna o filho à esquerda de B .
- B recebe a cor de A , enquanto A fica **rubro**.

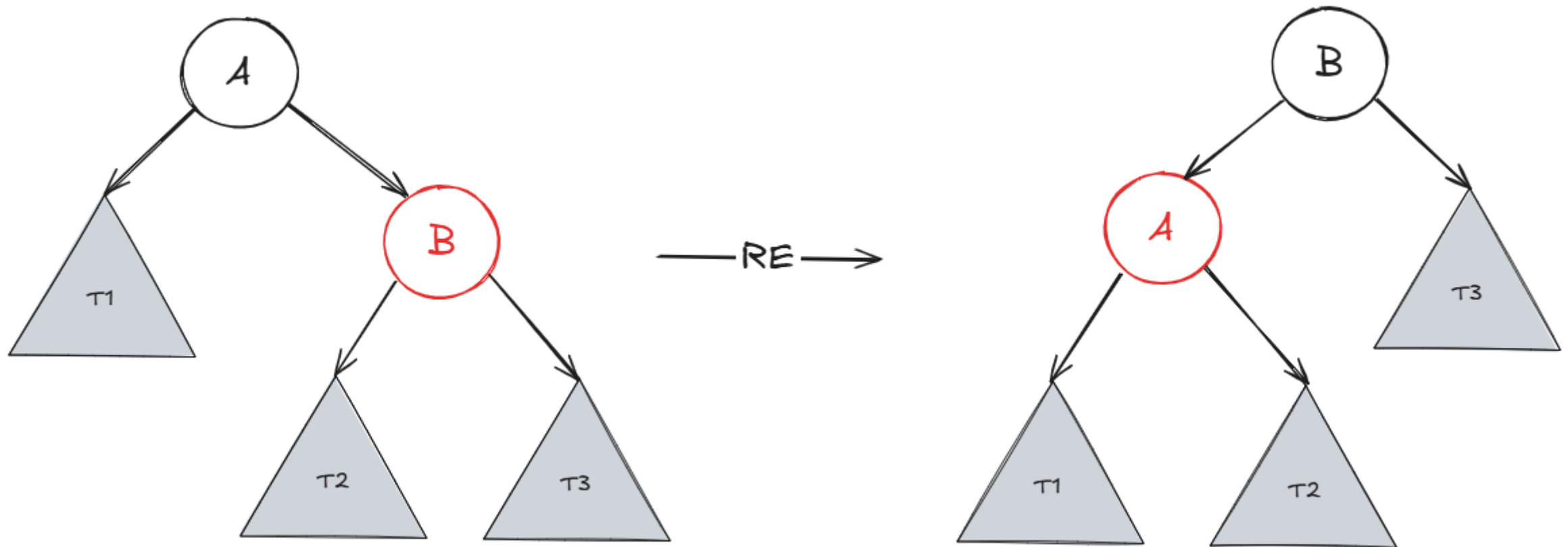
Balanceamento

Definição

Regras

→ **Balanceamento**

Inserção



Balanceamento

- A Rotação à Direita recebe um nó A que tem um outro nó, B, como filho esquerdo.
- A rotação move B para o lugar de A. Assim, A se torna o filho à direita de B.
- B recebe a cor de A, enquanto A fica **rubro**.

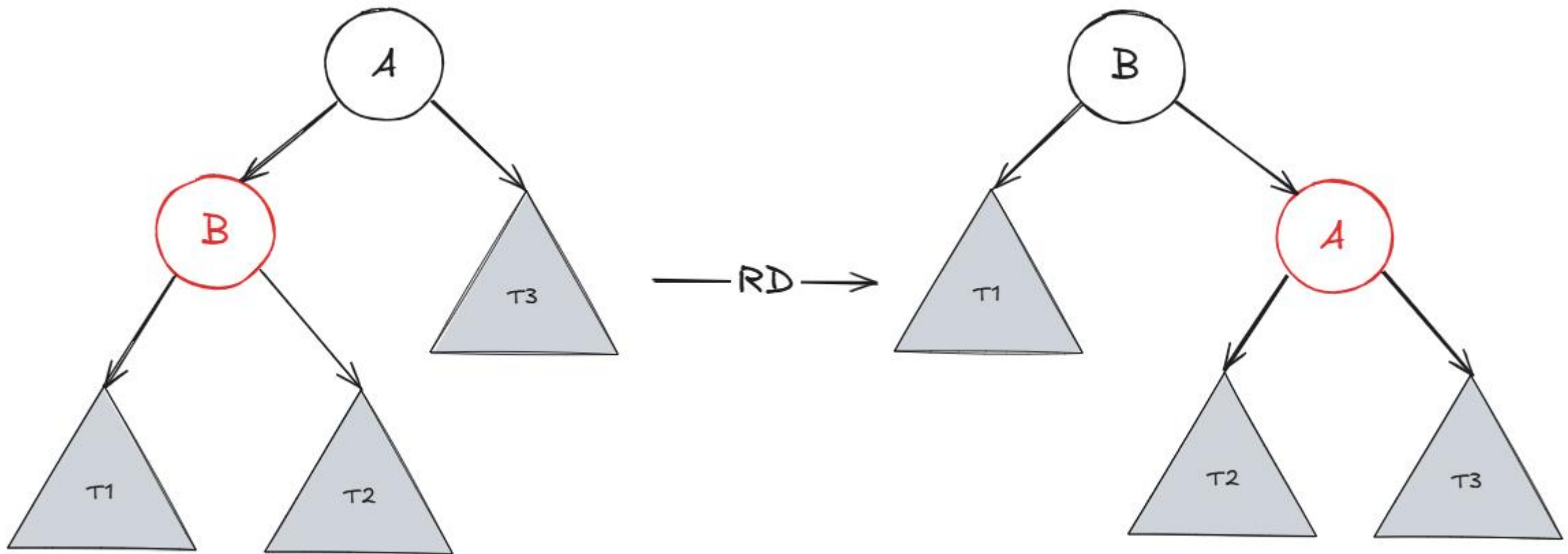
Balanceamento

Definição

Regras

→ **Balanceamento**

Inserção



Balanceamento

- Quando devemos recolorir a árvore (sem utilizar rotação)?
- Quando o nó inserido é **rubro** e seu pai também é **rubro**.
 - Isso viola a regra 5 (não pode haver dois nós **rubros** consecutivos).

Balanceamento

- Quando devemos recolorir a árvore (sem utilizar rotação) (*cont.*)?
- Se o tio do nó também for **rubro**, então:
 - Precisamos recolorir o pai e o tio para **negro**.
 - Recolore o avô para **rubro**.
 - Em seguida, verifica se isso criou um novo problema "para cima" na árvore (pode precisar continuar o ajuste).

Balanceamento

- Quando devemos rotacionar?
- Basicamente, existem duas situações onde uma rotação é necessária:
 - i. Dois nós **rubros** consecutivos aparecem (pai e filho são **rubros**) MAS o tio é **negro**.
 - ii. Desequilíbrio de altura após uma inserção/remoção (isto afeta a "altura **negra**", ou seja, o número de nós **negros** em cada caminho).

Inserção

- Para exemplificarmos como será feita a recoloração e rotação na árvore rubro-negra, faremos a inserção dos seguintes elementos:

3 – 1 – 5 – 7 – 6 – 8 – 9 – 10

Inserção

Definição

Regras

Balanceamento

→ **Inserção**

3 – 1 – 5 – 7 – 6 – 8 – 9 – 10

- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;
- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Inserção

Definição

Regras

Balanceamento

→ **Inserção**

3 – 1 – 5 – 7 – 6 – 8 – 9 – 10

3

De qual cor será o primeiro elemento (3)?

- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;
- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:
- Após rotacionar:



Inserção

Definição

Regras

Balanceamento

→ **Inserção**

3 – 1 – 5 – 7 – 6 – 8 – 9 – 10

3

- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- • Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;
- Após rotacionar:
 - Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
 - Após trocar cores:



Inserção

Definição

Regras

Balanceamento

→ **Inserção**

3 – 1 – 5 – 7 – 6 – 8 – 9 – 10

3

- Todo nó é **rubro** ou negro;
- • Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Inserção

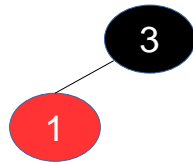
Definição

Regras

Balanceamento

→ **Inserção**

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- • Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Inserção

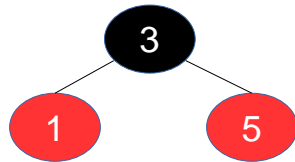
Definição

Regras

Balanceamento

→ **Inserção**

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- • Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Inserção

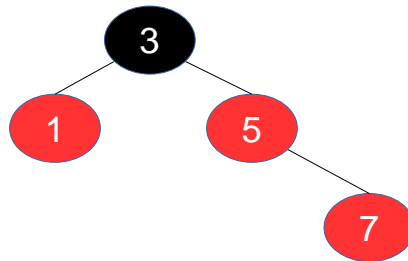
Definição

Regras

Balanceamento

→ **Inserção**

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



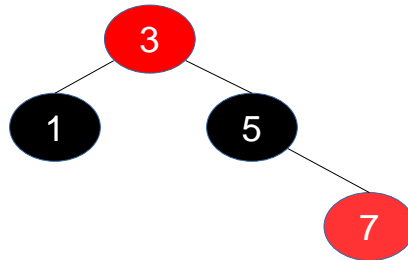
- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- • Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- • Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;
- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
 - Após trocar cores:
- Após rotacionar:



7 é rubro. Pai 5 é rubro → violação: dois rubros seguidos.
Tio (irmão de 5) é 1R → caso de tio rubro.

Inserção

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
 - Após trocar cores:

- Após rotacionar:



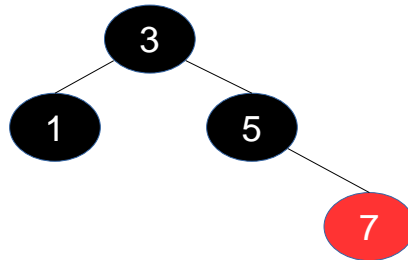
Recoloração:

Pai 5 e tio 1 → ficam pretos

Avô 3 → fica rubro

Inserção

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;
- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
 - Após trocar cores:
- Após rotacionar:

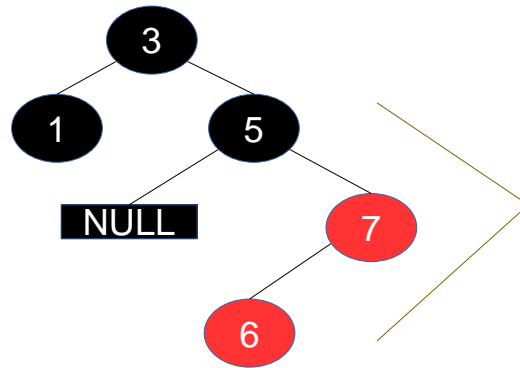


Recoloração:

Mas 3 é raiz → volta a ser preto

Inserção

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;

Nenhum caminho pode ter dois nós **rubros** consecutivos;

NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



6 é rubro. Pai 7 é rubro → violação: dois rubros seguidos.

Tio (nulo) → preto

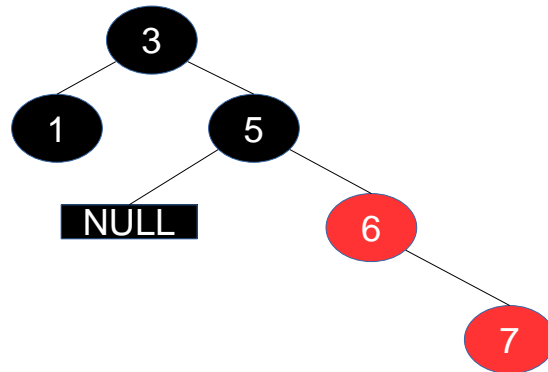
Forma um triângulo (inserido na subárvore esquerda do filho direito):

→ Fazer rotação direita em 7, depois rotação esquerda em 5

→ Depois, recoloração

Inserção

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
 - Após trocar cores:

- Após rotacionar:

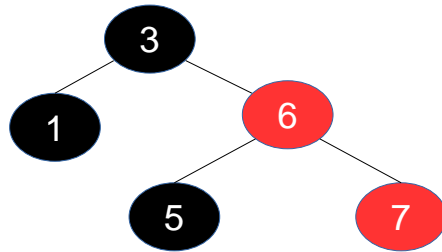


Aplicação:

Rotação direita entre 7 e 6 → 6 sobe

Inserção

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
 - Após trocar cores:

- Após rotacionar:



Aplicação:

Rotação esquerda entre 5 e 6 → 6 sobe de novo
6 vira pai, 5 e 7 filhos

Inserção

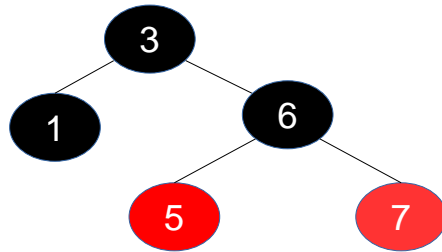
Definição

Regras

Balanceamento

→ **Inserção**

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Recoloração:
6 fica preto
5 e 7 rubros

Inserção

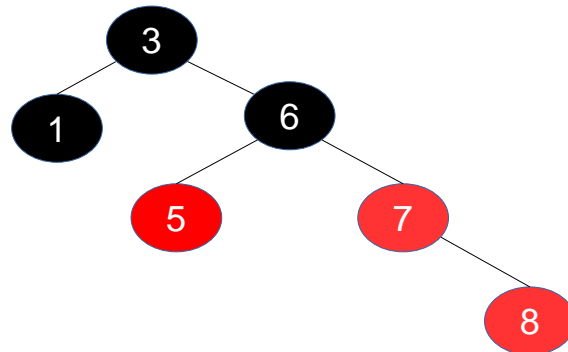
Definição

Regras

Balanceamento

→ **Inserção**

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



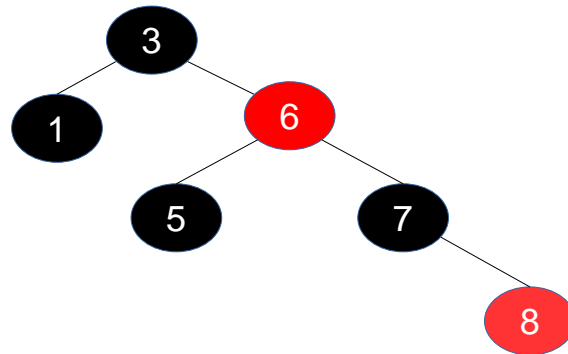
8 é rubro.

Pai 7 é rubro → violação

Tio 5 é rubro → caso de tio rubro

Inserção

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



Recoloração:

5 e 7 ficam pretos

6 vira rubro

3 continua preto → sem mais violações.

- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- **NULL** é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Inserção

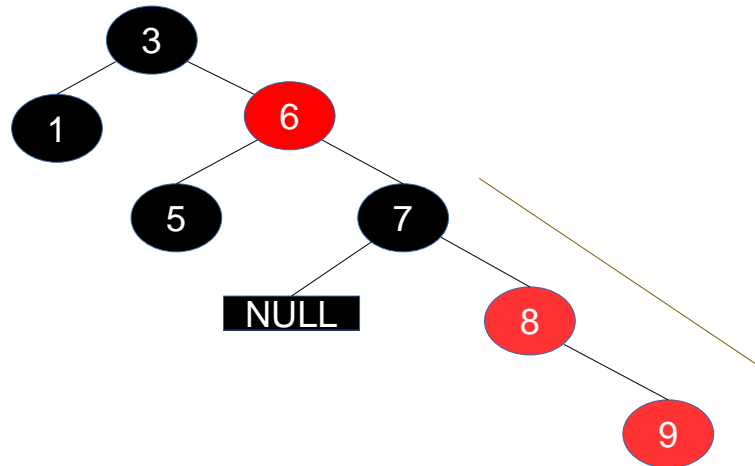
Definição

Regras

Balanceamento

→ Inserção

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



9 é rubro. Pai 8 é rubro → violação
Tio é nulo (preto)
Forma linha reta à direita (pai 8, avô 7):
→ Rotação esquerda em 7, seguida de recoloração

- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- **NULL** é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Inserção

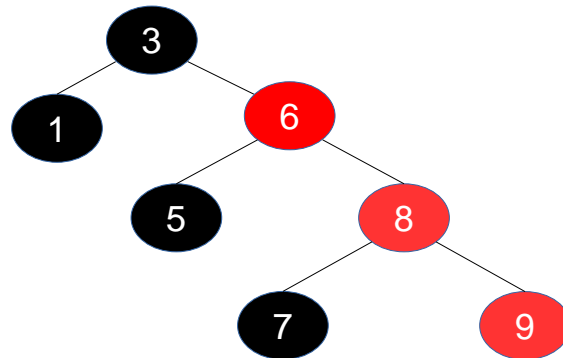
Definição

Regras

Balanceamento

→ **Inserção**

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;
- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Rotação:

8 sobe, 7 vira filho esquerdo de 8

Inserção

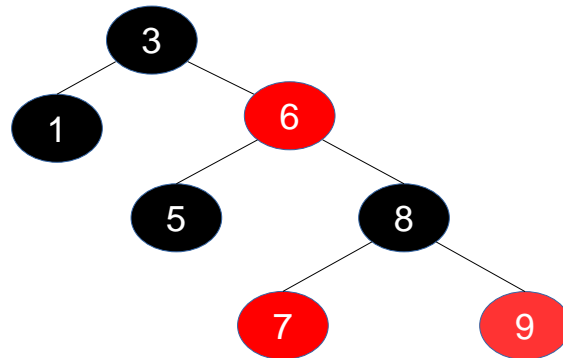
Definição

Regras

Balanceamento

→ **Inserção**

3 - 4 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Recoloração:
8 preto, 7 rubro

Inserção

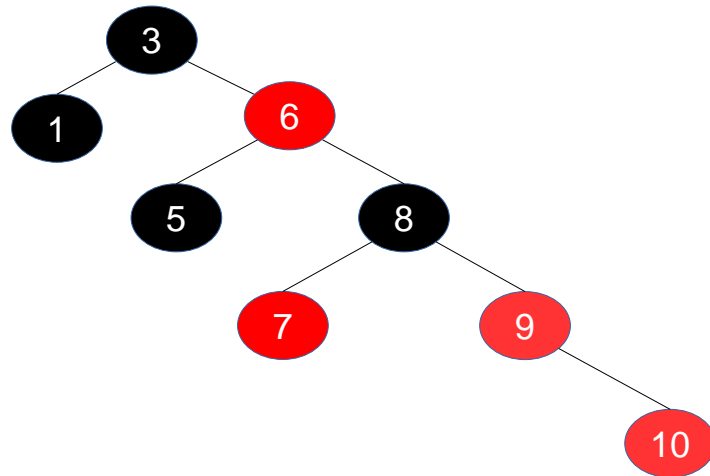
Definição

Regras

Balanceamento

→ **Inserção**

3 - 1 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



10 é rubro. Pai 9 é rubro → violação
Tio 7 também rubro → caso de tio rubro

Inserção

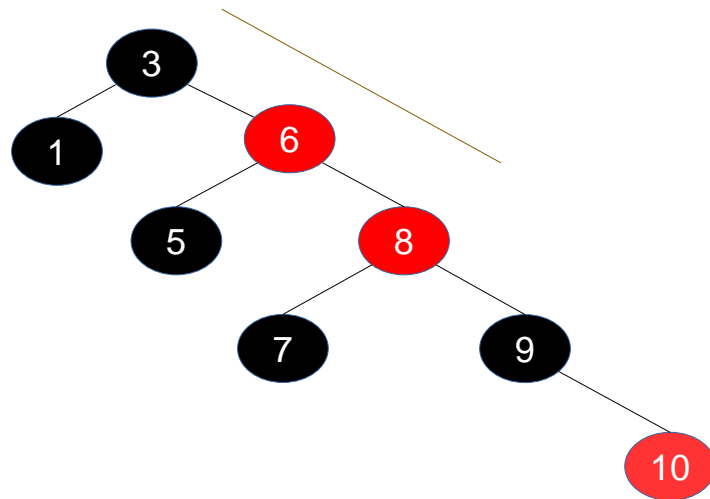
Definição

Regras

Balanceamento

→ **Inserção**

3 - 1 - 5 - 7 - 6 - 8 - 9 - 10



Recoloração:

7 e 9 viram pretos

8 vira rubro

6 é rubro → agora dois rubros (6 e 8) → violação

- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- **NULL** é considerado negro;

• Operações:

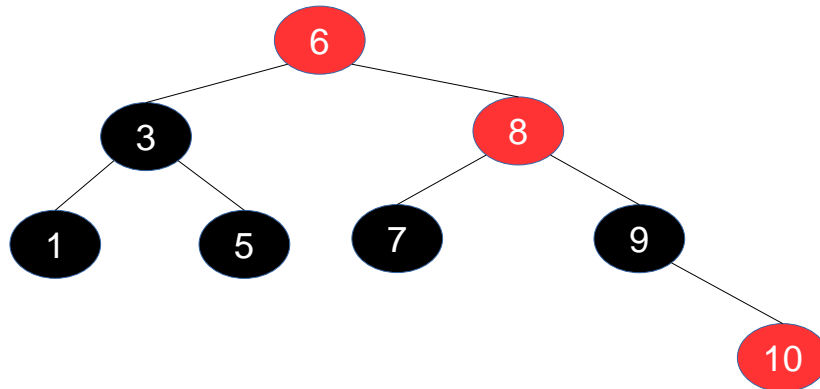
- Tio negro: rotaciona;
- Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



Inserção

3 - 1 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

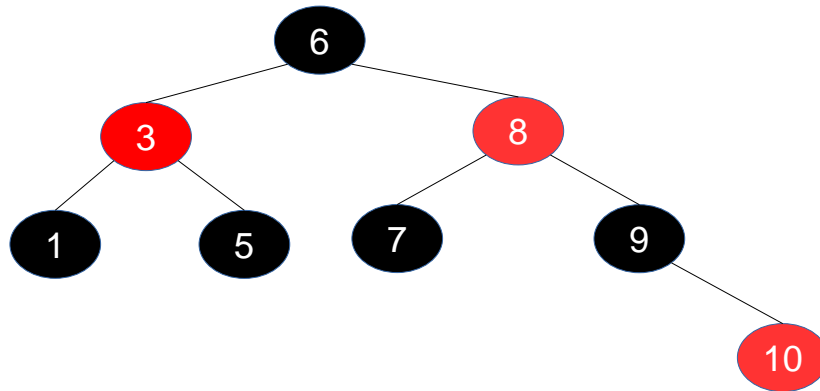
- Após rotacionar:



Agora, o pai 6 e o tio 1 são pretos (já era), e o avô 3 é preto
Aplicar rotação esquerda em 3, com 6 subindo

Inserção

3 - 1 - 5 - 7 - 6 - 8 - 9 - 10



- Todo nó é **rubro** ou negro;
- Raiz é sempre negra;
- Novo nó é sempre **rubro**;
- Todo caminho da raiz até algum nó folha terá o mesmo número de nós negros;
- Nenhum caminho pode ter dois nós **rubros** consecutivos;
- NULL é considerado negro;

- Operações:
 - Tio negro: rotaciona;
 - Tio **rubro**: troca cores;
- Após trocar cores:

- Após rotacionar:



6 fica preto, 3 e 8 ficam rubros

Inserção

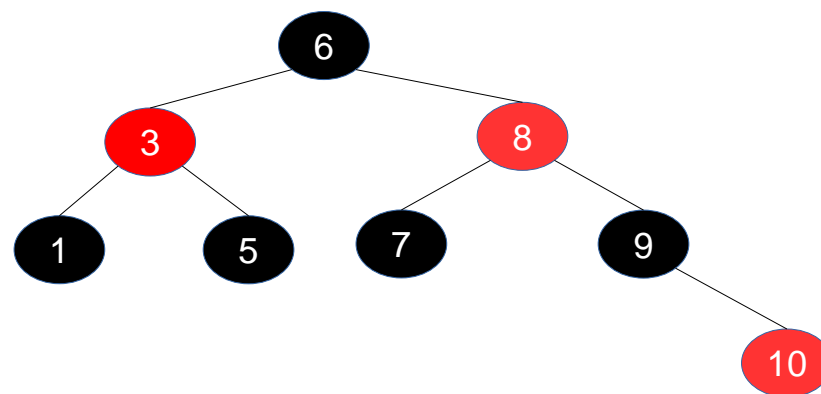
Definição

Regras

Balanceamento

→ **Inserção**

3 - 1 - 5 - 7 - 6 - 8 - 9 - 10



ÁRVORE FINAL

Algoritmos e Estrutura de Dados II

Prof. Fellipe Guilherme Rey de Souza

Aula 15 –Árvore Rubro-Negra