Árvore (Binária) de Pesquisa Balanceada

Vermelho-Preto (Red-Back)

Implementação com ligações simples

(usando ponteiros e memória dinâmica)

Introdução

Uma árvore Vermelho-Preto é

- Uma árvore binária de pesquisa
- Possui um campo extra para guardar a cor de cada nodo (**Vermelho** ou **Preto**).

Cada nodo é composto pelos seguintes campos

- Elemento (os “dados” a tratar)
- Cor (0/1 = **Vermelho**/**Preto**)
- Esquerda (ponteiro para a subárvore esquerda)
- Direita (ponteiro para a subárvore direita)
- Pai (ponteiro para o seu pai).

Por serem balanceadas,

- As suas operações de pesquisar, inserir e remover, tem ordem de complexidade, no pior caso, logaritmica: O(log(n)), em que n é o número de nodos da árvore.

Árvore Vermelho-Preta

Definição

- Uma árvore Vermelho-Preto está balanceada se satisfaz as seguintes condições

P1: Qualquer nodo é com **Vermelho** ou com **Preto**

P2: A raiz é um nodo com **Preto**

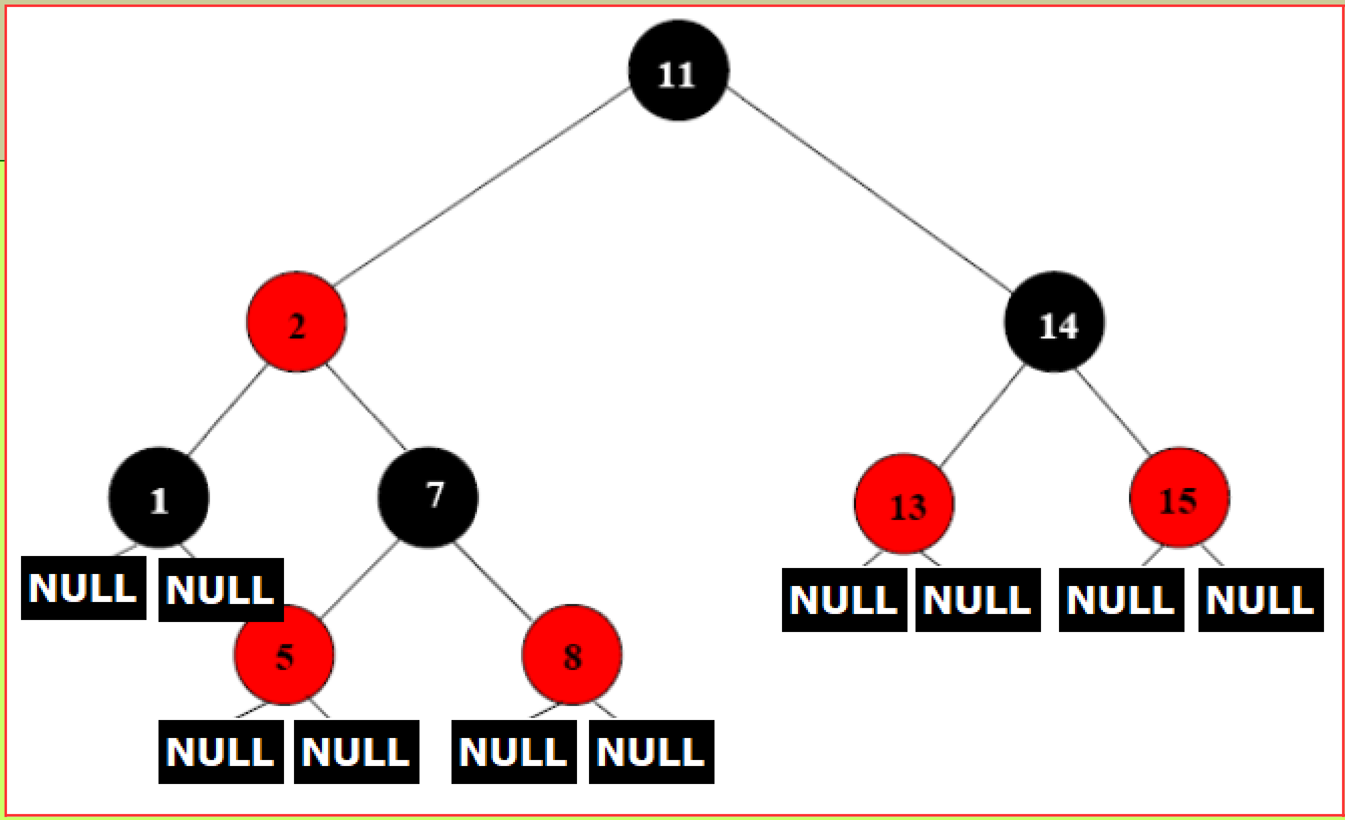
P3: Qualquer nodo que representa um subárvore vazia (NULL) é com **Preto**

P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

P5: Para cada nodo, todos os caminhos deste nodo para as folhas descendentes contém o mesmo número de nodos com **Preto**

- Numa árvore Vermelho-Preto balanceada todos os nodos estão balanceados
- Num caminho da raiz até qualquer subárvore vazia
  - **não** podem existir dois nodos com **Vermelho** consecutivos
  - logo, um nodo com **Vermelho** não pode ser filho de um nodo com **Vermelho**

Representação mais geral



P1: Qualquer nodo é com **Vermelho** ou com **Preto**

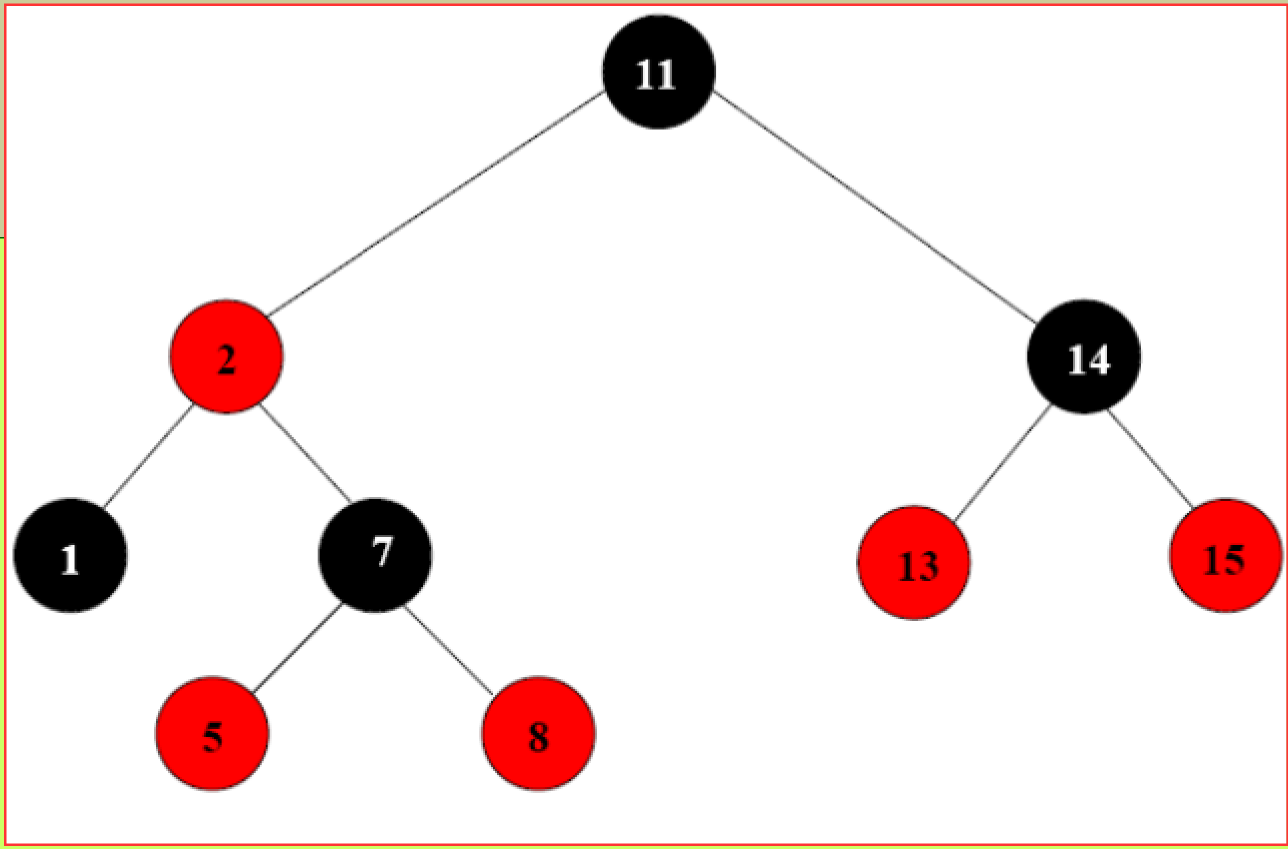
P2: A raiz é um nodo com **Preto**

P3: Qualquer nodo que representa um subárvore vazia (NULL) é com **Preto**

P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

P5: Para cada nodo, todos os caminhos deste nodo para as folhas descendentes contém o mesmo número de nodos com **Preto**

Representação mais simplificada



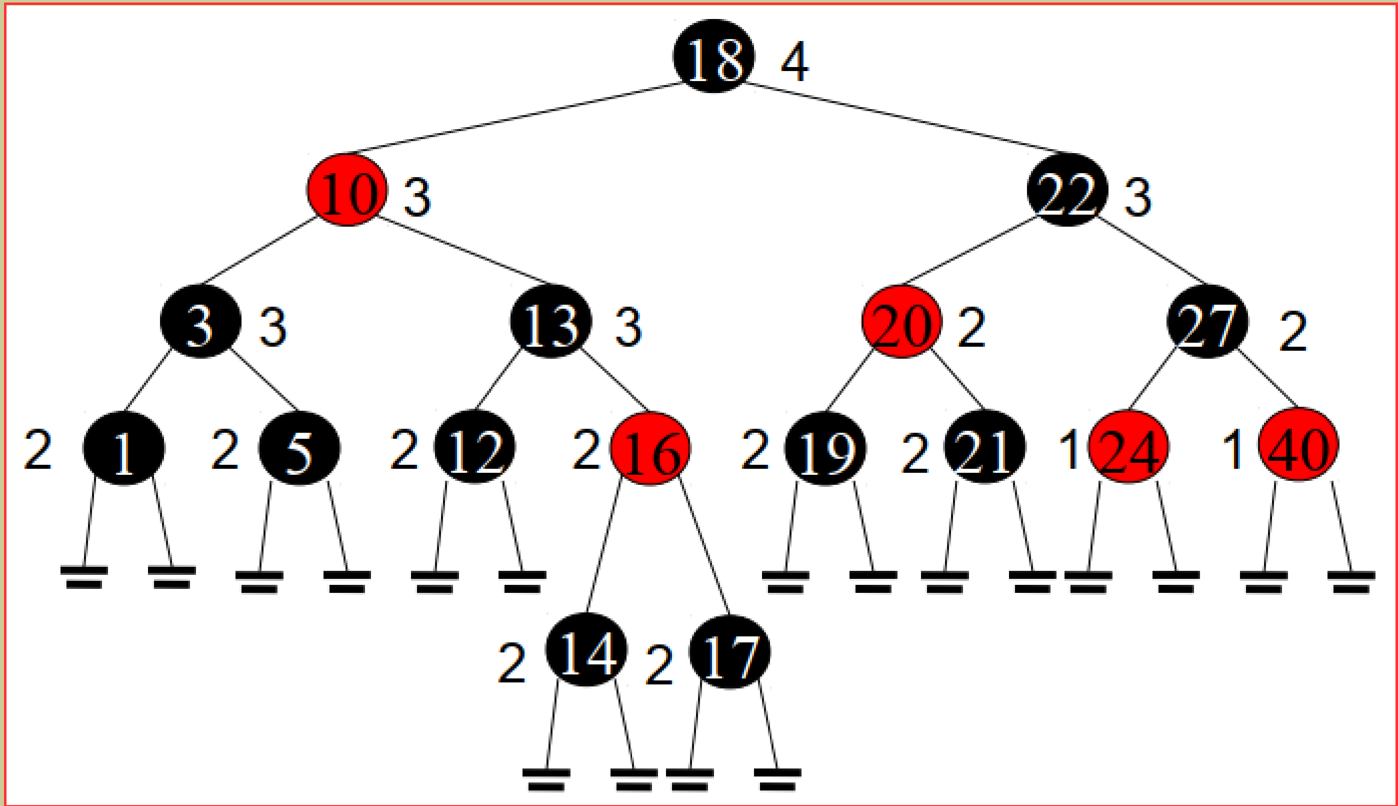
- P1: Qualquer nodo é com **Vermelho** ou com **Preto**
- P2: A raiz é um nodo com **Preto**
- P3: Qualquer nodo que representa um subárvore vazia (NULL) é com **Preto**
- P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**
- P5: Para cada nodo, todos os caminhos deste nodo para as folhas descendentes contém o mesmo número de nodos com **Preto**

Operações sobre uma Árvore Vermelho-Preto

- Após a realização de uma operação sobre uma árvore
  - é testado se as propriedades são satisfeitas
  - se alguma não é satisfeita, então
    - são realizadas rotações e/ou ajustes de cores, para que a árvore se mantenha balanceada

Altura preta

- É o número de nodos com **Preto** que existem no caminho entre um dado nodo e qualquer folha sua descendente



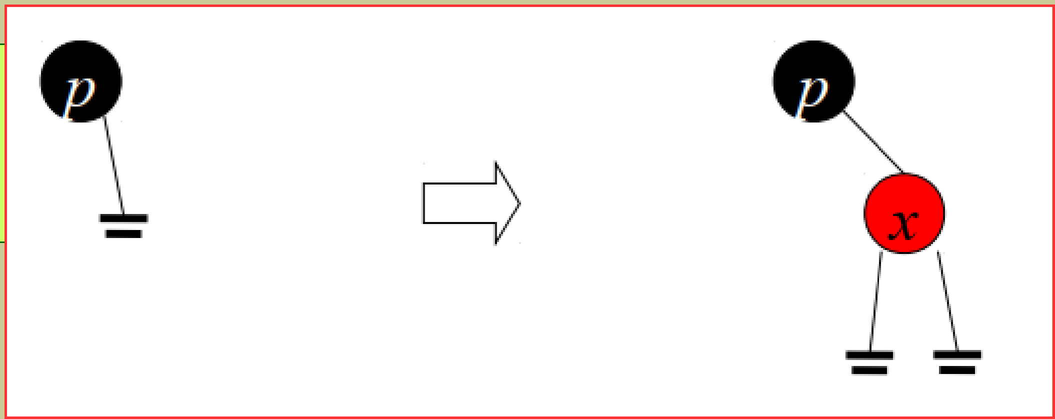
Operações básicas sobre uma árvore

As operações de Inserir e Remover

- Podem fazer com que algumas das propriedades deixem de ser satisfeitas
  - ou seja, podem desequilibrar a árvore
- São muito semelhantes às operações para Árvores Binárias de Pesquisa
  - apenas é necessário alterar (acertar) as cores dos nodos, para que as propriedades de árvore Vermelho-Preto sejam satisfeitas

Inserir um nodo numa árvore

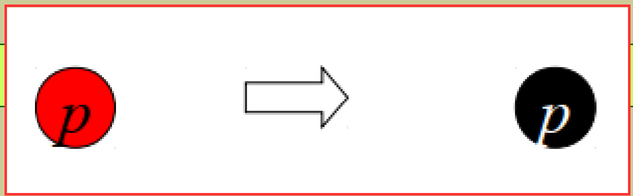
- Modo de inserir
  - um nodo é inserido sempre com **Vermelho**, não alterando a Altura Preta da árvore
    - se o nodo fosse inserido com **Preto**, a propriedade 5 não era satisfeita, pois um dos caminhos tinha mais um nodo com **Preto** que os restantes



P5: Para cada nodo, todos os caminhos deste nodo para as folhas descendentes contém o mesmo número de nodos com **Preto**

Inserir um nodo numa árvore

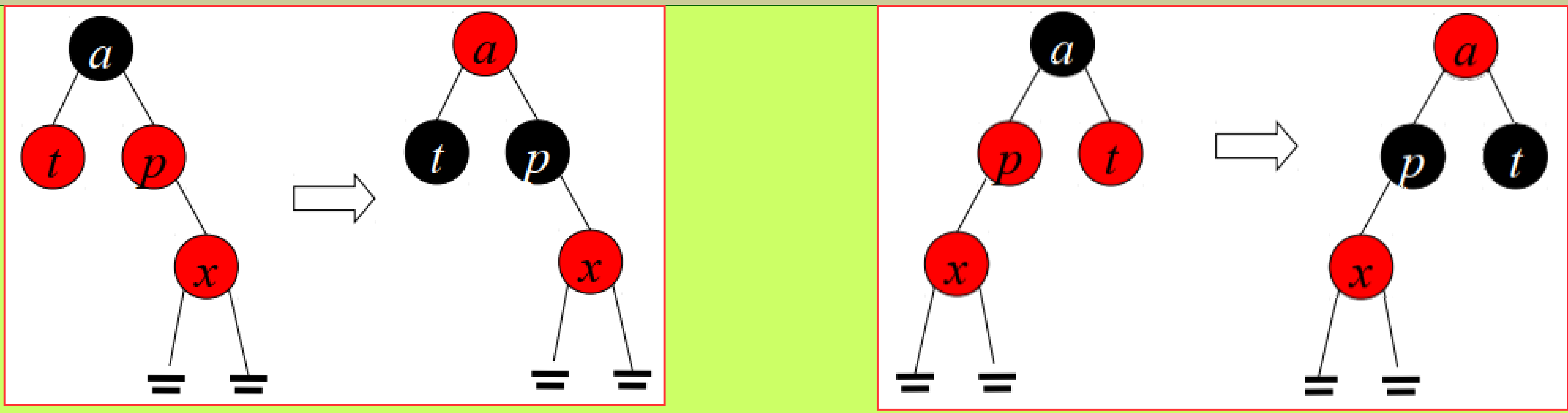
- Pesquisar a posição onde o novo nodo deve ser inserido, partindo da raiz em direção aos nodos com o valor mais próximo do nodo a ser inserido
  - Caso 1: inserir o nodo **p** numa árvore vazia
    - inserir com **Vermelho** e alterar a cor do nodo (que passa a ser a raiz) para **Preto**, satisfazendo a propriedade 2



P2: O nodo da raiz é com **Preto**

Inserir um nodo numa árvore

- Pesquisar a posição onde o novo nodo deve ser inserido, partindo da raiz em direção aos nodos com o valor mais próximo do nodo a ser inserido
  - Caso 2: inserir um nodo **x**, em que o seu tio é **Vermelho** (nodo **t**)
    - necessário fazer a recoloração (alterar cores) dos nodos **a**, **t** e **p** (avô, tio e pai de **x**)

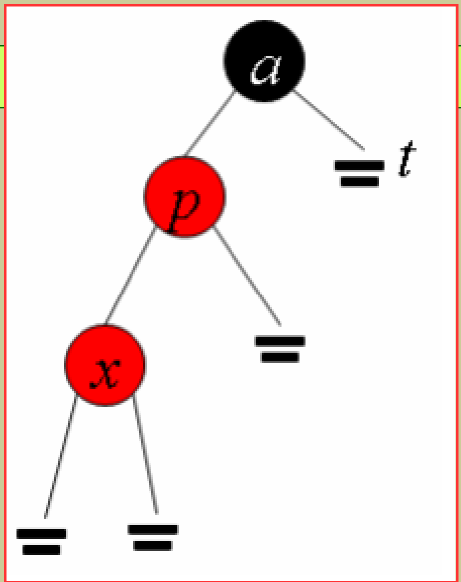


P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

  - Observações:
    - se o pai de **a** é **Vermelho**, o reequilíbrio tem que ser feito de novo;
    - se **a** é raiz, então **a** passa a ser **Preto**.

Inserir um nodo numa árvore

- Pesquisar a posição onde o novo nodo deve ser inserido, partindo da raiz em direção aos nodos com o valor mais próximo do nodo a ser inserido
  - Caso 3: inserir um nodo com **x**, em que o seu tio (raiz da subárvore **t**) é com **Preto**
    - para satisfazer a propriedade 4, é necessário fazer rotações envolvendo **a**, **t**, **p** e **x**



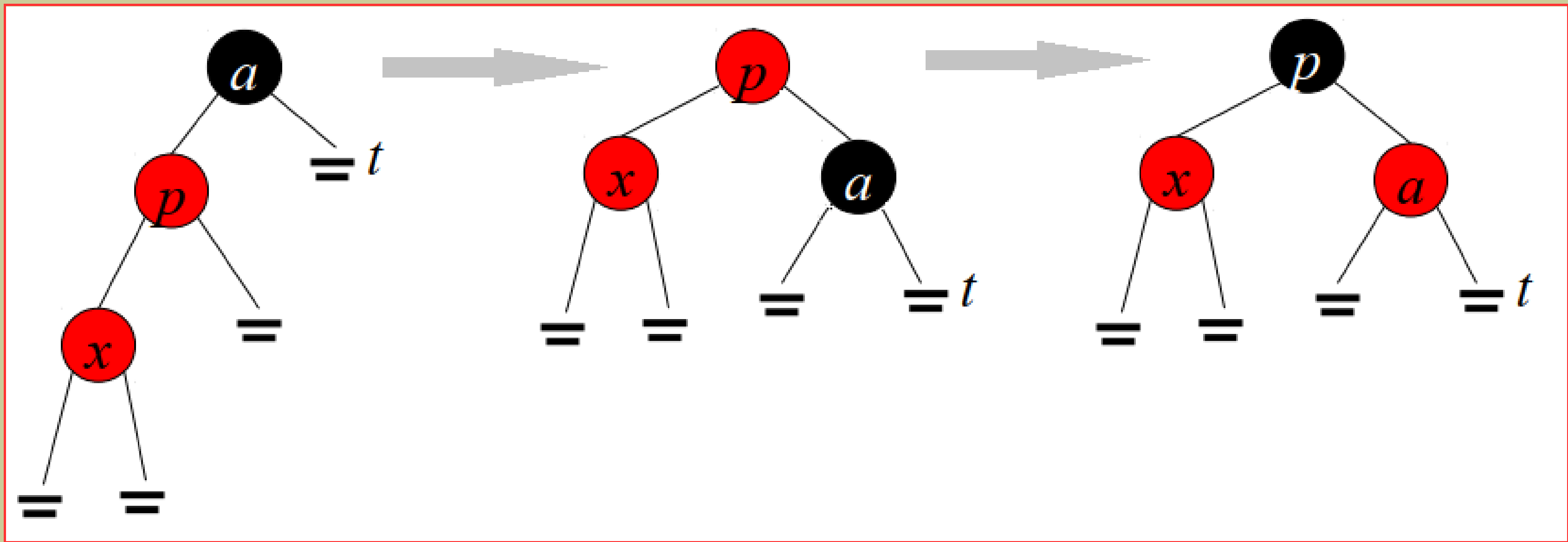
P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

- Há 4 possíveis rotações:
  - rotação à Direita
  - rotação à Esquerda
  - rotação Dupla Esquerda (Direita seguida de Esquerda)
  - rotação Dupla Direita (Esquerda seguida de Direita)

Inserir um nodo numa árvore

- Pesquisar a posição onde o novo nodo deve ser inserido, partindo da raiz em direção aos nodos com o valor mais próximo do nodo a ser inserido
  - Caso 3: inserir um nodo **x**, em que o seu tio (raiz da subárvore **t**) é com **Preto**

    a) Rotação à Direita do nodo **a** (avô de **x**) usando o nodo **p** (pai de **x**)

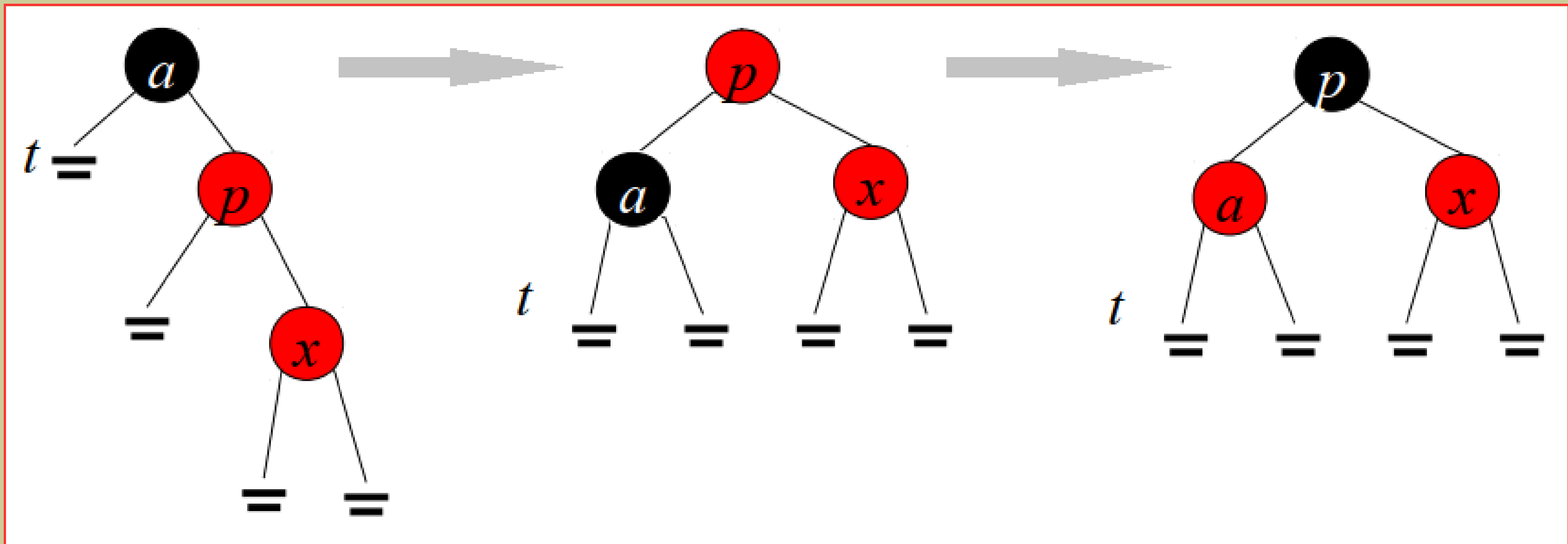


- primeiro: inserir o nodo com **x** como folha
- segundo: rodar à direita o nodo com **a** usando o seu filho esquerdo (com **p**)
- terceiro: recoloração dos nodos com **p** e **a** (pai e avô de **x**)

Inserir um nodo numa árvore

- Pesquisar a posição onde o novo nodo deve ser inserido, partindo da raiz em direção aos nodos com o valor mais próximo do nodo a ser inserido
  - Caso 3: inserir um nodo **x**, em que o seu tio (raiz da subárvore **t**) é com **Preto**

    b) Rotação à Esquerda do nodo **a** (avô de **x**) usando o nodo **p** (pai de **x**)

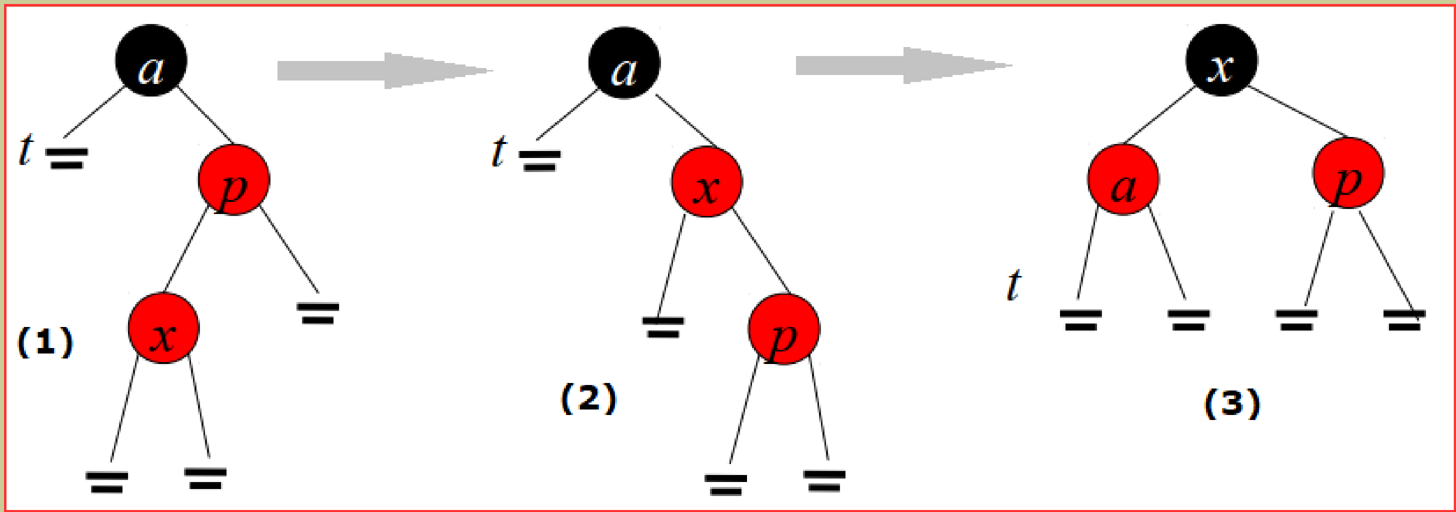


- primeiro: inserir o nodo com **x** como folha
- segundo: rodar à esquerda o nodo com **a** usando o seu filho direito (com **p**)
- terceiro: recoloração dos nodos com **p** e **a** (pai e avô de **x**)

Inserir um nodo numa árvore

- Pesquisar a posição onde o novo nodo deve ser inserido, partindo da raiz em direção aos nodos com o valor mais próximo do nodo a ser inserido
  - Caso 3: inserir um nodo **x**, em que o seu tio (raiz da subárvore **t**) é com **Preto**

    c) Rotação Dupla Esquerda (Direita seguida de Esquerda)

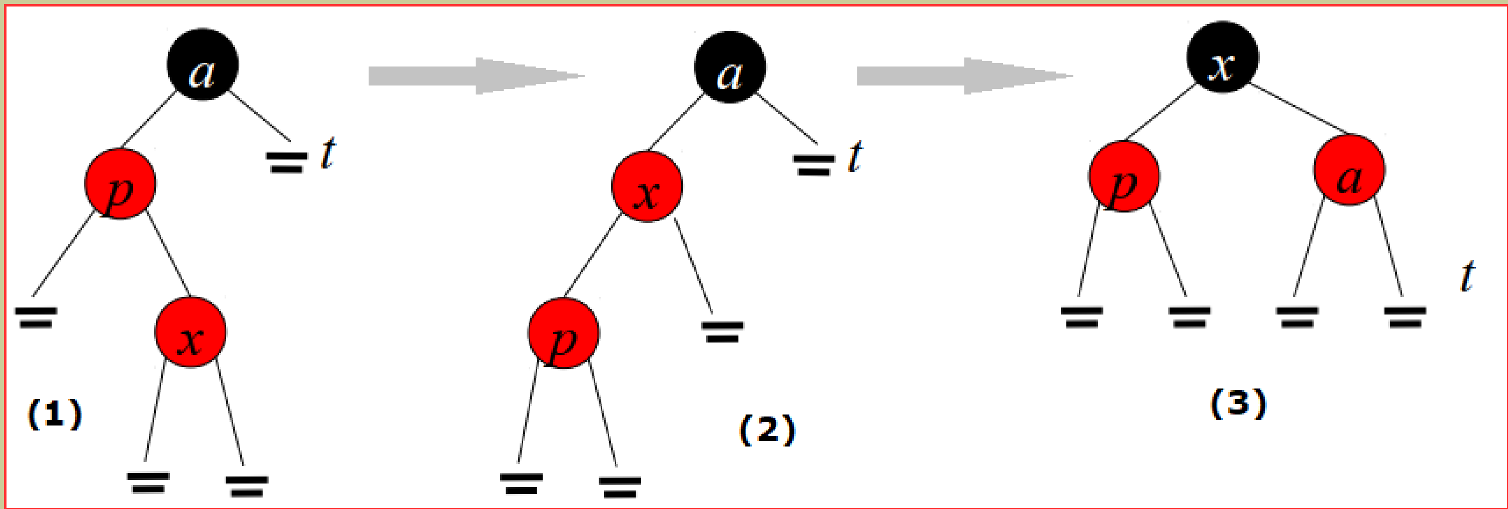


- a árvore (1) é obtida com a inserção do nodo com **x**
- a árvore (2) é obtida com uma rotação à **Direita** do nodo com **p** na árvore (1)
- a árvore (3) é obtida com uma rotação à **Esquerda** do nodo com **a** da árvore (2), seguida de recoloração (alteração de cores) dos nodos **x** e **a**

Inserir um nodo numa árvore

- Pesquisar a posição onde o novo nodo deve ser inserido, partindo da raiz em direção aos nodos com o valor mais próximo do nodo a ser inserido
  - Caso 3: inserir um nodo **x**, em que o seu tio (raiz da subárvore **t**) é com **Preto**

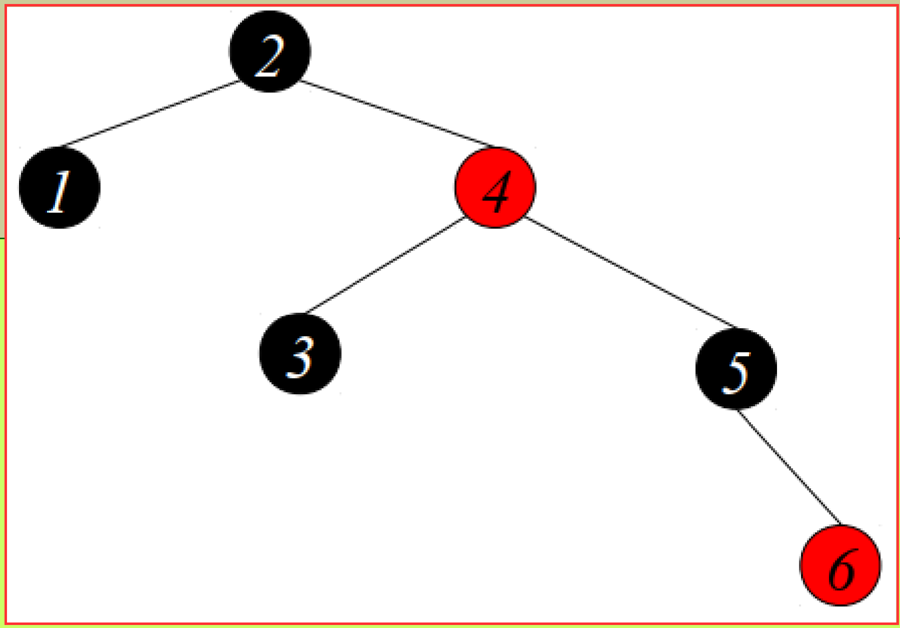
    d) Rotação Dupla Direita (Esquerda seguida de Direita)



- a árvore (1) é obtida com a inserção do nodo com **x**
- a árvore (2) é obtida com uma rotação à **Esquerda** do nodo com **p** da árvore (1)
- a árvore (3) é obtida com uma rotação à **Direita** do nodo com **a** da árvore (2), seguida de recoloração (alteração de cores) dos nodos **x** e **a**

Inserir um nodo numa árvore – exemplo 1

- Estado inicial da árvore



P1: Qualquer nodo é com **Vermelho** ou com **Preto**

P2: A raiz é um nodo com **Preto**

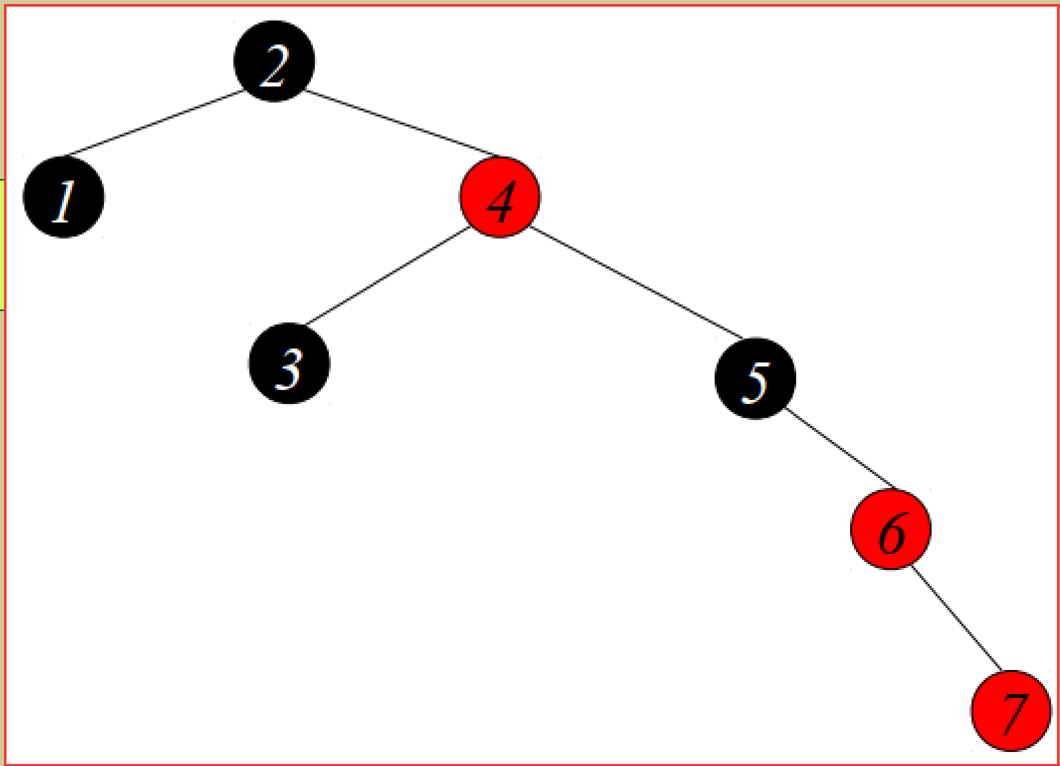
P3: Qualquer nodo que representa um subárvore vazia (NULL) é com **Preto**

P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

P5: Para cada nodo, todos os caminhos deste nodo para as folhas descendentes contém o mesmo número de nodos com **Preto**

Inserir um nodo numa árvore – exemplo 1

- Inserção do nodo com o valor 7

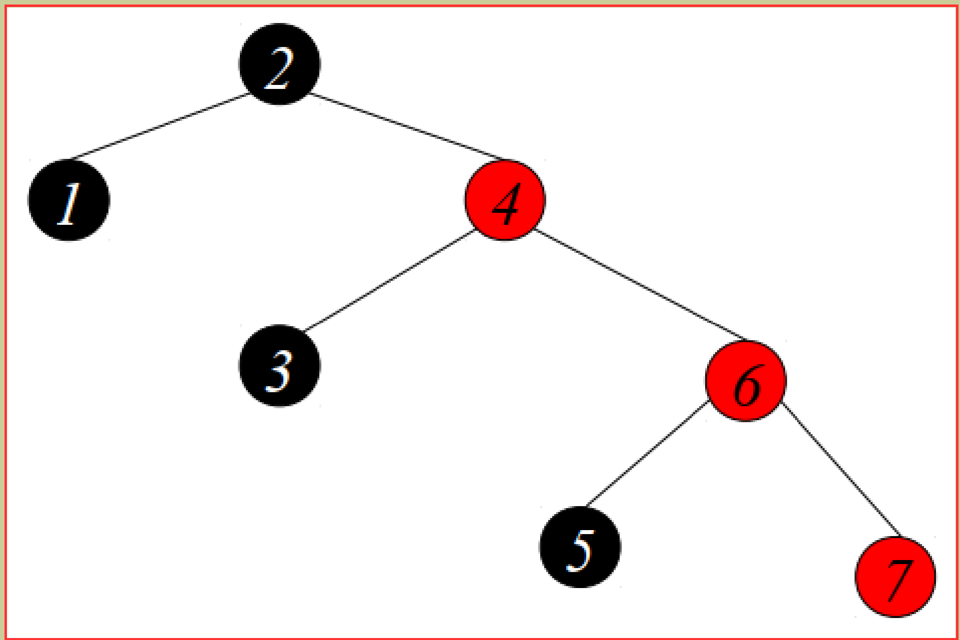


P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

- A propriedade 4 não é satisfeita:
  - o nodo pai do nodo com 7 (**Vermelho**) também é com **Vermelho**
- Para reequilibrar, aplicar o **caso 3.b)**
  - **rotação à esquerda** do nodo com **5** (avô de 7) usando o nodo com **6** (pai de **7**)

Inserir um nodo numa árvore – exemplo 1

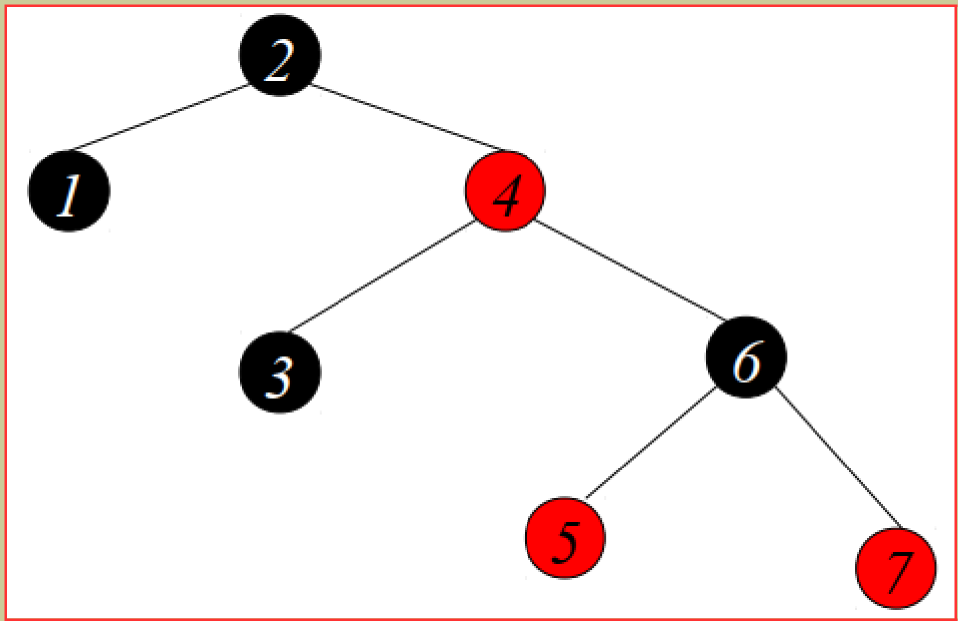
- Rotação à Esquerda do nodo 5 (avô de 7)



- Segue-se a recoloração dos nodos 5 e 6.

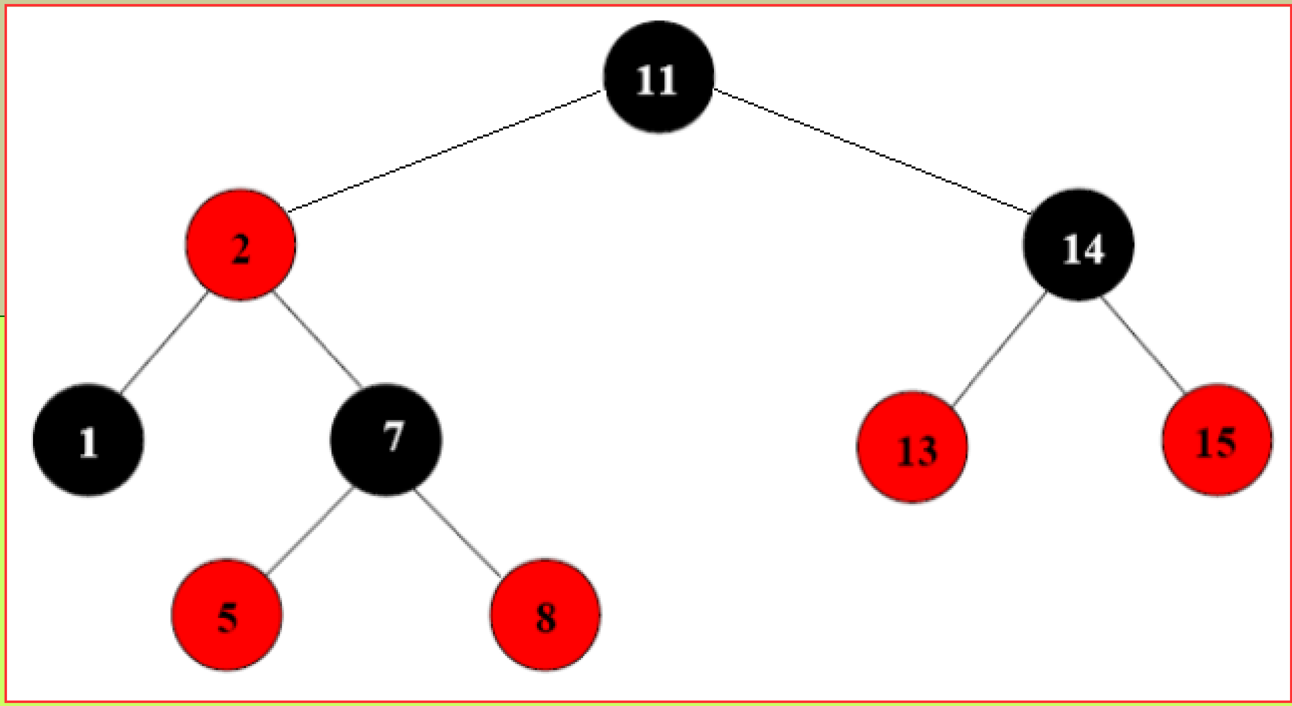
Inserir um nodo numa árvore – exemplo 1

- Recoloração dos nodos 5 e 6



Inserir um nodo numa árvore – exemplo 2

- Estado inicial da árvore



P1: Qualquer nodo é com **Vermelho** ou com **Preto**

P2: A raiz é um nodo com **Preto**

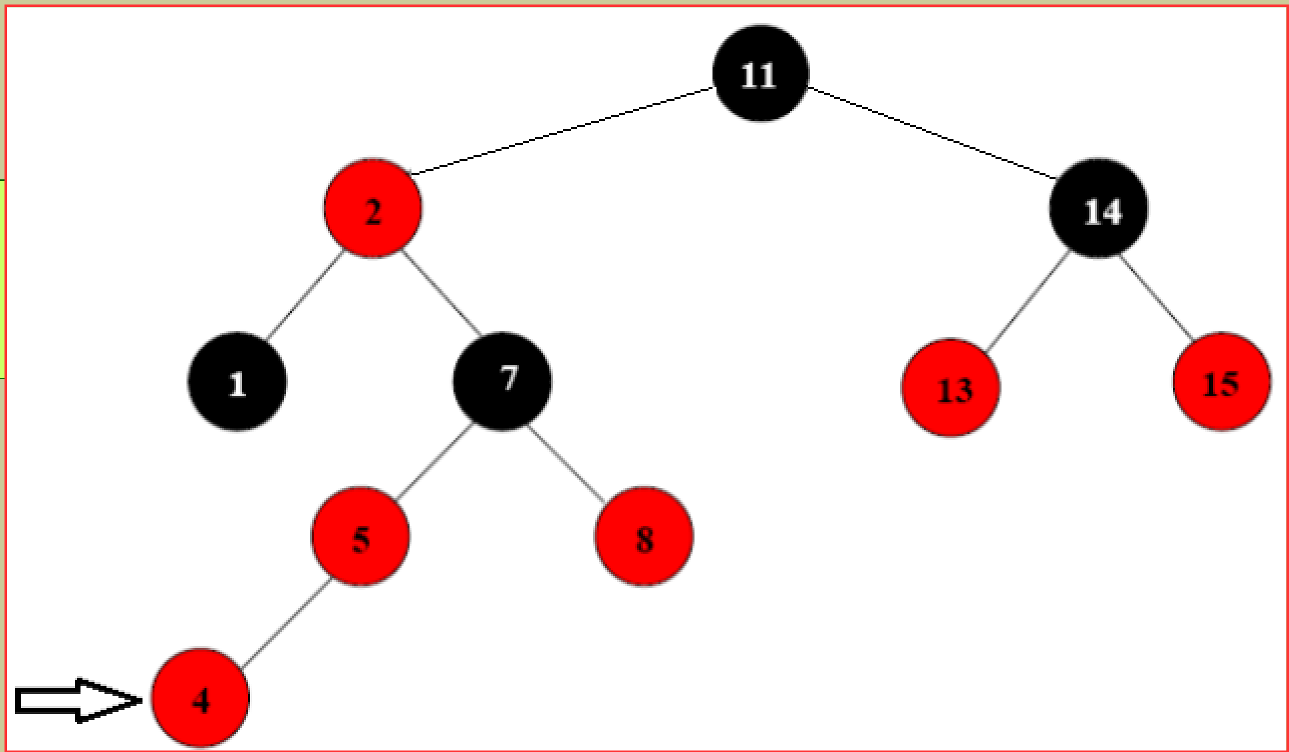
P3: Qualquer nodo que representa um subárvore vazia (NULL) é com **Preto**

P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

P5: Para cada nodo, todos os caminhos deste nodo para as folhas descendentes contém o mesmo número de nodos com **Preto**

Inserir um nodo numa árvore – exemplo 2

- Inserção do nodo com o valor 4

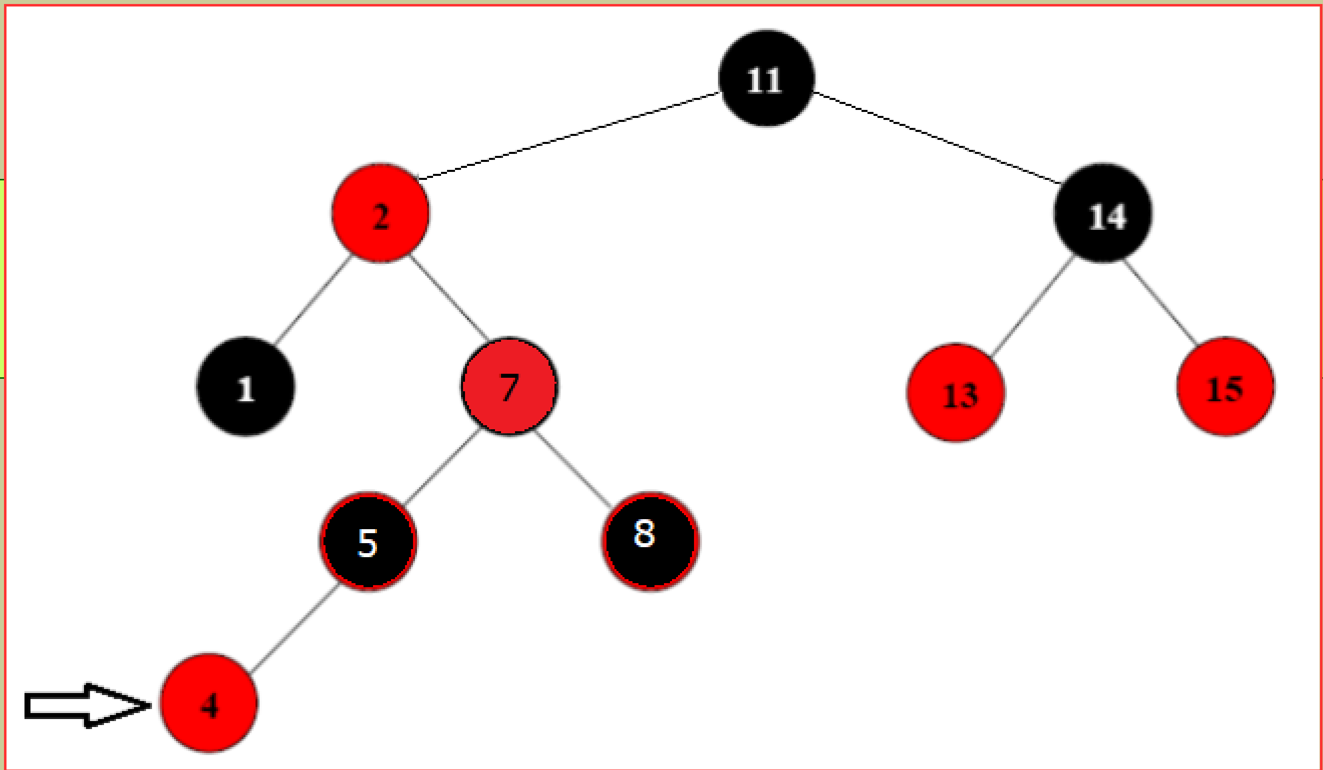


P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

- a propriedade 4 não é satisfeita: o nodo 4 e o seu pai (nodo 5) são ambos **Vermelho**
- para reequilibrar, aplicar o **caso 2** (tio de **4** é **Vermelho**)

Inserir um nodo numa árvore – exemplo 2

- Caso 2: o tio do nodo 4 (nodo 8) é com **Vermelho**, logo recolorar os nodos 7, 5 e 8
    - os nodos 5 e 8 passam a **Preto** e o nodo 7 passa a **Vermelho**



P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

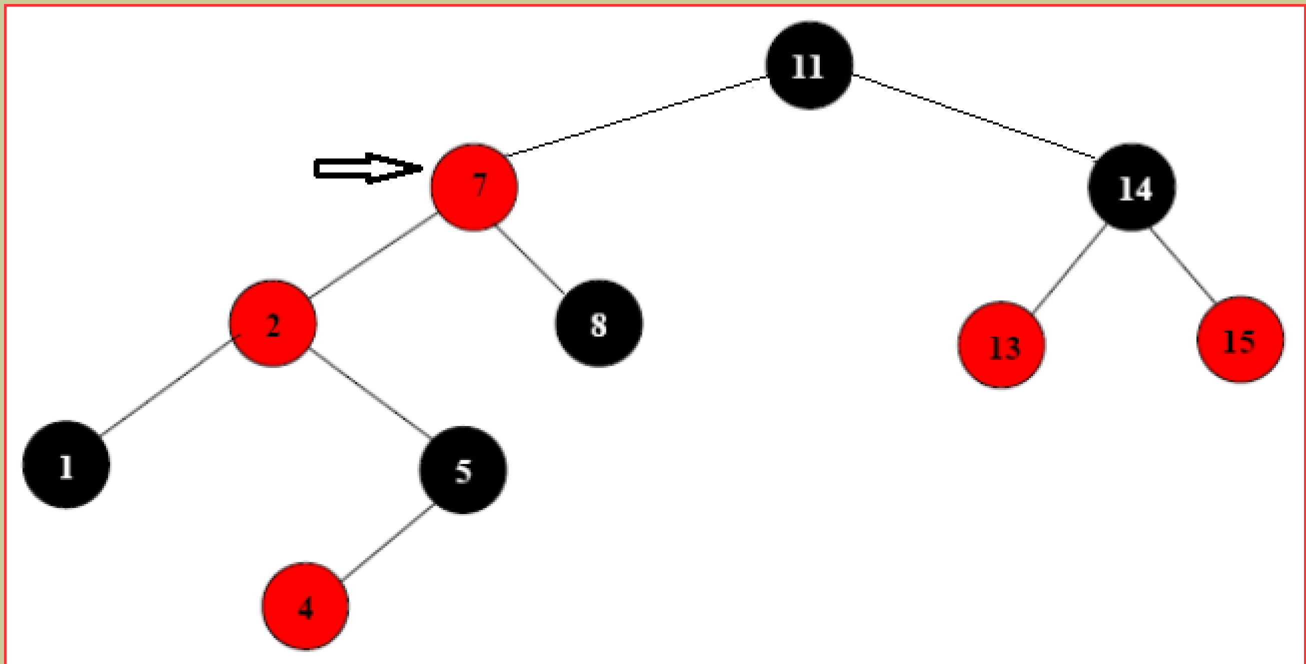
- a propriedade 4 não é satisfeita: o nodo 7 e o seu pai (nodo 2) são com **Vermelho**
- para reequilibrar, aplicar o **caso 3.d)** (o tio de 7 é com **Preto** - *rotação Dupla Direita*)

Inserir um nodo numa árvore – exemplo 2

- Caso 3: o tio do nodo 7 (nodo 14) é **Preto**

  d.1) rotação à esquerda do nodo 2 (pai de 7)

  - nodo 7 é filho à direita do nodo 2



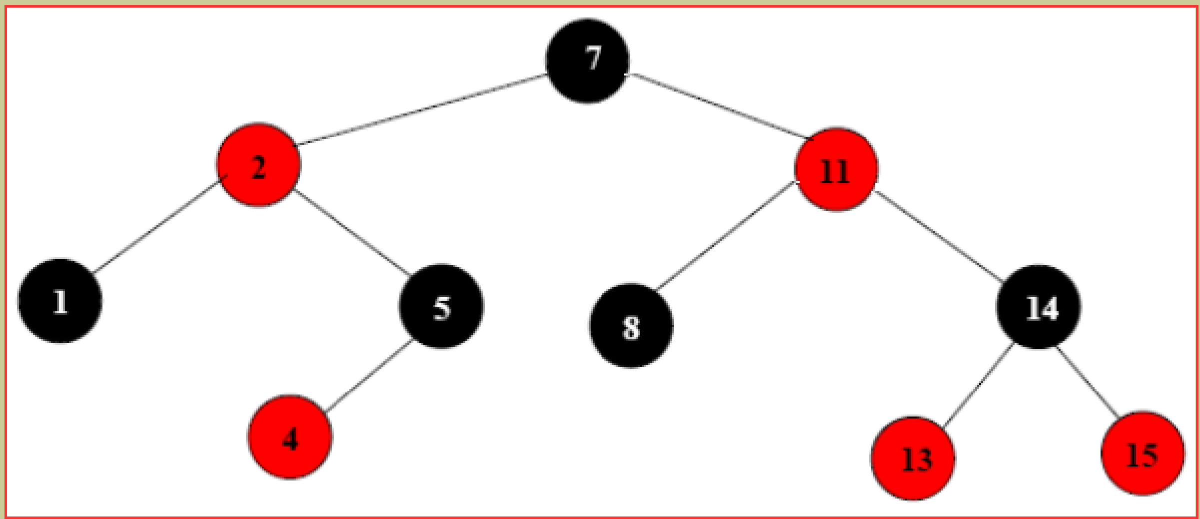
  - o filho à esquerda do nodo 7 passa a ser filho à direita do nodo 2
  - de seguida, aplica-se uma rotação à Direita

Inserir um nodo numa árvore – exemplo 2

- Caso 3: o tio do nodo 7 (nodo 14) é com **Preto**

  d.2) rotação à Direita do nodo 11

  - o tio do nodo 2 (nodo 8) é com **Preto**, e
  - o nodo 2 é filho à esquerda do nodo 7

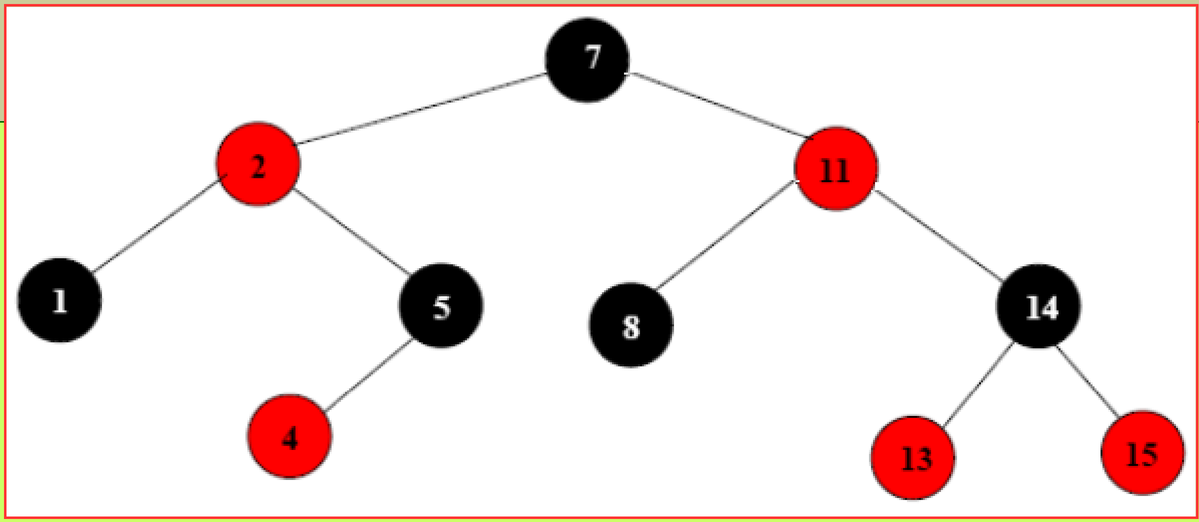


  - o filho à direita do nodo 7 passa a ser filho à esquerda do nodo 11
  - o nodo 7 (raiz) é colorido com **Preto.**

- A propriedade 4 volta a ser satisfeita

Inserir um nodo numa árvore – exemplo 2

- Todas as propriedades são satisfeitas



P1: Qualquer nodo é com **Vermelho** ou com **Preto**

P2: A raiz é um nodo com **Preto**

P3: Qualquer nodo que seja raiz de uma subárvore vazia (NULL) é com **Preto**

P4: Se um nodo é com **Vermelho**, então os seus filhos são nodos com **Preto**

P5: Para cada nodo, todos os caminhos deste nodo para as folhas descendentes contém o mesmo número de nodos com **Preto**

Inserir um nodo numa árvore

- Algoritmo

1. Criar um novo nodo com o elemento **x**, marcando-o com a cor **Vermelho**

2. Inserir aquele nodo na árvore, usando a operação Inserir numa ABP

3. Se o nodo com **x** é raiz, **então**

- **a)** trocar de cor o nodo com **x** para **Preto**
- **b)** incrementar em 1 unidade a altura Preta da árvore

4. Se a cor do pai do nodo com **x** não é **Preto** e o nodo com **x** não é raiz, **então**

- **a)** Se o tio do nodo com **x** é **Vermelho** (cor do avô também é **Preto** – P5)
  - **i)** Alterar a cor do pai e do tio para **Preto**
  - **ii)** A cor do avô é **Vermelho**
  - **iii)** O nodo **x** passa a ser o seu avô e repetir os passos 3 e 4 para o novo **x**
- **b) Se** a cor do tio do nodo **x** é **Preto**, **então** aplicar um dos seguintes 4 casos:
  - **i)** rotação à Direita do nodo avô (caso 3.a) e trocar as cores dos nodos avô e pai
  - **ii)** rotação à Esquerda do avô (caso 3.b) e trocar as cores dos nodos avô e pai
  - **iii)** rotação Dupla Direita (caso 3.c) e trocar as cores dos nodos **x** e avô
  - **iv)** rotação Dupla Esquerda (caso 3.d) e trocar as cores dos nodos **x** e avô

Remover um nodo duma árvore

- Em qualquer das operações (Inserção ou Remoção), para que as propriedades se mantenham satisfeitas, é necessário
  - a recoloração dos nodos e,
  - realizar rotações de nodos.
- Para se selecionar o caso apropriado para reequilibrar a árvore
  - na Inserção, verifica-se a cor do nodo tio do nodo que provoca desequilíbrio
  - na Remoção, verifica-se a cor do irmão do nodo que provoca desequilíbrio.
- A principal propriedade que é violada
  - após a Inserção: a existência de 2 nodos consecutivos num caminho de cor **Vermelho**
  - após a Remoção: a alteração da **altura preta** nas subárvores, pois a remoção de um nodo pode causar a redução da **altura preta**, num dos caminhos entre a raiz e uma das folhas suas descendentes

Remover um nodo duma árvore – Método

- O processo de Remover um nodo usa a noção de nodo **Preto Duplo**
  - quando um nodo **Preto** é removido e substituído por um **filho Preto**, este filho é marcado como **Preto Duplo**
  - a tarefa principal é agora de converter este nodo **Preto Duplo** em apenas **Preto**.

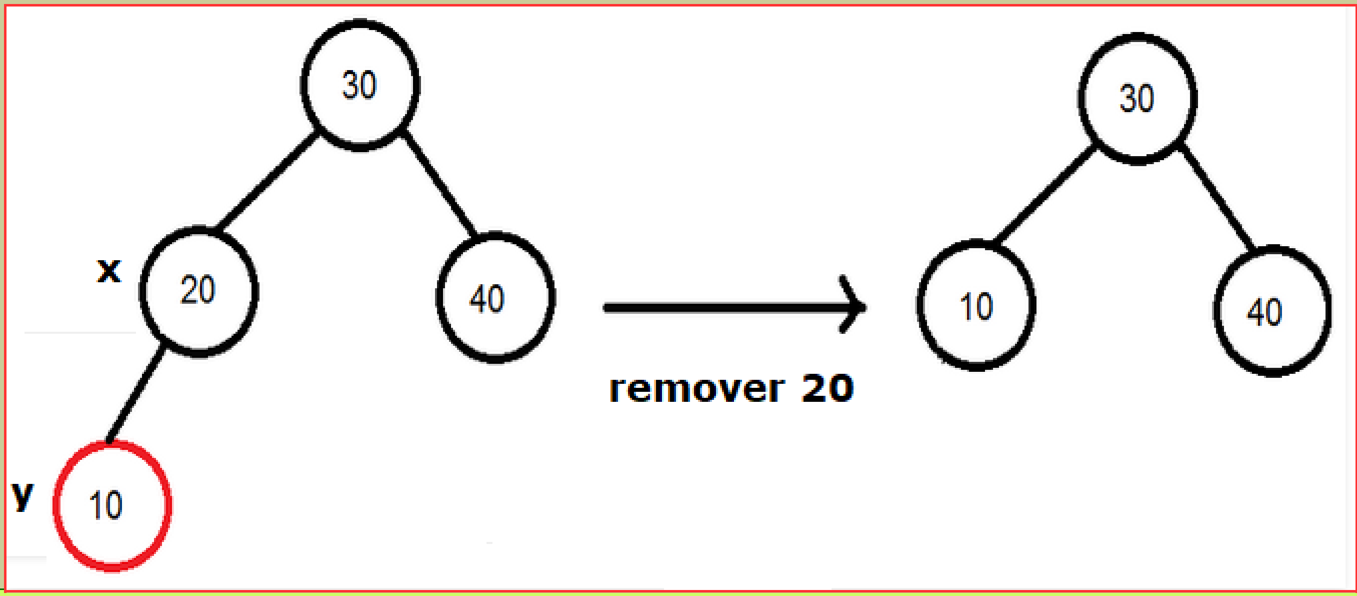
Remover um nodo duma árvore – Método

Passo 1: realizar a operação Remover associada às ABP
- Na prática, o nodo removido é sempre uma folha ou um nodo com apenas um filho
- O processo de remoção de um nodo com 2 filhos (subárvor não vazia) tem 2 etapas:
  - substituir o valor deste nodo pelo
    - seu sucessor (nodo mais esquerda do seu filho à direita), ou
    - antecessor (nodo mais à direita do seu filho à esquerda)
  - remover o nodo substituto (que é uma folha ou tem só um filho)
- Desta forma, basta analisar estes 2 casos
  - seja **x** o nodo a remover e **y** o nodo filho de **x** que o vai substituir
  - note-se que **y** = NULL quando **x** é uma folha e a cor de NULL é **Preto**.

Remover um nodo duma árvore – Método

Passo 2 (caso 1): x ou **y** é **Vermelho** (não podem ser os dois **Vermelho**)

- Depois de remover o nodo **x** (passo 1), marcar o nodo **y** com **Preto**
  - não há alteração na **altura Preta**, pois segundo a propriedade 4, **x** e **y** não podem ser ambos **Vermelho** (**x** é pai de **y**)



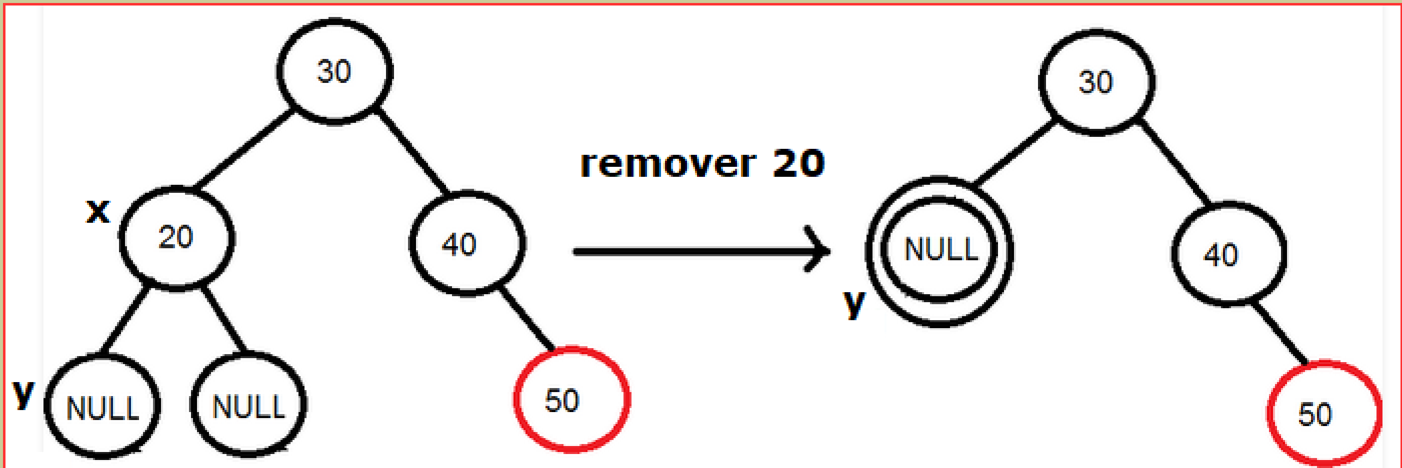
P4: Se um nodo é com **Vermelho**, então os seus nodos filhos são com **Preto**

Remover um nodo duma árvore – Método

Passo 2 (caso 2): **x** e **y** são ambos **Preto**

1. Depois de removido o nodo **x** (passo 1) marcar o nodo **y** com **Duplo Preto**

- notar que se **x** é uma folha, então **y** é NULL que é **Preto**;
  - assim, a remoção de uma folha **Preto** provoca um **Duplo Preto**



- quando o nodo 20 é removido, é substituído por NULL e o NULL torna-se **Duplo Preto**
- mas a operação ainda não terminou:
  - o nodo **Duplo Preto** tem que tornar-se apenas com **Preto**

Remover um nodo duma árvore – Método

Passo 2 (caso 2): **x** e **y** são ambos **Preto**

2. Seja **s** o irmão do nodo **y** (antes era de **x**) e **p** o pai daqueles 2 nodos

enquanto o nodo atual **y** é **Duplo Preto** e **y** não é raiz, **fazer**:

a) Se o nodo **s** é **Preto** e um (ou dois) dos seus filhos é **Vermelho** fazer rotações

Seja **r** o filho **Vermelho** de **s** (4 subcasos, dependendo da posição de **s** e **r**):

i) rotação à Direita

- **s** é filho esquerdo de **p** e **r** é filho esquerdo de **s**

ii) rotação à Esquerda

- **s** é filho direito de **p** e **r** é filho direito de **s**

iii) rotação Dupla Direita

- **s** é filho esquerdo de **p** e **r** é filho direito de **s**

iv) rotação Dupla Esquerda

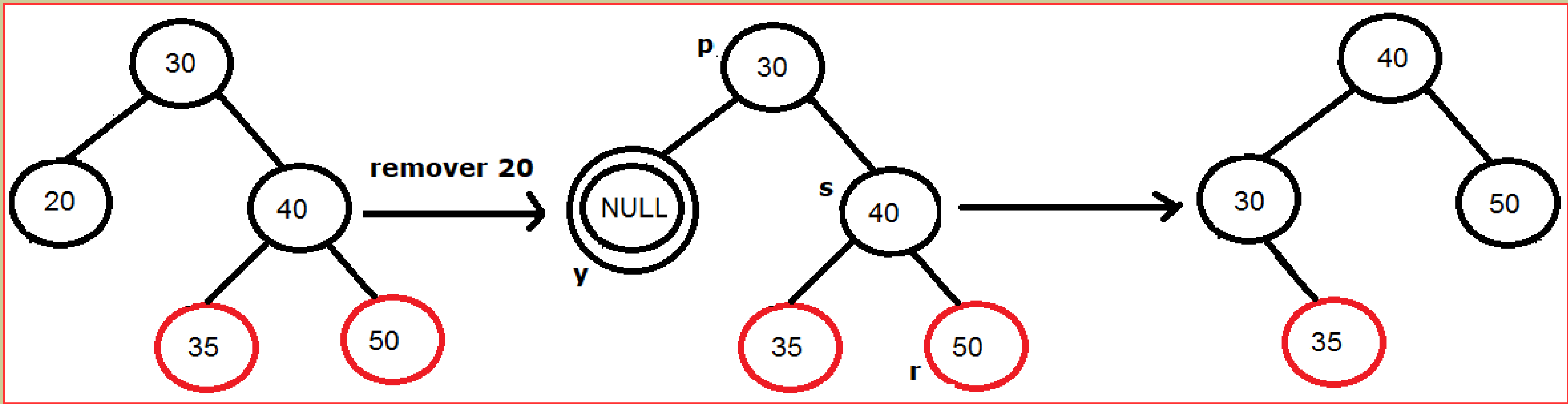
- **s** é filho direito de **p** e **r** é filho esquerdo de **s**

Remover um nodo duma árvore – Método

Passo 2 (caso 2): **x** e **y** são ambos **Preto**

2. Exemplo de Rotação à Esquerda – a.ii)

- o nodo **s** (irmão de y) é **Preto** com pelo menos um filho **Vermelho**
- o nodo **s** é filho direito de **p** e o filho direito de **s** é **Vermelho**

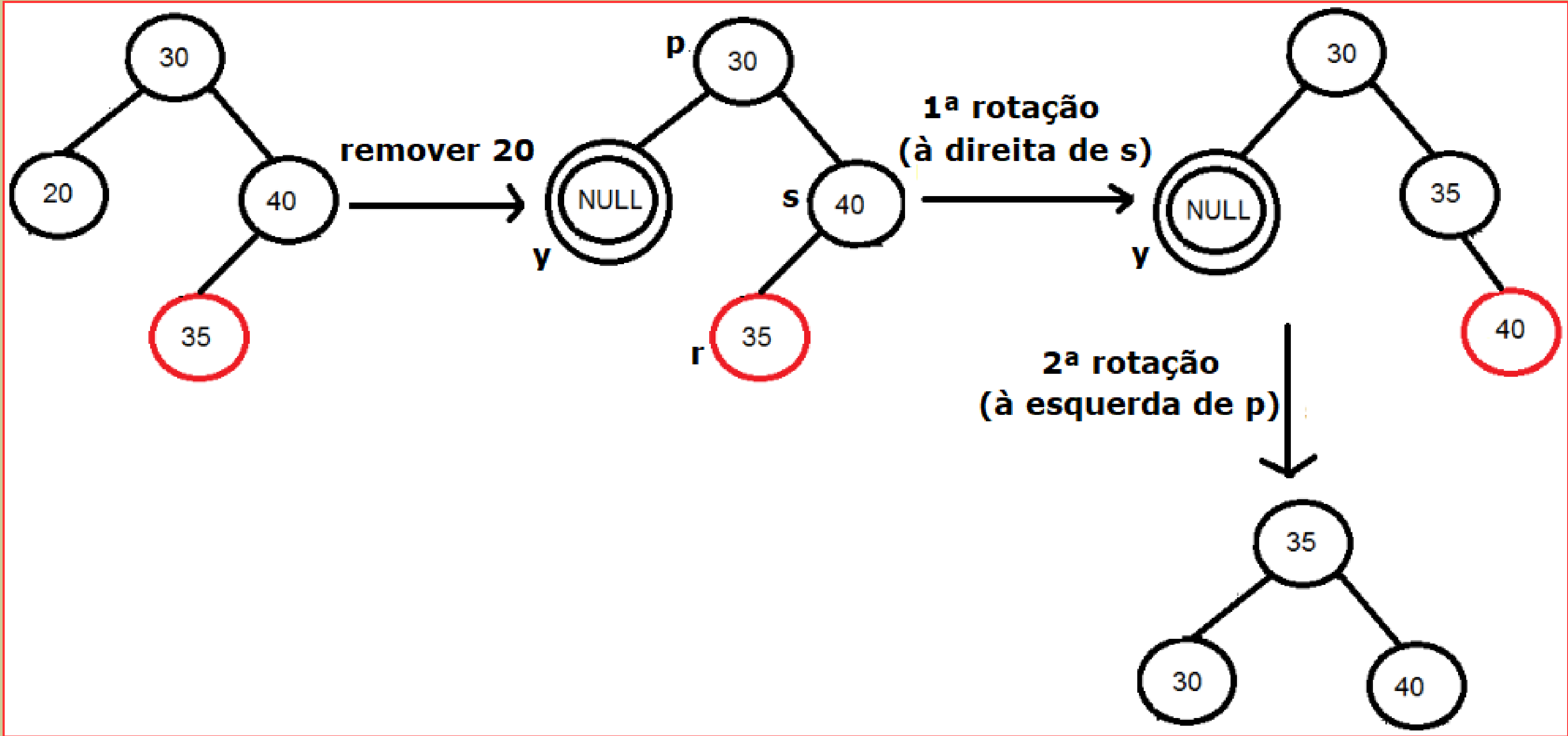


Remover um nodo duma árvore – Método

Passo 2 (caso 2): **x** e **y** são ambos **Preto**

2. Exemplo de Rotação Dupla Esquerda – a.iv)

- o nodo **s** (irmão de **y**) é **Preto** com pelo menos um filho **Vermelho**
- o nodo **s** é filho direito de **p** e o nodo **r** que é filho esquerdo de **s** é **Vermelho**



Remover um nodo duma árvore – Método

Passo 2 (caso 2): **x** e **y** são ambos **Preto**

2. Seja **s** o irmão do nodo **y** (antes de **x**) e **p** o pai daqueles 2 nodos

enquanto o nodo atual **y** é Duplo Preto e **y** não é raiz, **fazer**:

b) se o nodo **s** é **Preto** e ambos os seus filhos são **Preto**, **então**

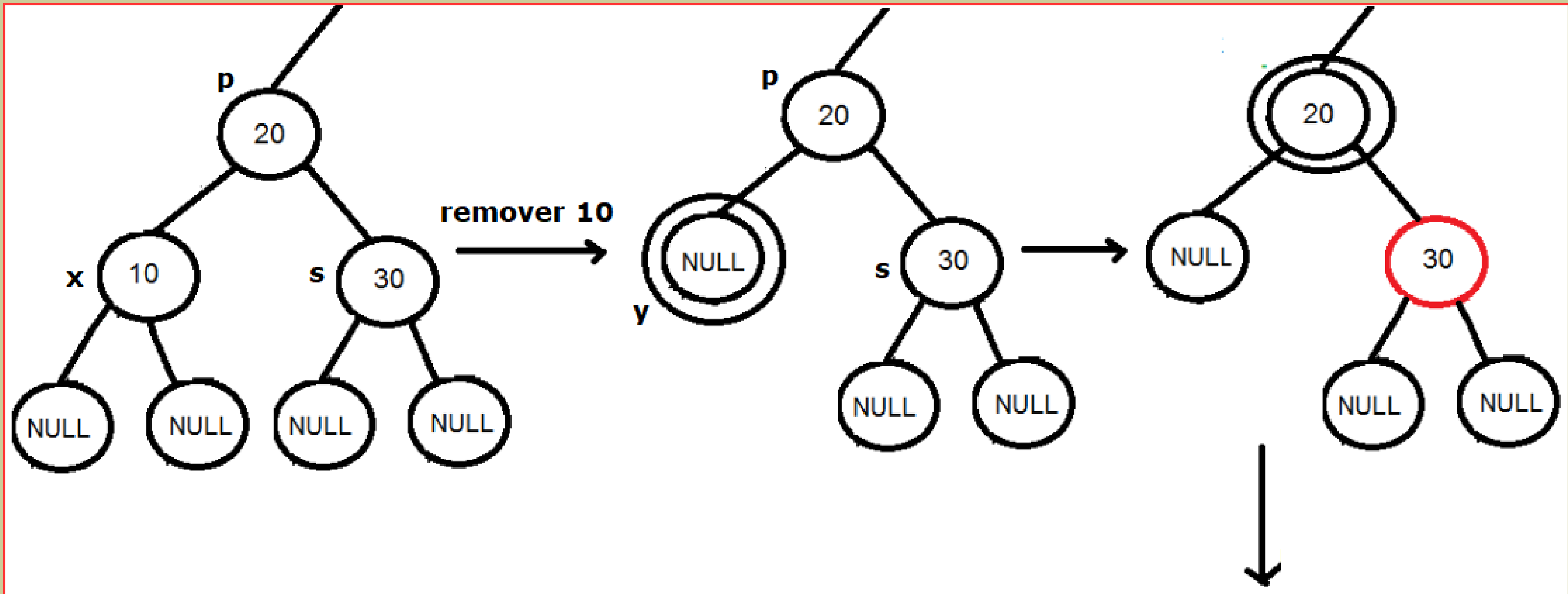
- realizar a recoloração
- recorrer ao nodo **p**, se este for **Preto**
  - passar a marca de **Duplo Preto** ao nodo **p**
- se o nodo **p** for **Vermelho**, então marcar o nodo **p** com **Preto**
  - **Vermelho** + **Duplo Preto** = **Preto**

Remover um nodo duma árvore – Método

Passo 2 (caso 2): **x** e **y** são ambos **Preto**

2. Seja **s** o irmão do nodo **y** (antes de **x**) e **p** o pai daqueles 2 nodos

- Exemplo de 2.b)



Remover um nodo duma árvore – Método

Passo 2 (caso 2): **x** e **y** são ambos **Preto**

2. Seja **s** o irmão do nodo **y** (antes de **x**) e **p** o pai daqueles 2 nodos

enquanto o nodo atual **y** é Duplo Preto e **y** não é raiz, **fazer**:

c) se o nodo **s** é **Vermelho**, **então**

- realizar uma rotação à Esquerda para deslocar o nodo **s** para cima
- recolorir os nodos **s** e **p** (alterar as cores)

- Observações:
  - o novo irmão (novo nodo **s**) é sempre Preto (ver diagrama em baixo)
  - principalmente, este caso resume-se a
    - transformar a árvore recolorando o nodo **s** com **Preto** (com uma rotação), e
    - aplicar a esta nova árvore os casos a) ou b)
  - portanto, este caso pode ser subdividido em 2 subcasos:

    i) Rotação à Direita (**s** é filho esquerdo de **p**) – rodar para a direita o nodo **p**

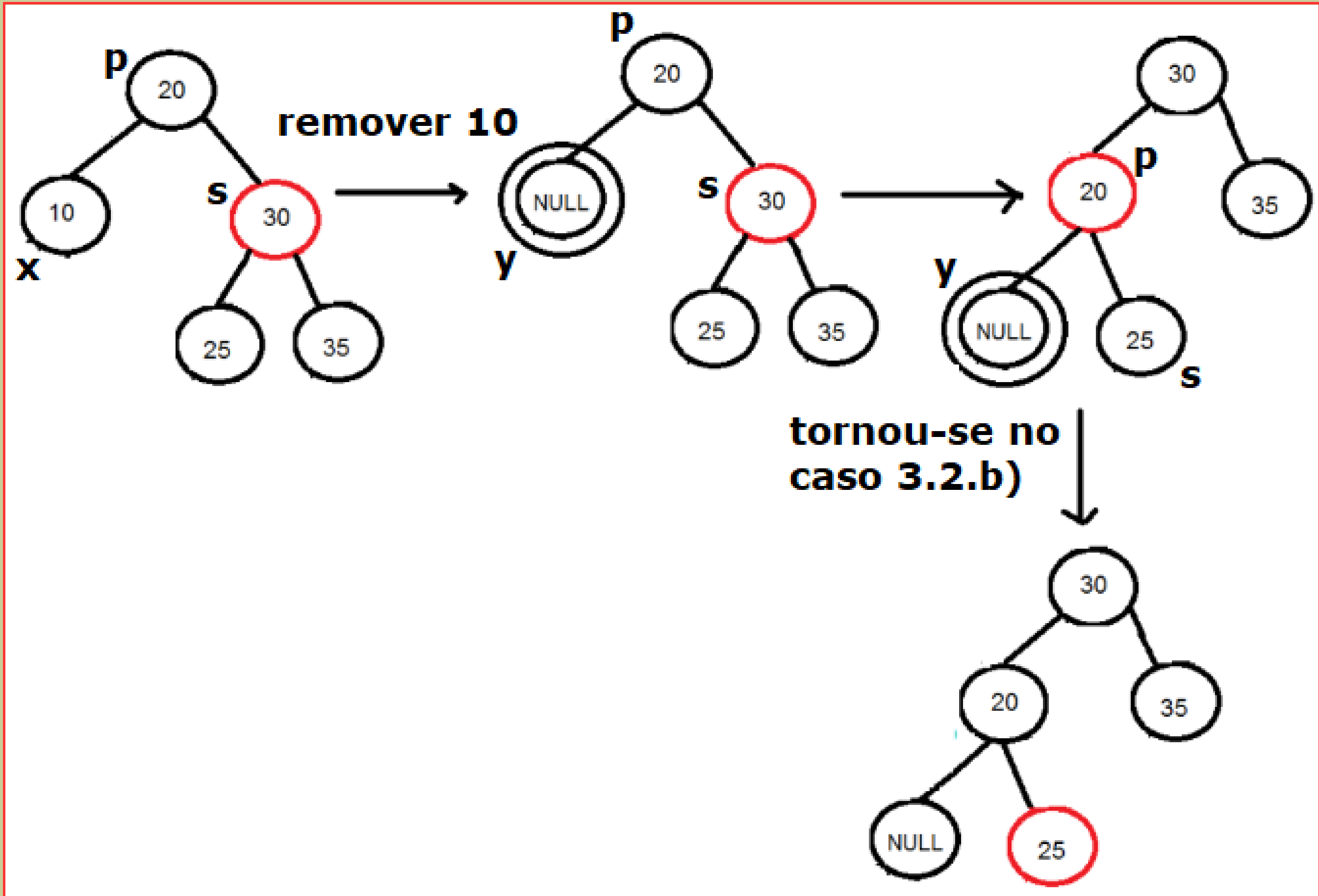
    ii) Rotação à Esquerda (**s** é filho direito de **p**) – rodar para a esquerda o nodo **p**

Remover um nodo duma árvore – Método

Passo 2 (caso 2): **x** e **y** são ambos **Preto**

2. Seja **s** o irmão do nodo **y** (antes de **x**) e **p** o pai daqueles 2 nodos

- Exemplo de 2.c)



Remover um nodo duma árvore – Método

Passo 2 (caso 2): **x** e **y** são ambos **Preto**

3. se y é raiz, **então**

- marcar o nodo y com **Preto**
- reduzir em 1 unidade a altura Preta da árvore, e
- terminar.