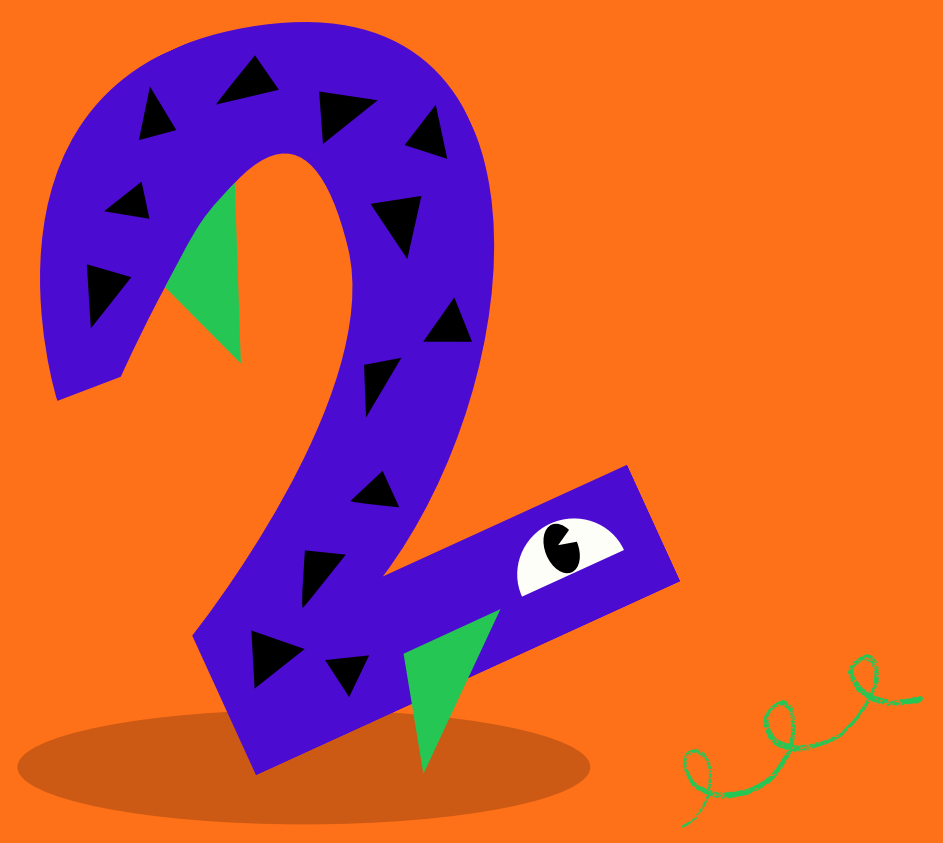


# Árvore Rubro-Negra

Senhor Monitor

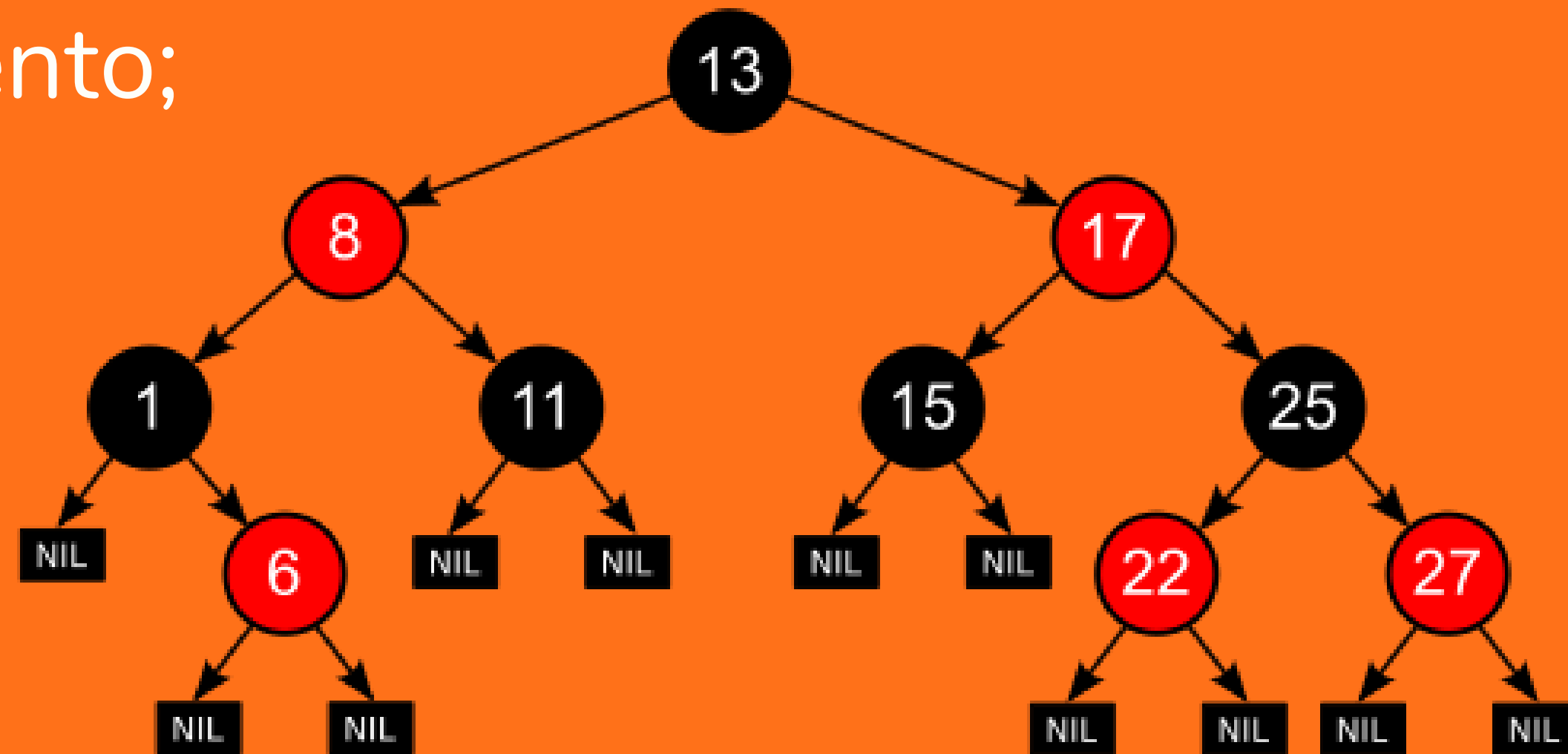
Fernando Antonio Marques

Schettini



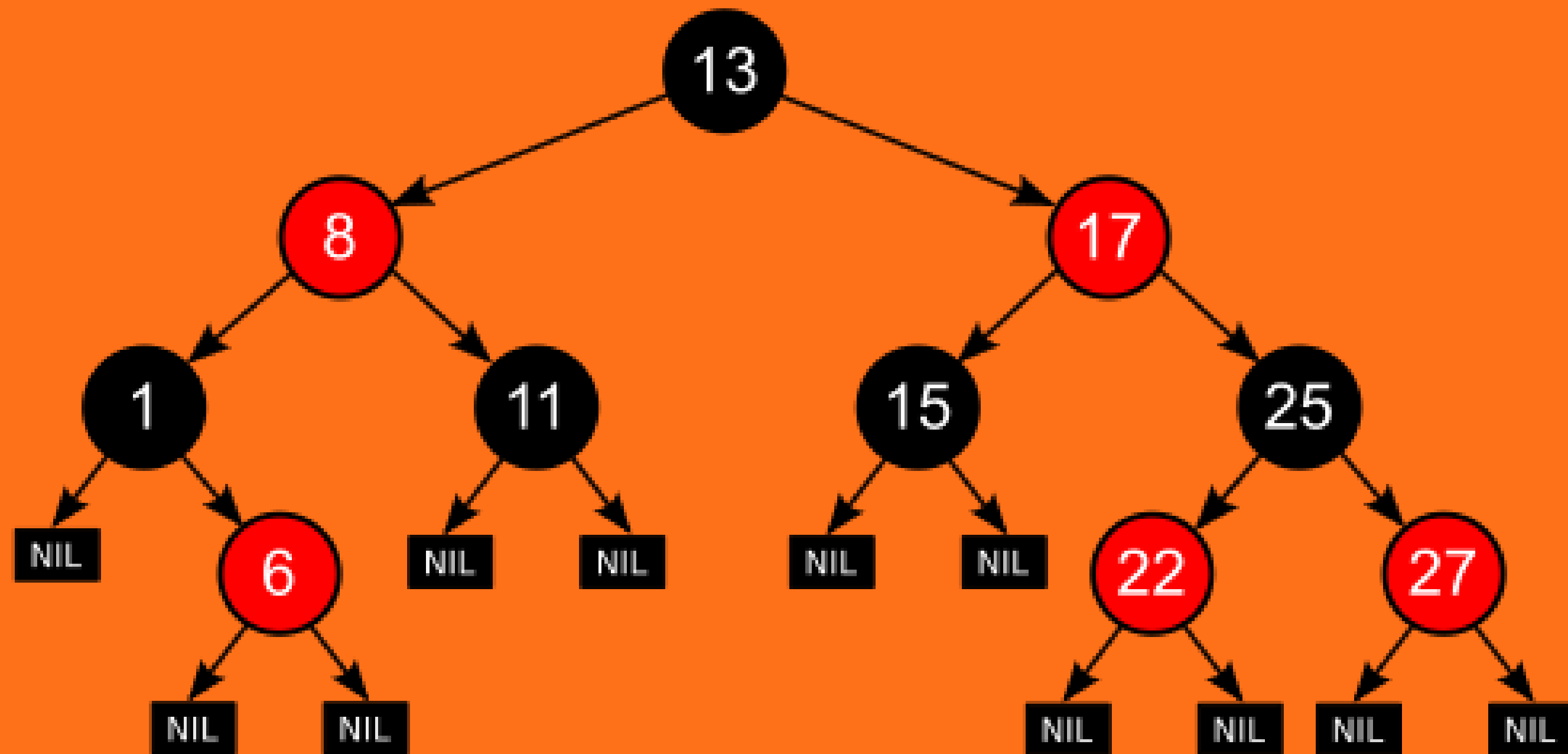
# Definição

- Um tipo de Árvore Balanceada, cada nó pode ter um nó rubro, ou negro;
- Utiliza do sistema de cores para manter o balanceamento;
- $O(\log N)$ ;



# Propriedades

- Todo nó é obrigatoriamente vermelho ou preto;
- A raiz da árvore sempre será negra;
- Por padrão, todo nó nulo é negro;
- Não existem nós vermelhos consecutivos, logo, um nó vermelho sempre terá filhos negros;
- Para todo nó, todos os caminhos descendentes, até os nós folhas contêm o mesmo número de nós pretos;



# AVL vs Rubro-Negra

- AVL, é mais balanceada, por isso tende a ser mais rápida em operações de busca, e mais lenta em operações de inserção e remoção;
- Mesma complexidade algorítmica;
- Para a remoção de um elemento, no pior caso, a Árvore AVL pode exigir  $O(\log N)$  operações, enquanto a Rubro-Negra apenas 3 rotações;
- Para a RN Cada sub-Árvore irmã, pode ter no máximo o dobro da altura, sendo menos restrita;

# Inserção

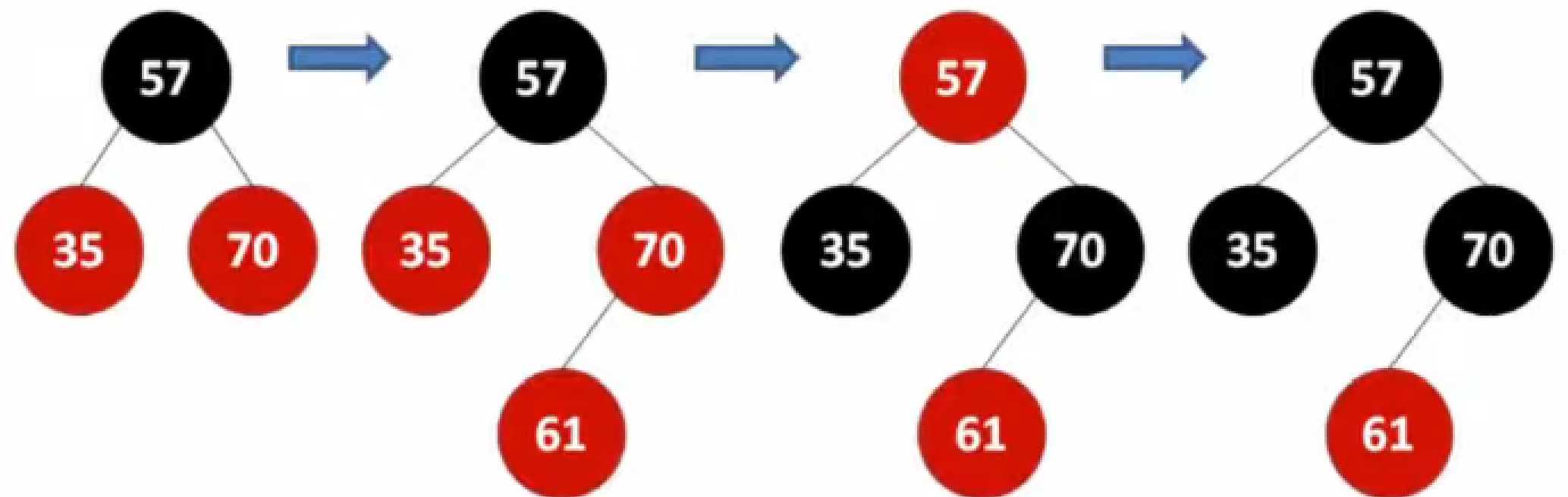
- Por definição, cada nó inserido é Rubro;
- Para inserção, lembre-se:
  - A raiz da árvore sempre é negra;
  - Se o pai de um nó inserido é negro, todas as propriedades se mantêm;
  - Se o pai de um nó inserido é rubro, o balanceamento será efetuado;

# Balanceamento

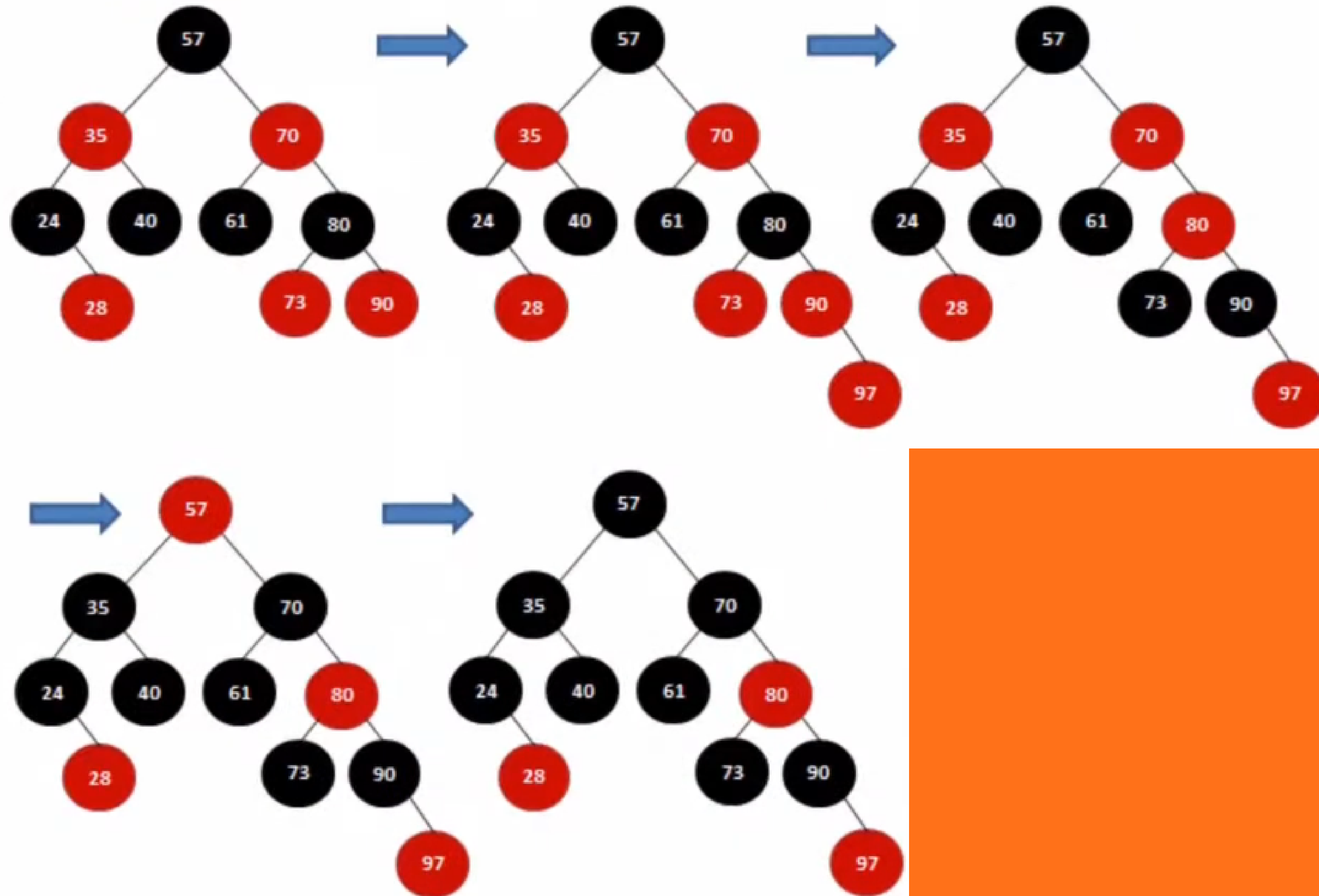
- Feitas para corrigir as propriedades quebradas;
- Troca de cores e "rotações";

## 1 Caso (Pai e Tio Rubros):

Pai vira negro,  
Avô vira Rubro,



Observe a inserção do número 97.

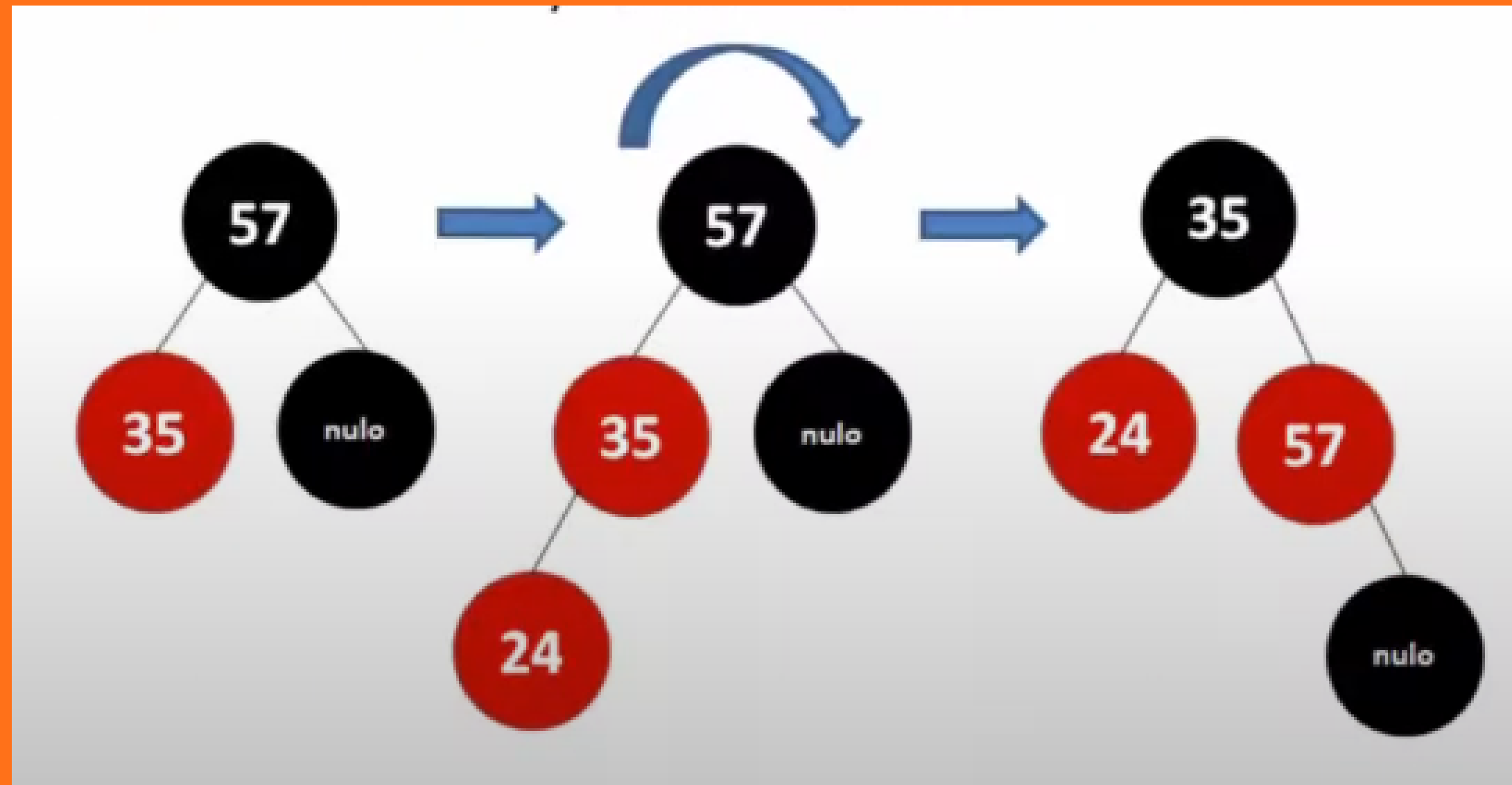




# 2 Caso (Pai Rubro, e Tio Negro):

1 Possibilidade (Esquerda - Esquerda):

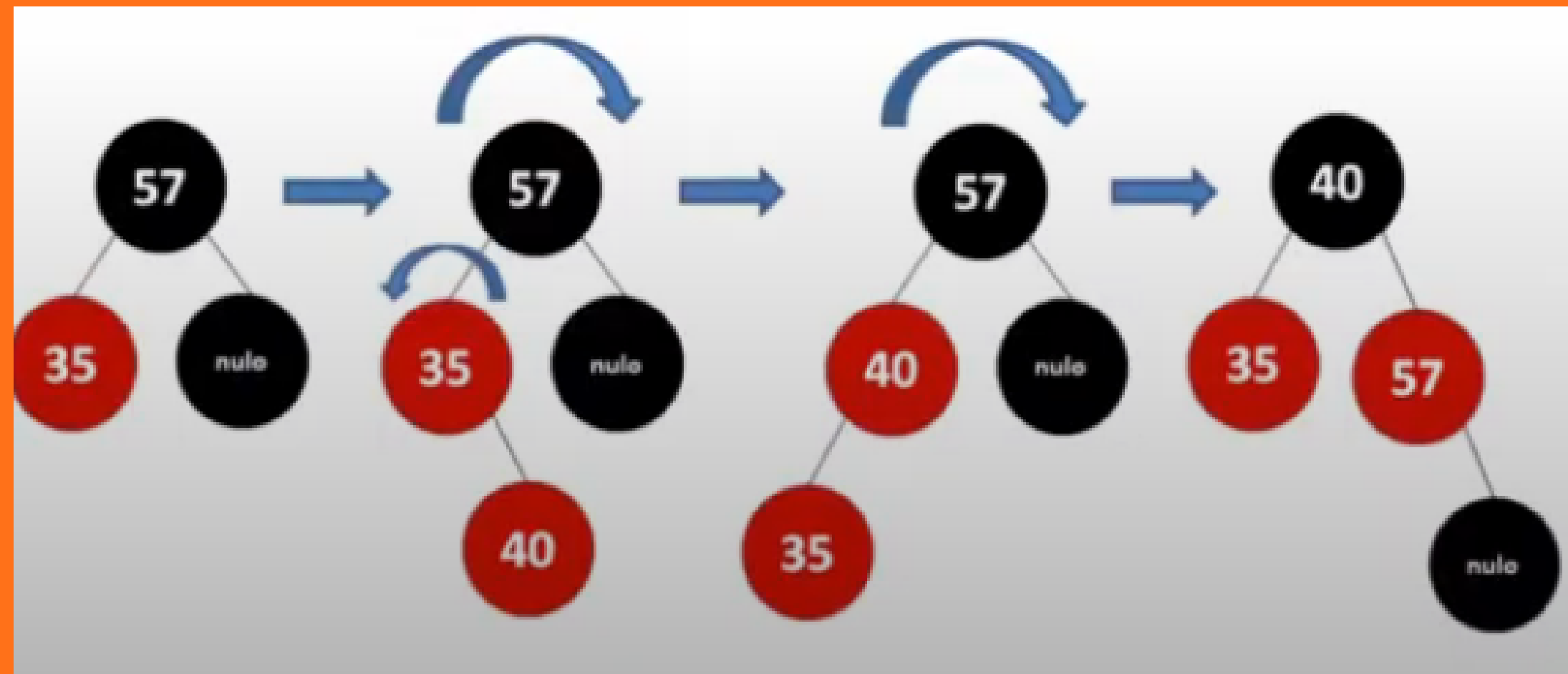
Pai vira negro, Avô vira Rubro, avô rotacionado para direita.



# 2 Caso (Pai Rubro, e Tio Negro):

2 Possibilidade (Esquerda - Direita):

Pai vira negro, Avô vira Rubro, pai rotaciona para esquerda, avô rotacionado para direita;



# Remoção

- Remove-se um nó exatamente como a ABB;
- Para remoção, lembre-se:
  - A raiz da árvore sempre é negra;
  - Se o pai de um nó removido é negro, todas as propriedades se mantêm;
  - Se o pai de um nó removido é rubro, todas as propriedades se mantêm;

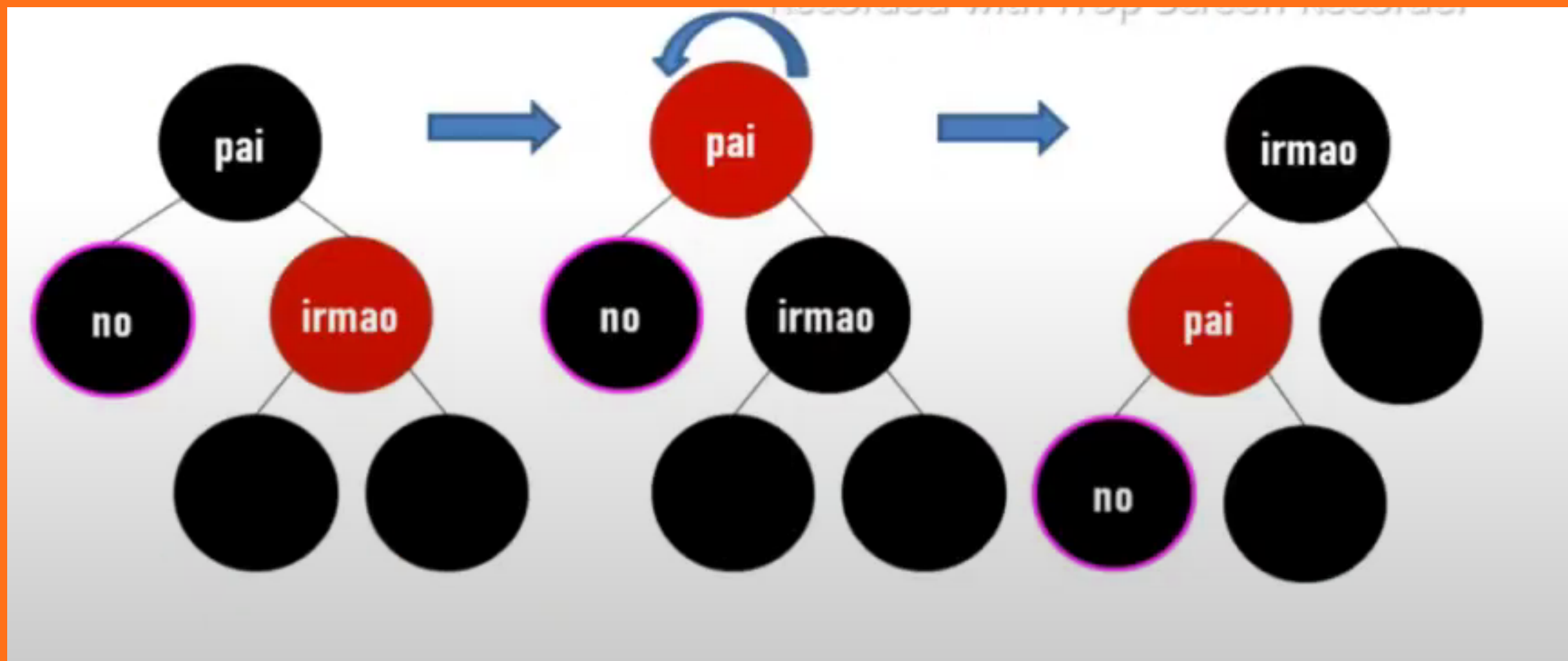
<b>X</b>	<b>Substituto</b>	<b>Resultado</b>
NEGRO	NEGRO	DUPLO NEGRO
NEGRO	NULO NEGRO	NULO DUPLO NEGRO
NEGRO	RUBRO	NEGRO
RUBRO	RUBRO	RUBRO
RUBRO	NULO NEGRO	NULO NEGRO

Seja *no* DUPLAMENTE NEGRO:

- Se *no* é raiz, troque a cor para NEGRA.
- Senão, vamos olhar para o *irmao* e para o *pai* de *no*

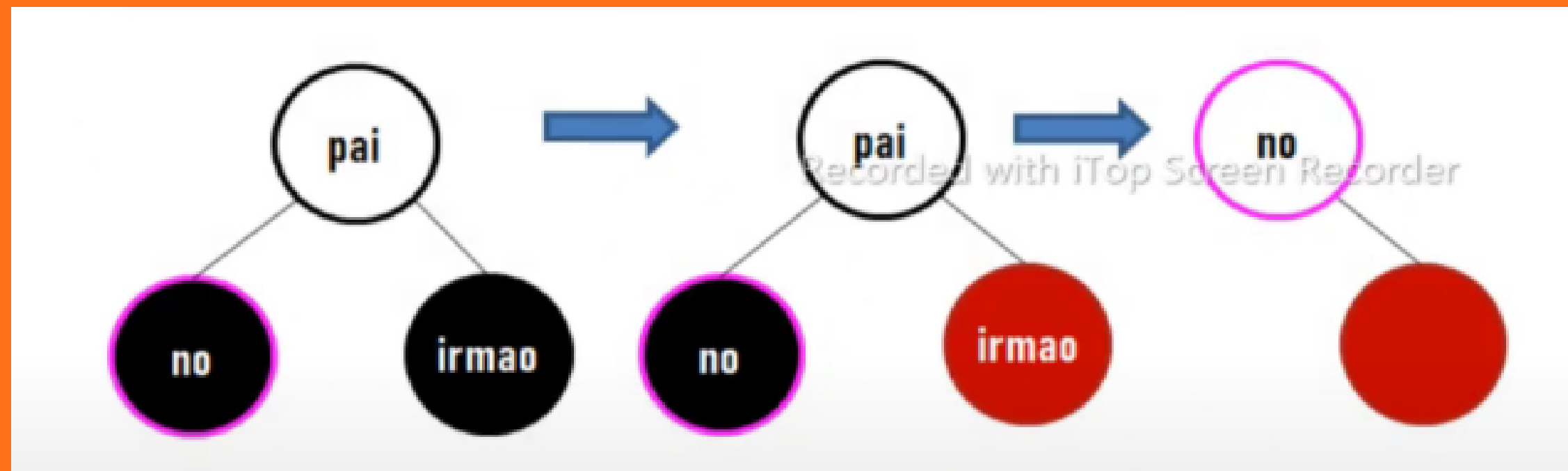
# 1 Caso :

- Necessariamente, é acompanhado por outros casos;
- No irmão ao DUPLAMENTE negro é rubro;
- Pai vira Rubro, o irmão vira negro.



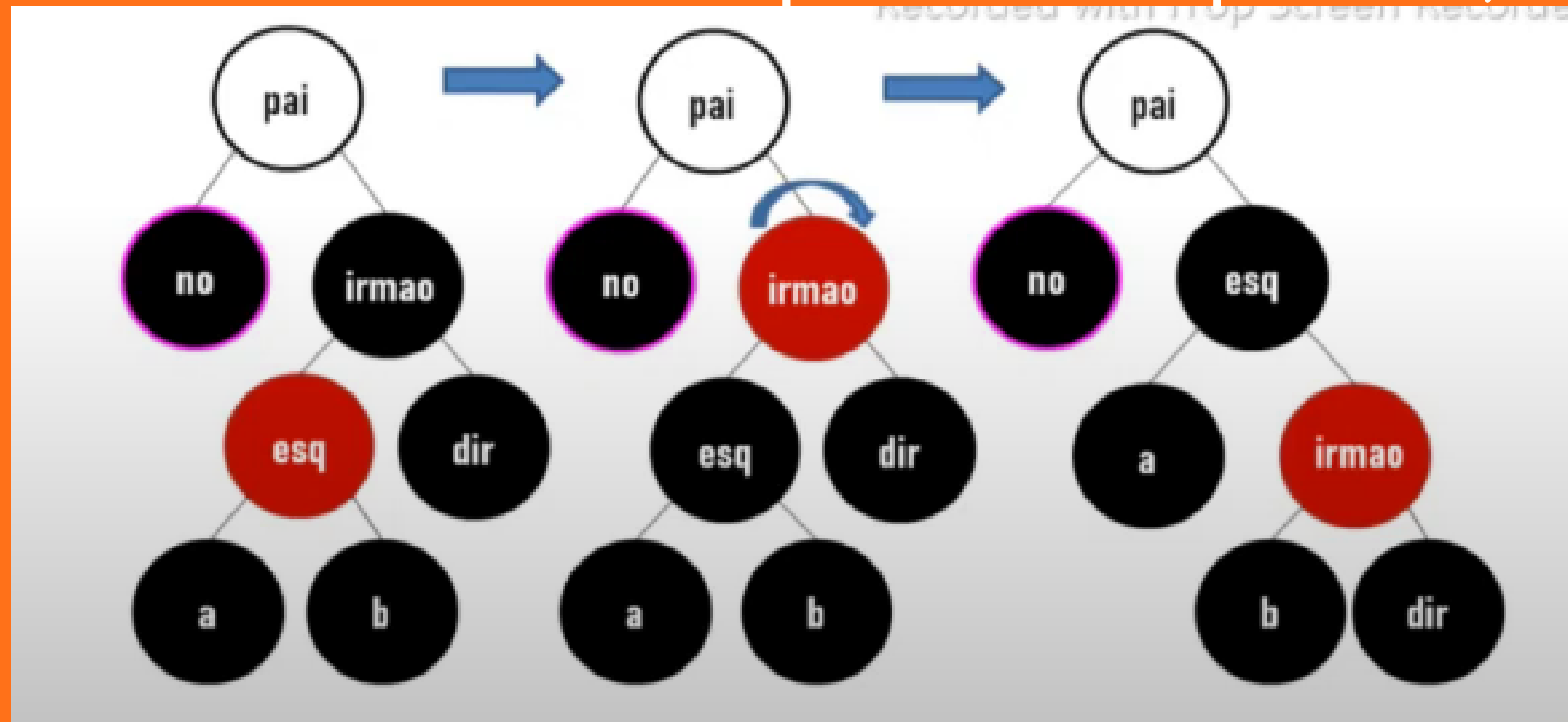
## 2 Caso :

- No irmão ao DUPLAMENTE e seus filhos são negros;
- O irmão vira rubro;
- Nó troca de lugar com o pai, e assume sua cor;



# 3 Caso :

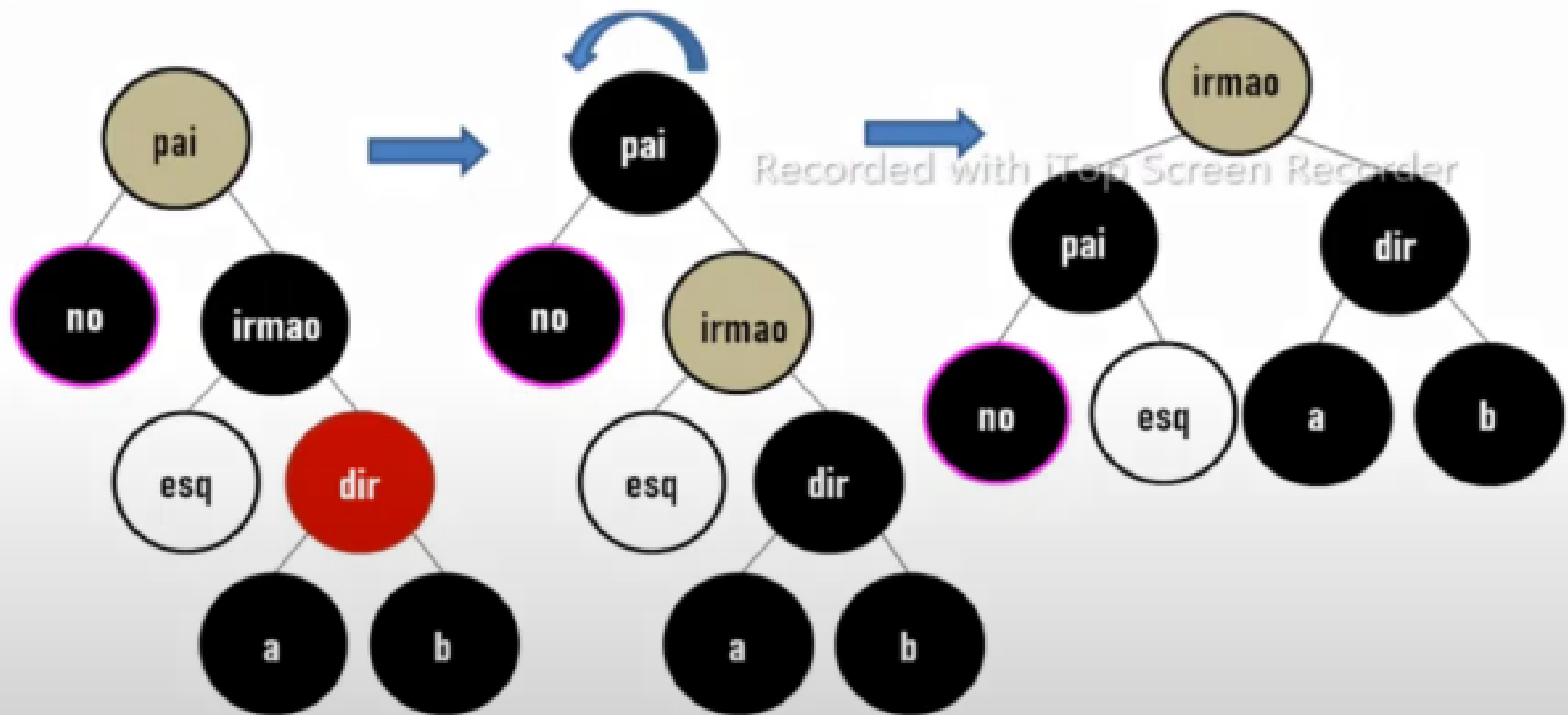
- Nó irmão ao DUPLAMENTE e apenas seu filho direito são negros;
- Necessariamente ocasiona no caso 4;
- O irmão se torna Rubro e é rotacionado para a esquerda;



# 4 Caso :

- No irmão ao DUPLAMENTE é NEGRO seu filho esquerdo é RUBRO;
- Irmão copia a cor do pai;
- Pai e filho direito do irmão se tornam NEGROS;
- Rotacione o Pai a direita;
- Se a raiz da árvore for RUBRA, troque para negra;





### CASO 1:

- Sempre ocasiona o CASO 2 ou CASO 3 ou CASO 4.
- 1-2, 1-3-4, 1-4.

### CASO 2:

- Pode ocorrer sozinho, porém as verificações propagam para o ancestral.

### CASO 3:

- Sempre ocasiona o CASO 4.
- 3-4.

### CASO 4:

- Pode ocorrer sozinho; e
- Após sua execução o balanceamento está concluído.