

# Unidade VI:

## Árvores Alvinegras

Prof. Max do Val Machado



**PUC Minas**

Instituto de Ciências Exatas e Informática  
Curso de Ciência da Computação

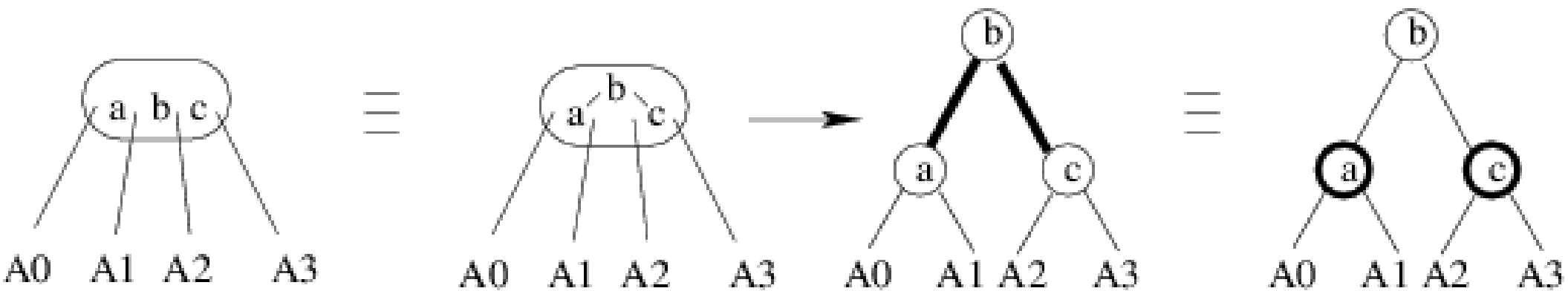
- Estrutura de dados mais eficiente de representar as árvores-2.3.4, evitando o desperdício de memória
- Substitui a representação múltipla de nós por uma representação única contendo os atributos: elemento, apontadores esq e dir e um bit de cor

- Simula a hierarquia 2.3.4, colorindo as ligações entre os nós de duas formas:
  - Se os elementos de dois nós pertencem ao mesmo nó da 2.3.4 (elementos gêmeos), ele será preto (traço grosso)
  - Se eles estiverem em nós diferentes da árvore inicial, ele será branco (traço fino)

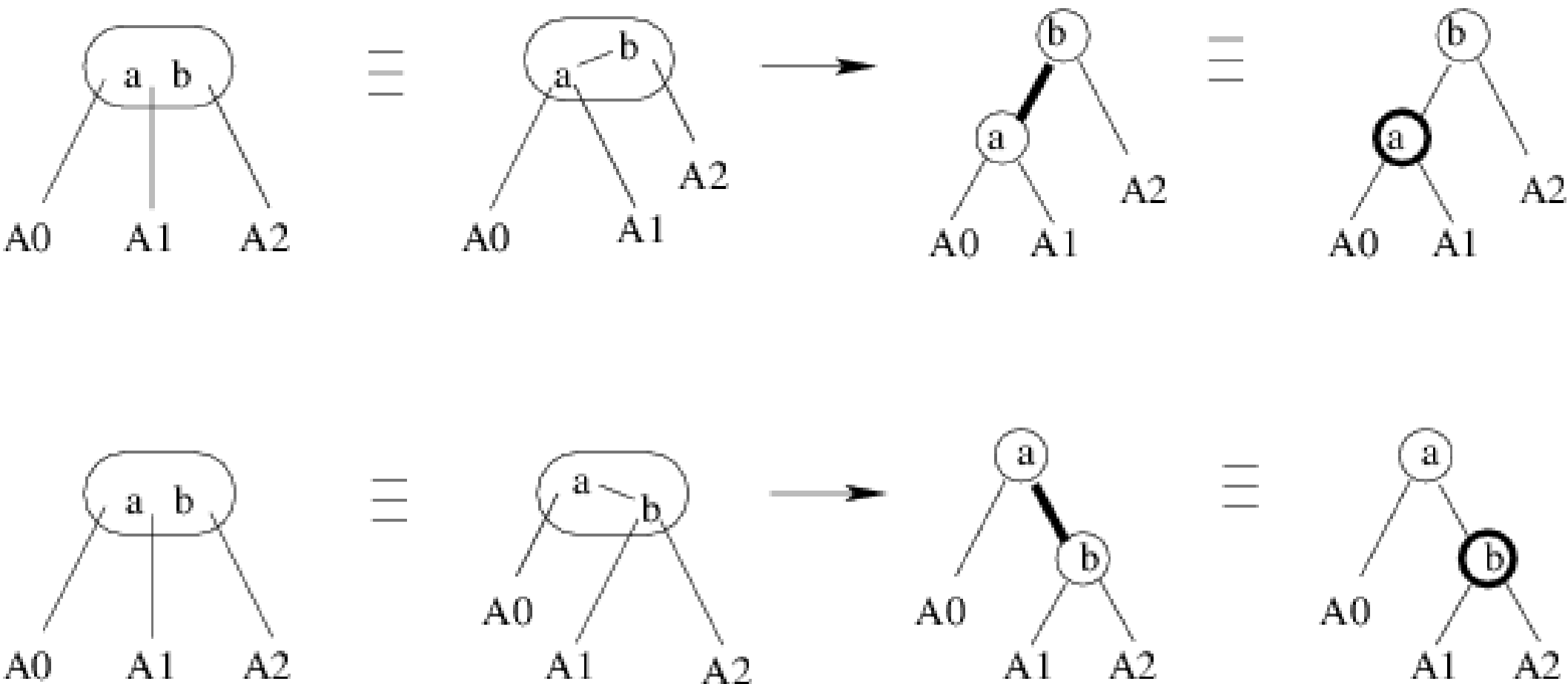
# Introdução

- Na verdade, colorimos os nós em vez das arestas
- A atribuição de cores é feita de forma que cada nó tem a cor da ligação que aponta para ele
- Assim, um nó será preto se, e somente se, seu elemento for gêmeo ao de seu pai

# Exemplo de 4-nó

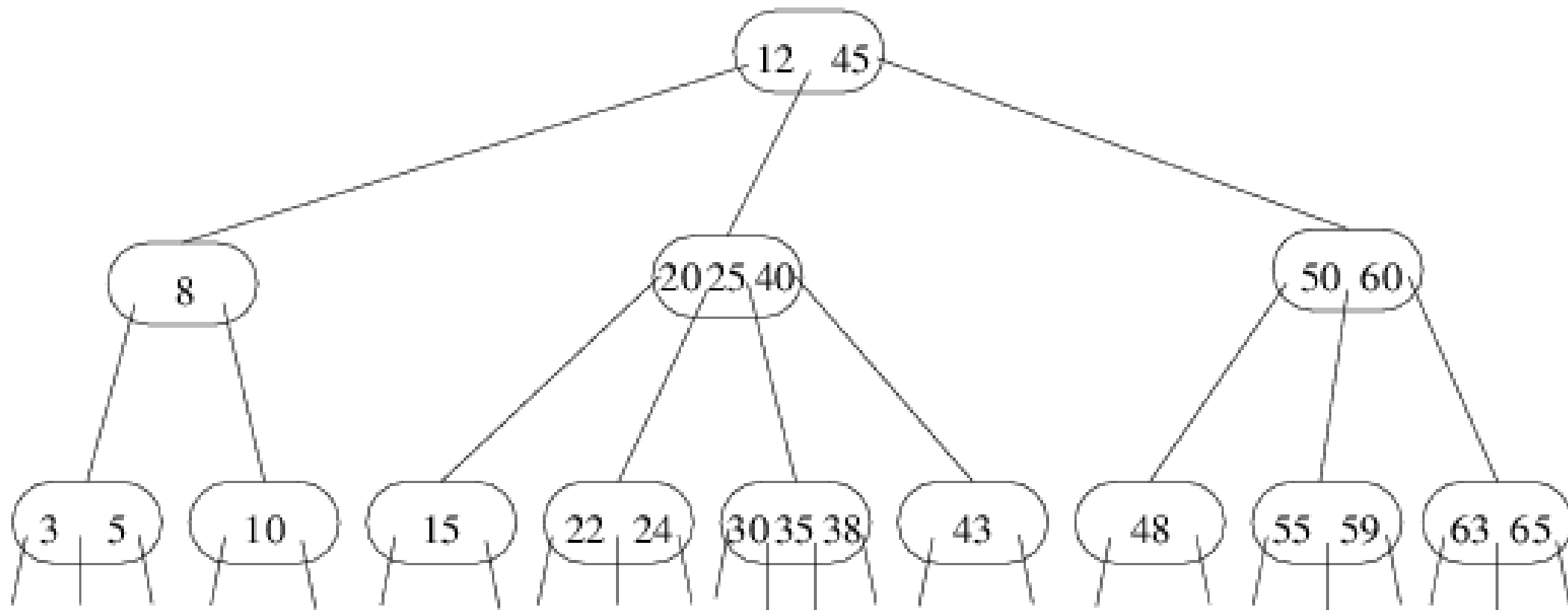


# Os Dois Exemplos de 3-nó

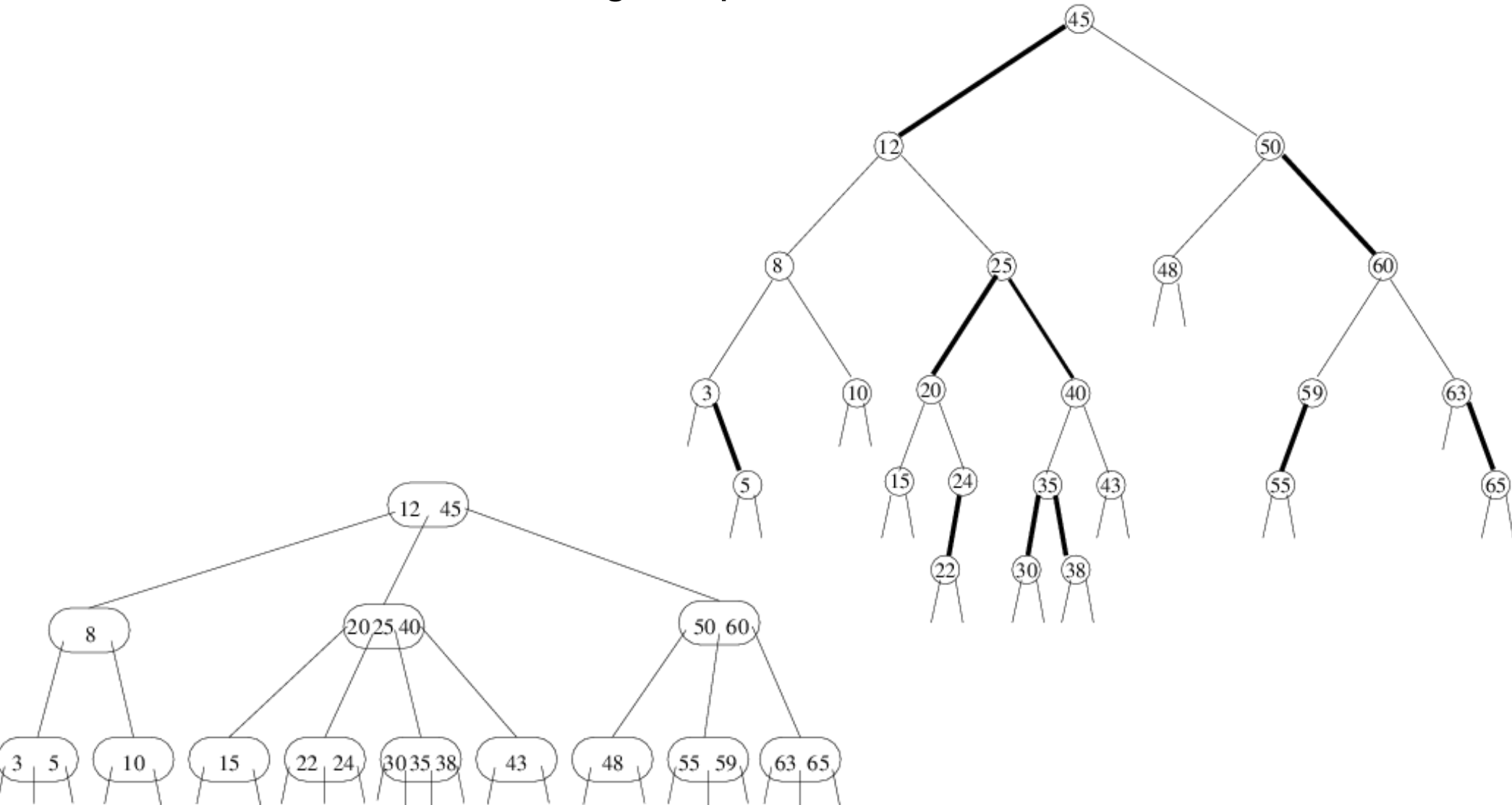


## Exercício

- Encontre uma árvore alvinegra equivalente à 2.3.4 abaixo:



- Encontre uma árvore alvinegra equivalente à 2.3.4 abaixo:





# Exercício

- Dada uma 2.3.4 qualquer com altura  $h$ , qual é a altura máxima da alvinegra correspondente?
- Dada uma 2.3.4 qualquer com altura  $h$ , qual é a altura mínima da alvinegra correspondente?

- Dada uma 2.3.4 qualquer com altura  $h$ , qual é a altura máxima da alvinegra correspondente?

- Resposta: Quando todos os nós da 2.3.4 são 4 ou 3-nó, temos  $2h$

- Dada uma 2.3.4 qualquer com altura  $h$ , qual é a altura mínima da alvinegra correspondente?

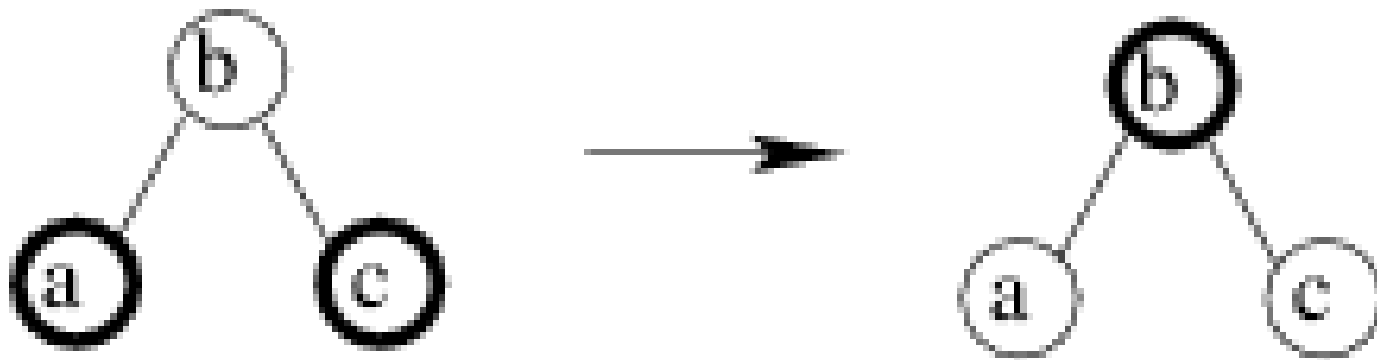
- Resposta: Quando todos os nós da 2.3.4 são 2-nó, temos  $h$

- Qual é o número máximo de ligações pretas consecutivas?
- Qual é a relação entre as arestas da 2.3.4 com as brancas da alvinegra

- Qual é o número máximo de ligações pretas consecutivas?
  - Resposta: Não existem ligações pretas consecutivas
- Qual é a relação entre as arestas da 2.3.4 com as brancas da alvinegra
  - Resposta: As brancas da alvinegra são exatamente as arestas da 2.3.4

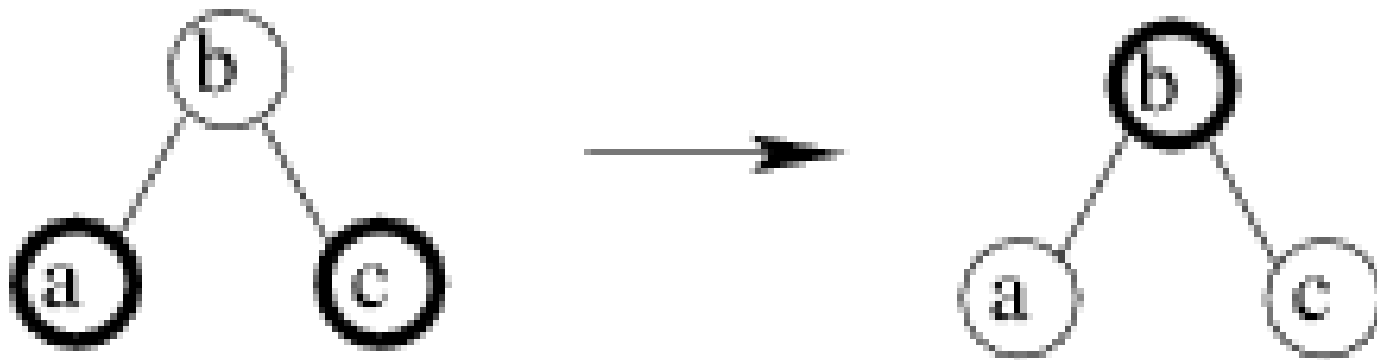
- Desconsidera as cores e utiliza o mesmo procedimento das árvores binárias
- No pior caso, tem-se  $\Theta(\lg n)$  comparações

- Pode ser feita simulando a inserção com fragmentação na descida em uma 2.3.4
- Ideia básica: Fragmentar um 4-nó é o mesmo que inverter as cores dos elementos gêmeos desse nó



- Algum efeito colateral?

- Pode ser feita simulando a inserção com fragmentação na descida em uma 2.3.4
- Ideia básica: Fragmentar um 4-nó é o mesmo que inverter as cores dos elementos gêmeos desse nó



- Algum efeito colateral? **Sim, podemos ter dois nós pretos consecutivos**

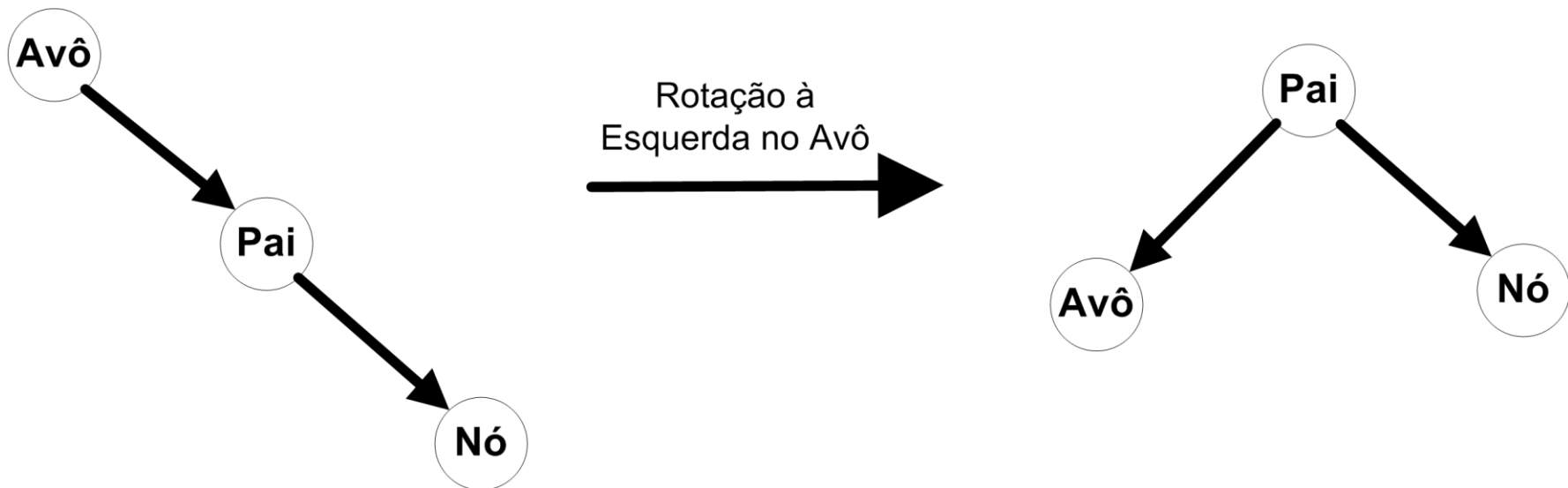
E agora José?



# Inserção

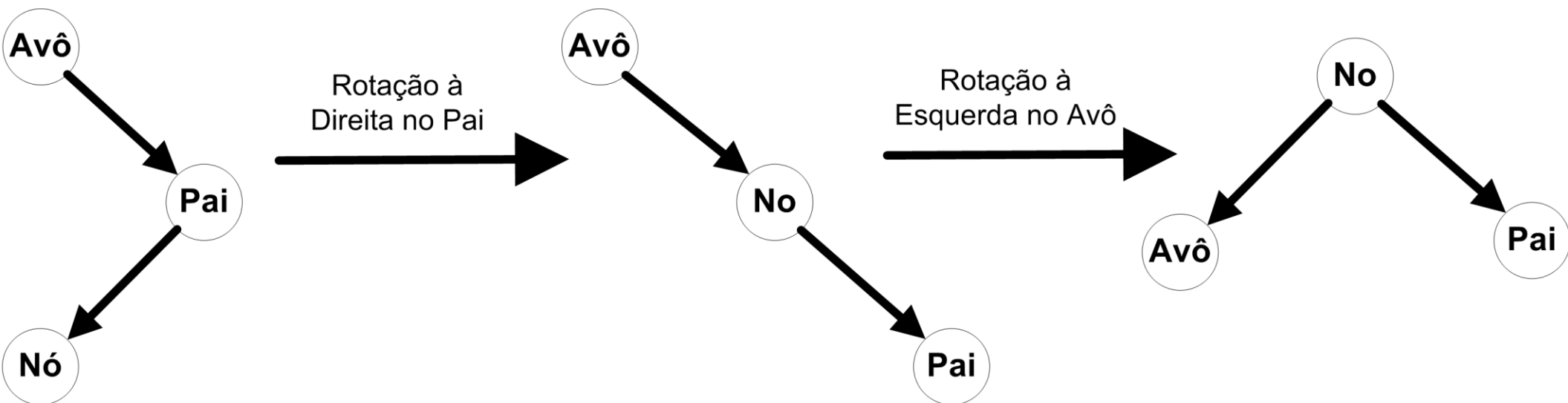
- Após a inversão de cores, se o pai do nó também for preto, rotacionamos o avô considerando o alinhamento entre avô, pai e nó

- Após a inversão de cores, se o pai do nó também for preto, rotacionamos o avô considerando o alinhamento entre avô, pai e nó
- Se  $\text{avô} < \text{pai}$  and  $\text{pai} < \text{nó}$ , rotação à esquerda (avô)



## Inserção

- Após a inversão de cores, se o pai do nó também for preto, rotacionamos o avô considerando o alinhamento entre avô, pai e nó
- Se  $\text{avô} < \text{pai}$  and  $\text{pai} > \text{nó}$ , rotação direita (pai) – esquerda (avô)



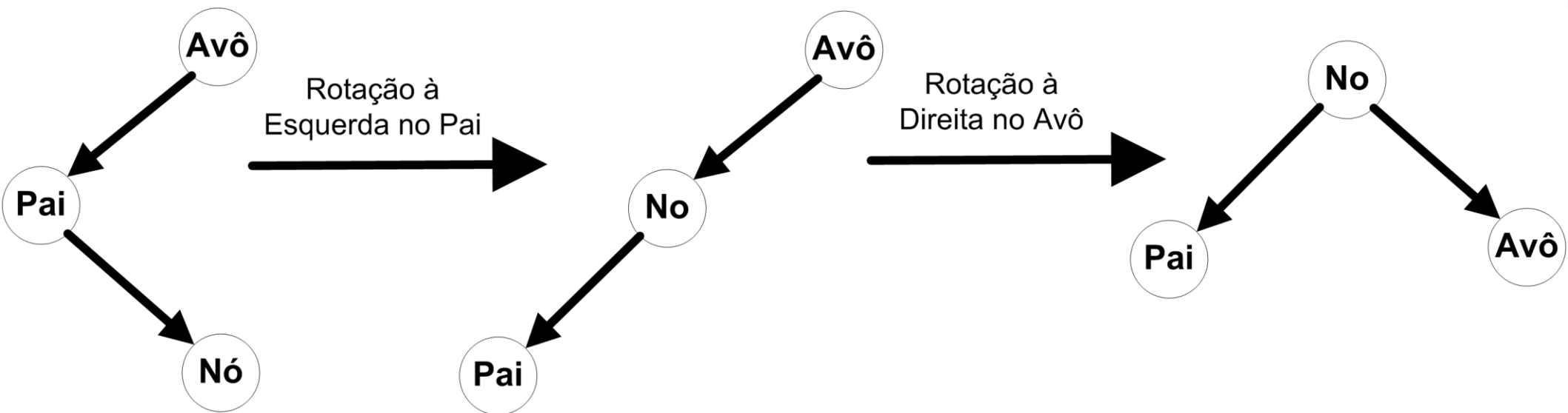
## Inserção

- Após a inversão de cores, se o pai do nó também for preto, rotacionamos o avô considerando o alinhamento entre avô, pai e nó
- Se  $\text{avô} > \text{pai}$  **and**  $\text{pai} > \text{nó}$ , rotação à direita (avô)



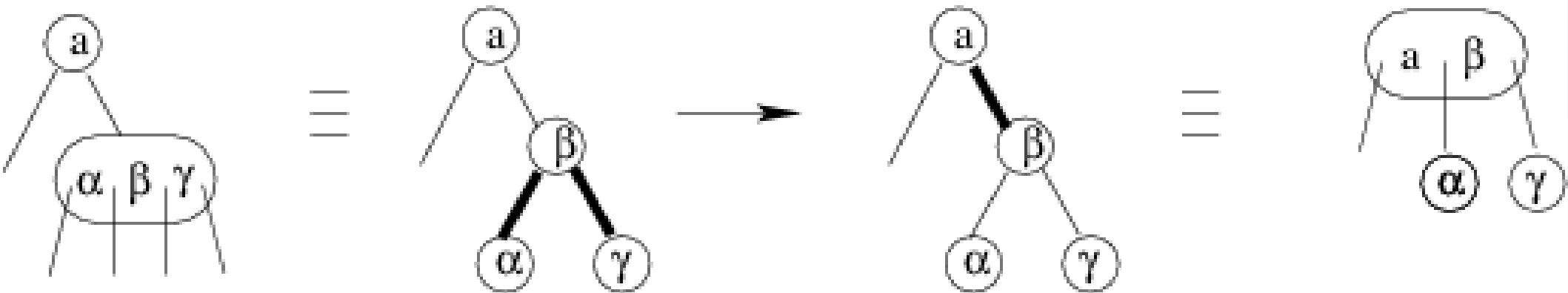
## Inserção

- Após a inversão de cores, se o pai do nó também for preto, rotacionamos o avô considerando o alinhamento entre avô, pai e nó
- Se  $\text{avô} > \text{pai}$  **and**  $\text{pai} < \text{nó}$ , rotação esquerda (pai) – direita (avô)



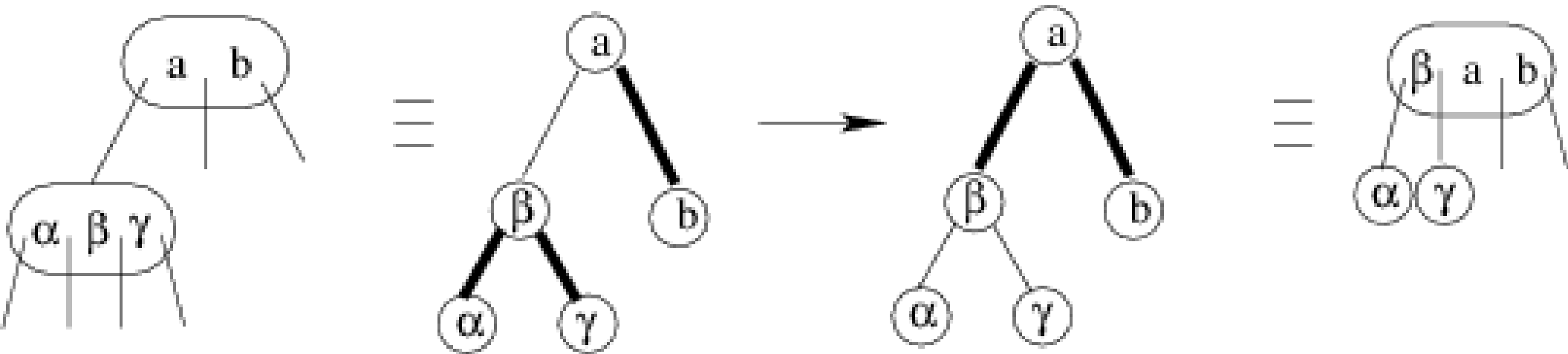
## Inserção

- Após a inversão de cores, se o pai do nó for branco, não temos problemas como, por exemplo:



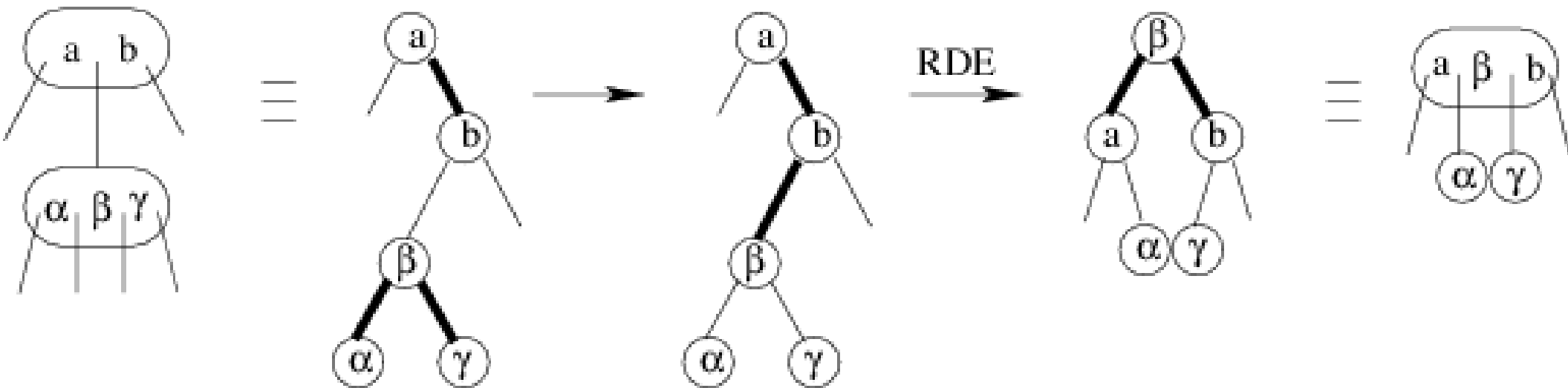
## Inserção

- Após a inversão de cores, se o pai do nó for branco, não temos problemas como, por exemplo:



# Inserção

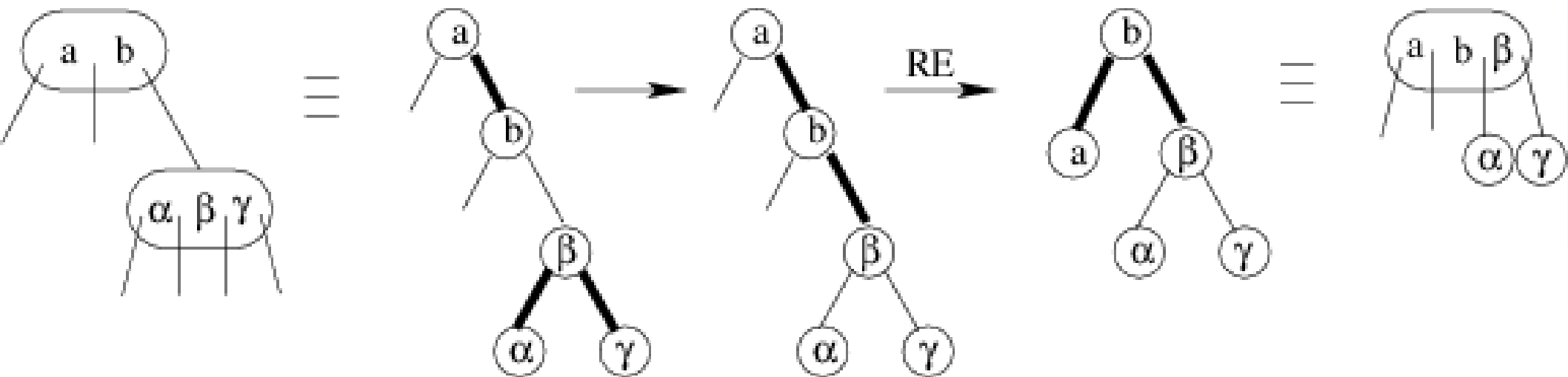
- No exemplo abaixo, após a inversão de cores, como avô < pai and pai > nó, temos uma rotação direita (pai) – esquerda (avô)





# Inserção

- No exemplo abaixo, após a inversão de cores, como  $avô < pai$  and  $pai < nó$ , temos uma rotação para a esquerda (avô)



# Ideia Básica da Inserção

- Consiste em procurar a folha em que o novo elemento será inserido
- Se um nó tiver dois filhos pretos (4-nó na 2.3.4), invertemos as cores desse nó e de seus filhos (exceto a raiz que continuará branca porque ela não tem pai para ser gêmeo)
- Após a inversão de cores, se o pai também for preto, rotacionamos o avô considerando o alinhamento entre avô, pai e nó
- Após a rotação, o elemento central fica branco e seus novos filhos, pretos

# Ideia Básica da Inserção

- Continuar a descida até chegar em uma folha
- A inserção sempre acontece em uma folha que ficará preta porque o novo elemento é gêmeo (na 2.3.4) do pai e do irmão (se esse existir)

# Exemplo de Inserção

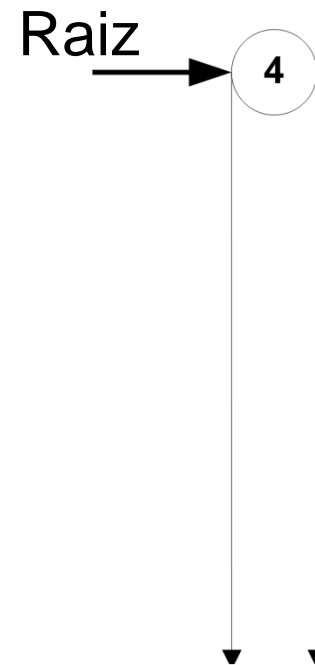
- Crie uma árvore alvinegra através de inserções sucessivas do 4, 35, 10, 13, 3, 30, 15, 12, 7, 40 e 20 respectivamente

# Exemplo de Inserção

- Inserindo o 4

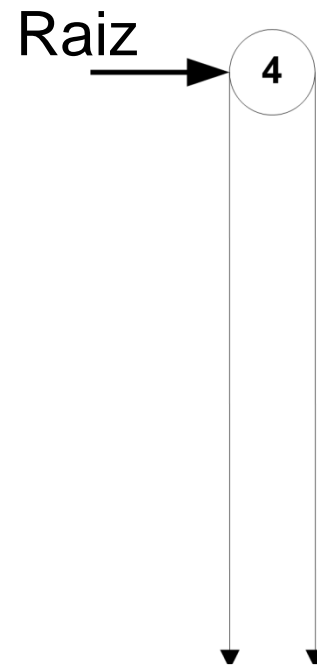
## Exemplo de Inserção

- Inserido 4 que será branco pois é o da ``raiz``



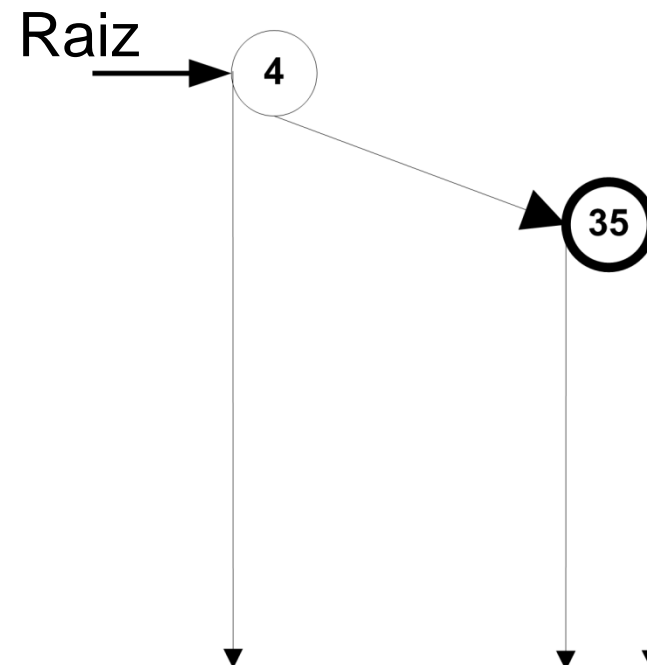
# Exemplo de Inserção

- Inserindo o 35



## Exemplo de Inserção

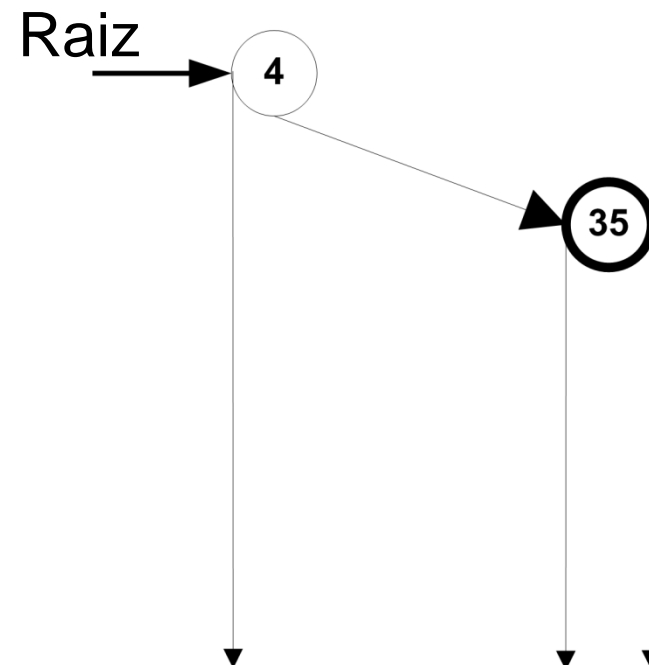
- Inserido o 35 (que será preto como todas as folhas)





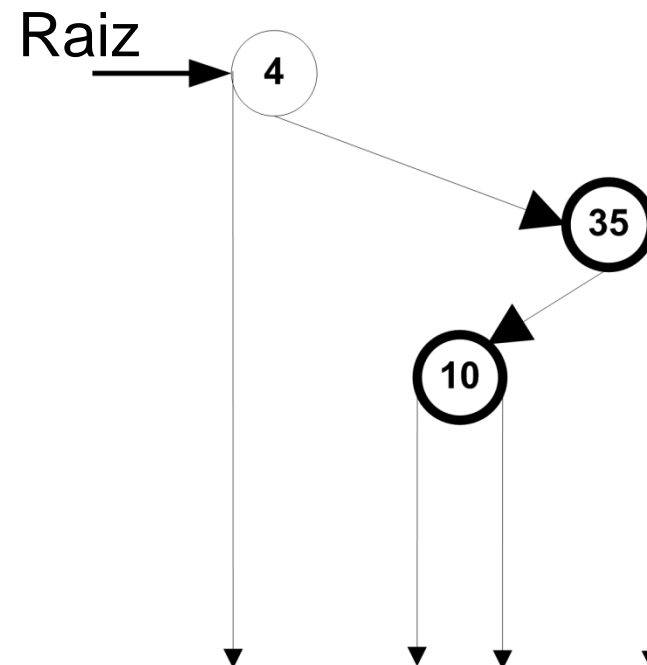
## Exemplo de Inserção

- Inserindo o 10



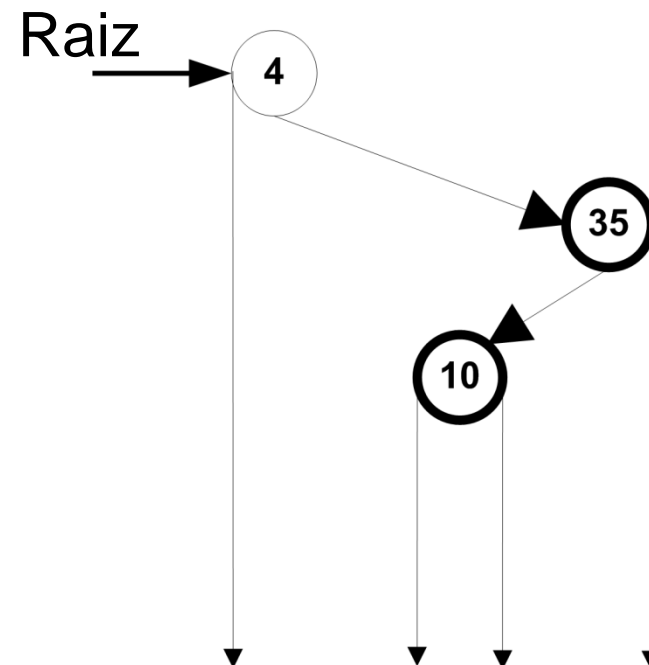
## Exemplo de Inserção

- Inserido o 10



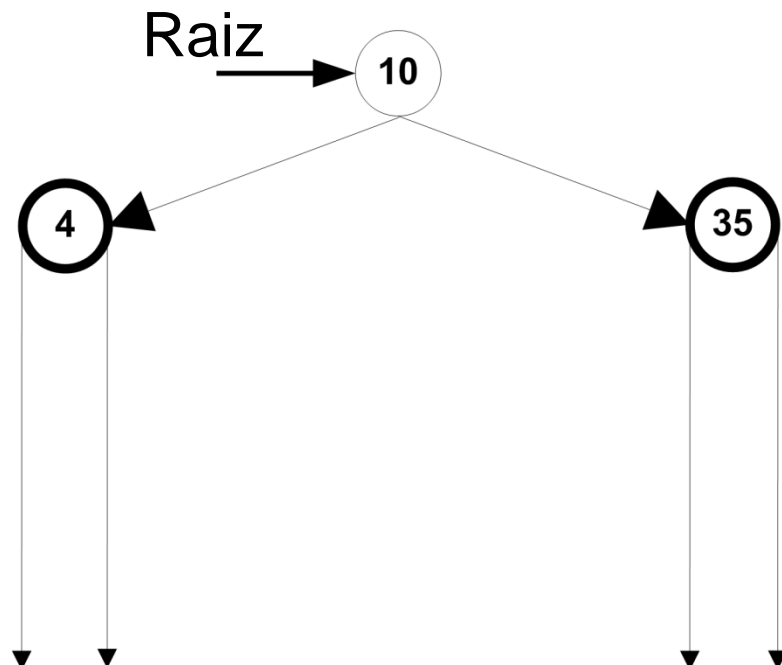
## Exemplo de Inserção

- Rotação Direita - Esquerda no 4



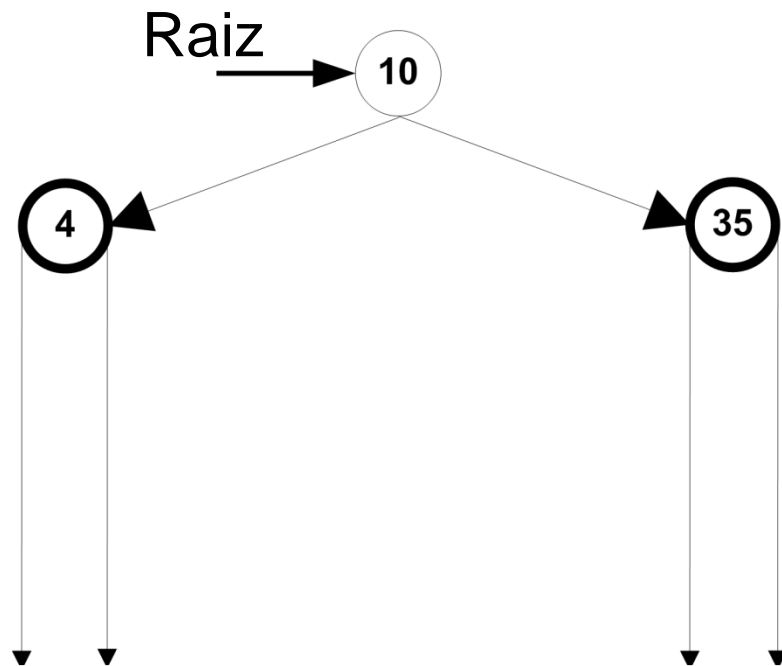
## Exemplo de Inserção

- Efetuada a Rotação Direita - Esquerda no 4



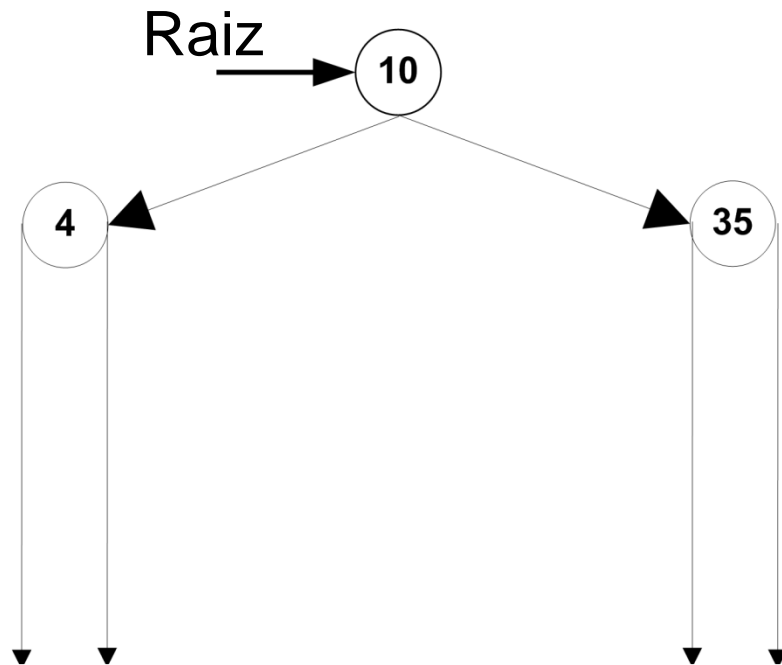
## Exemplo de Inserção

- Inserindo o 13



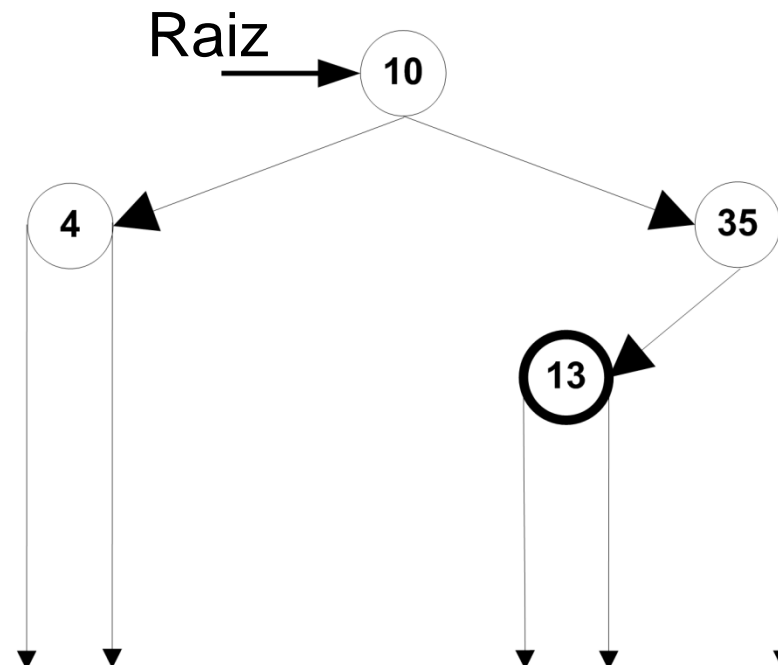
## Exemplo de Inserção

- Antes de inserir o 13, como o 10 tem dois filhos pretos, invertamos as cores do 4, 10 e 35 (10 continua branco porque é raiz)



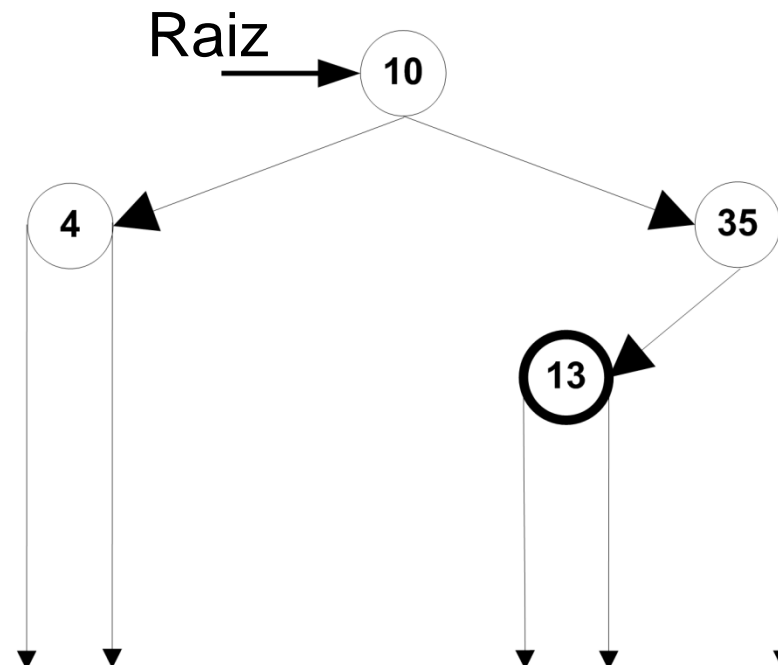
## Exemplo de Inserção

- Inserido o 13



## Exemplo de Inserção

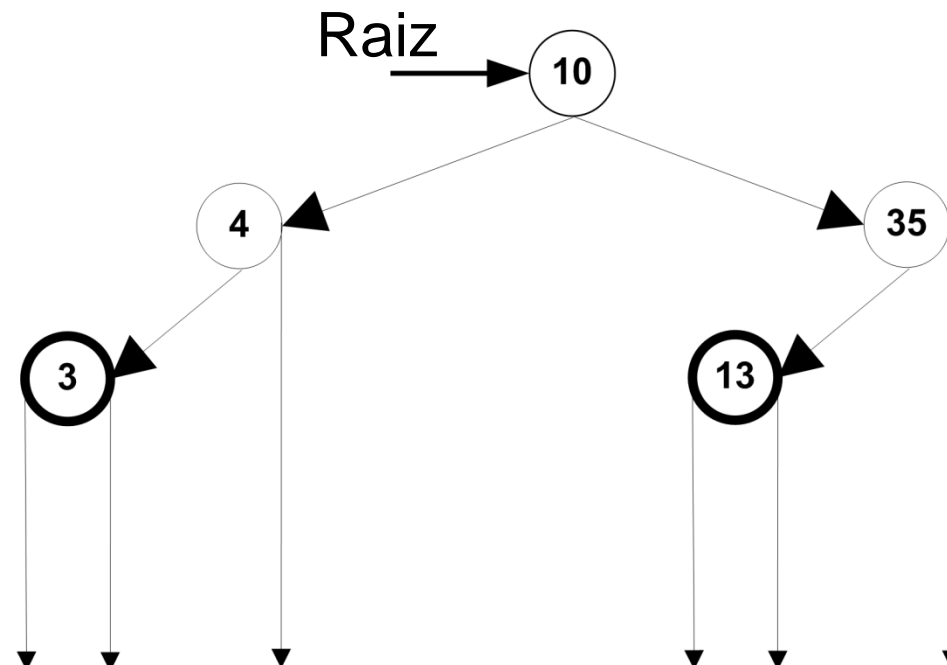
- Inserindo o 3





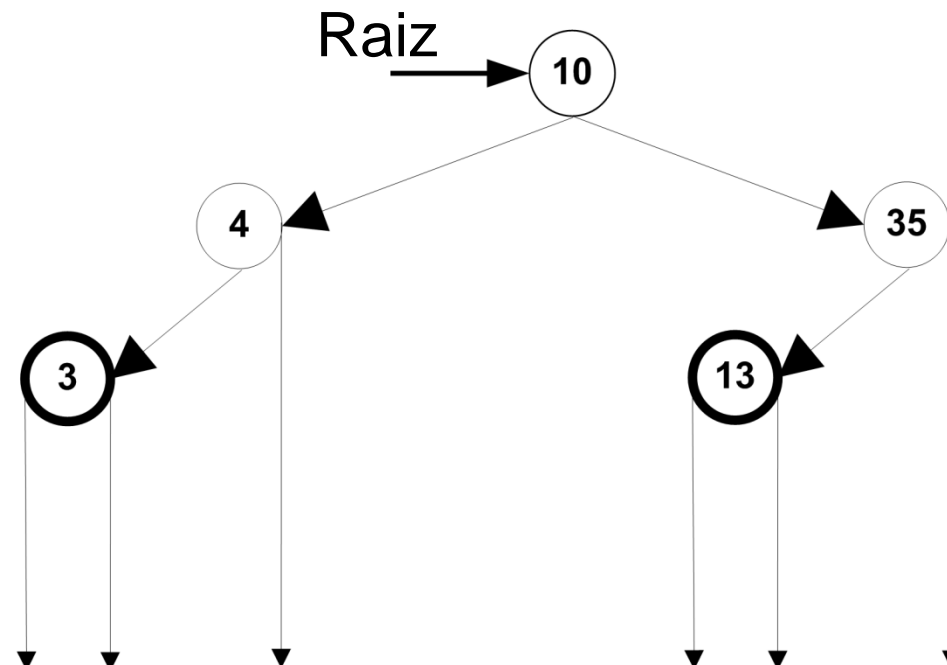
## Exemplo de Inserção

- Inserido o 3



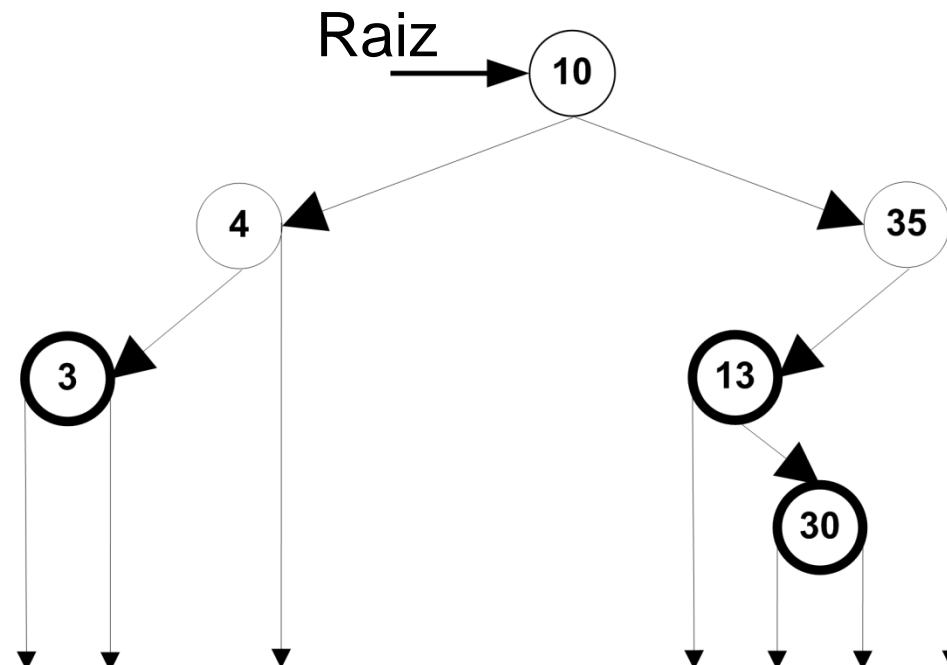
## Exemplo de Inserção

- Inserindo o 30



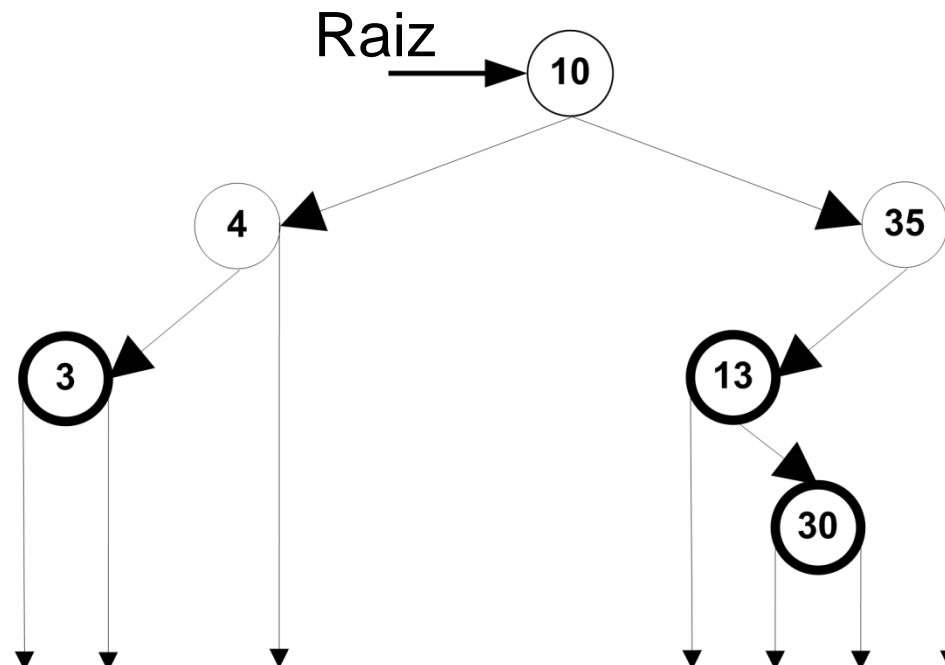
## Exemplo de Inserção

- Inserido o 30



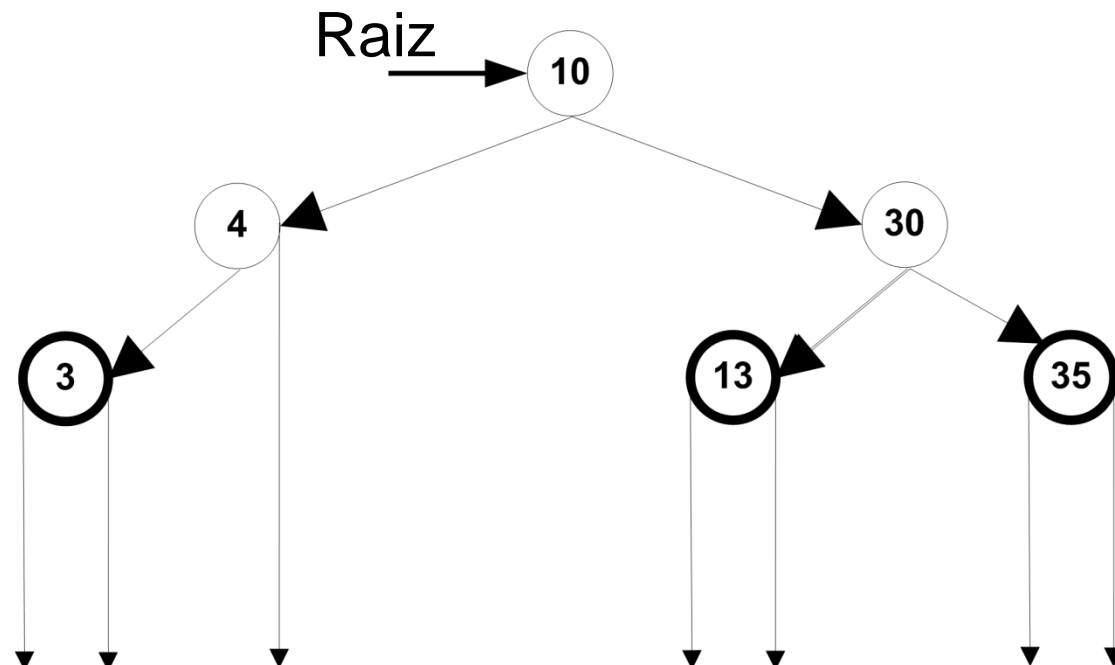
## Exemplo de Inserção

- Como temos dois nós pretos, rotacionamos o 13 para a direita e o 35 para a esquerda



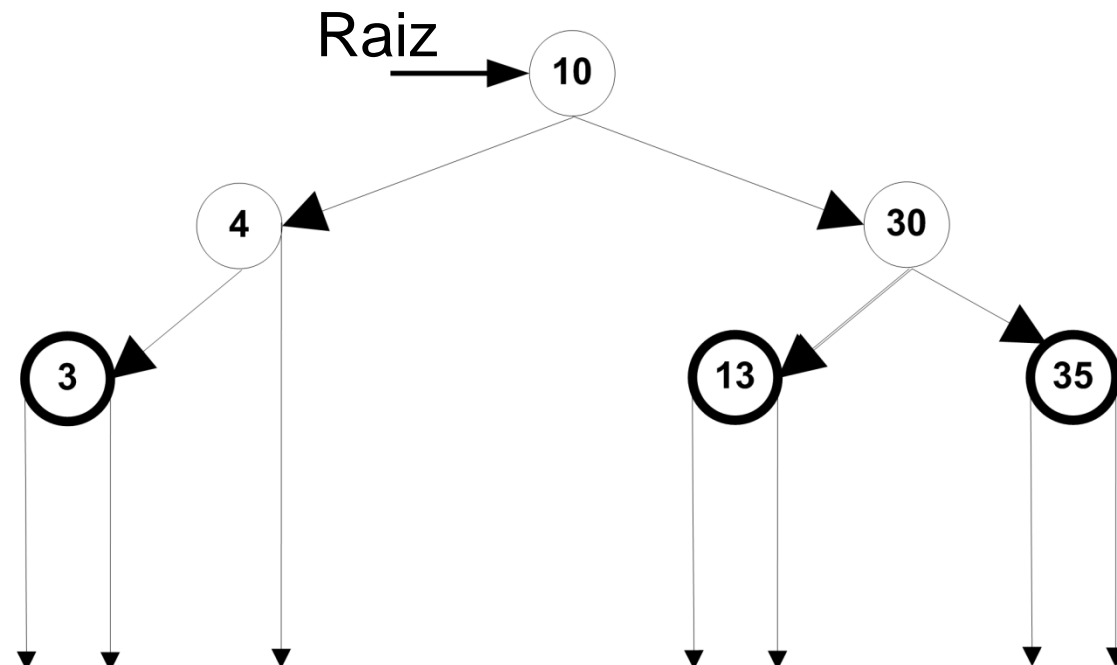
# Exemplo de Inserção

- Como temos dois nós pretos, rotacionamos o 13 para a direita e o 35 para a esquerda e, em seguida, vamos reorganizar as cores



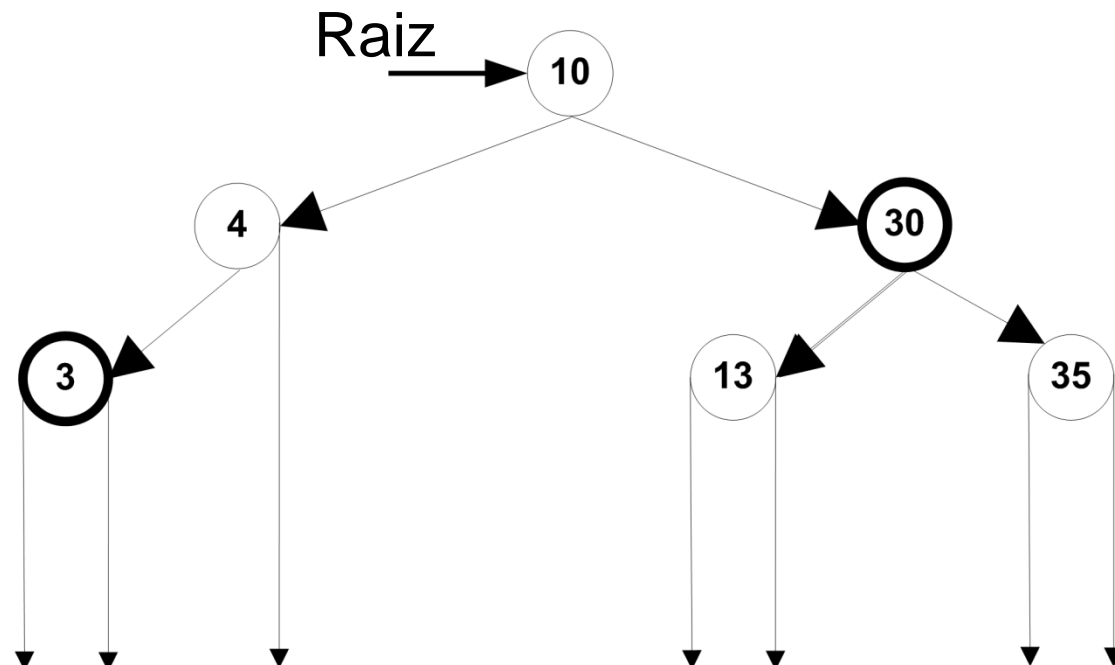
## Exemplo de Inserção

- Inserindo o 15, temos um nó (30) com dois filhos pretos



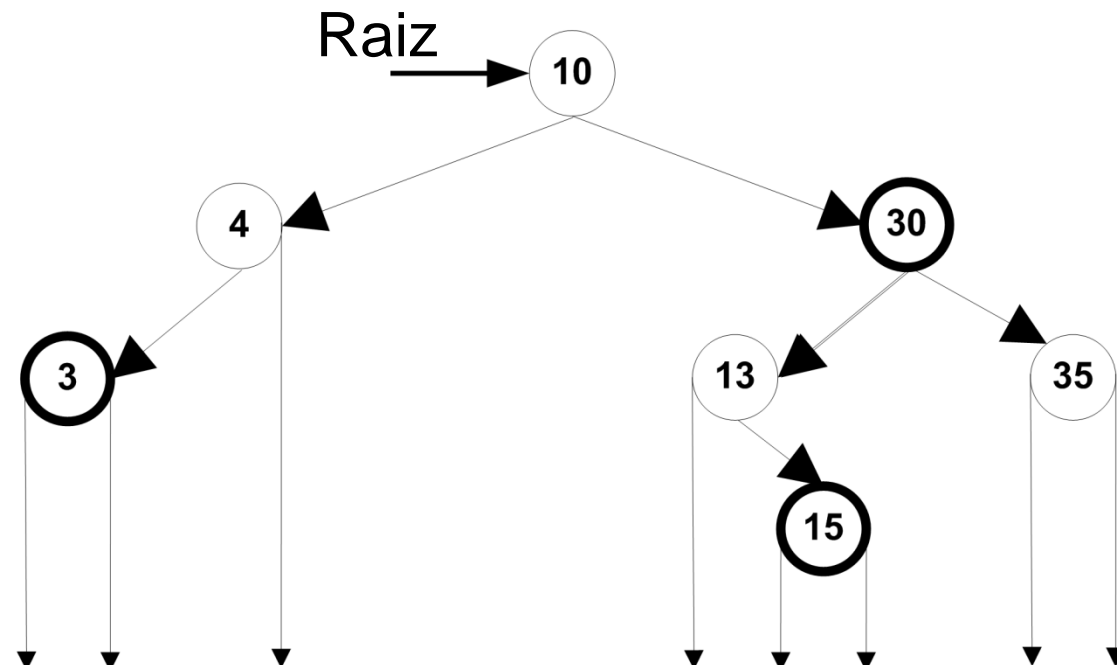
## Exemplo de Inserção

- Inserindo o 15, temos um nó (30) com dois filhos pretos



## Exemplo de Inserção

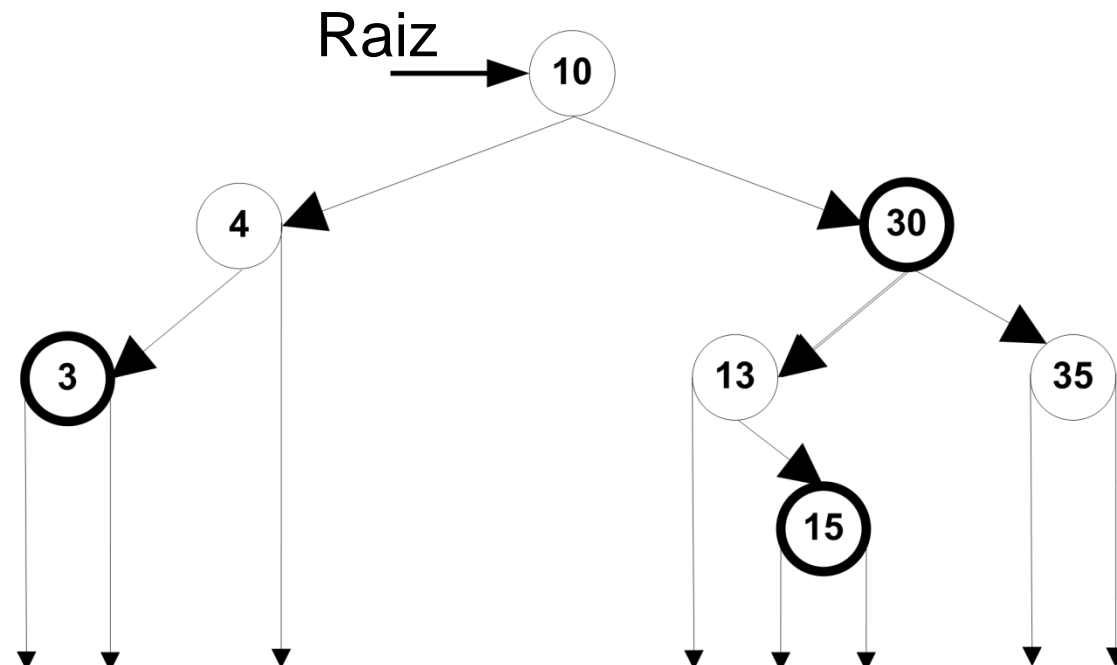
- Inserido o 15





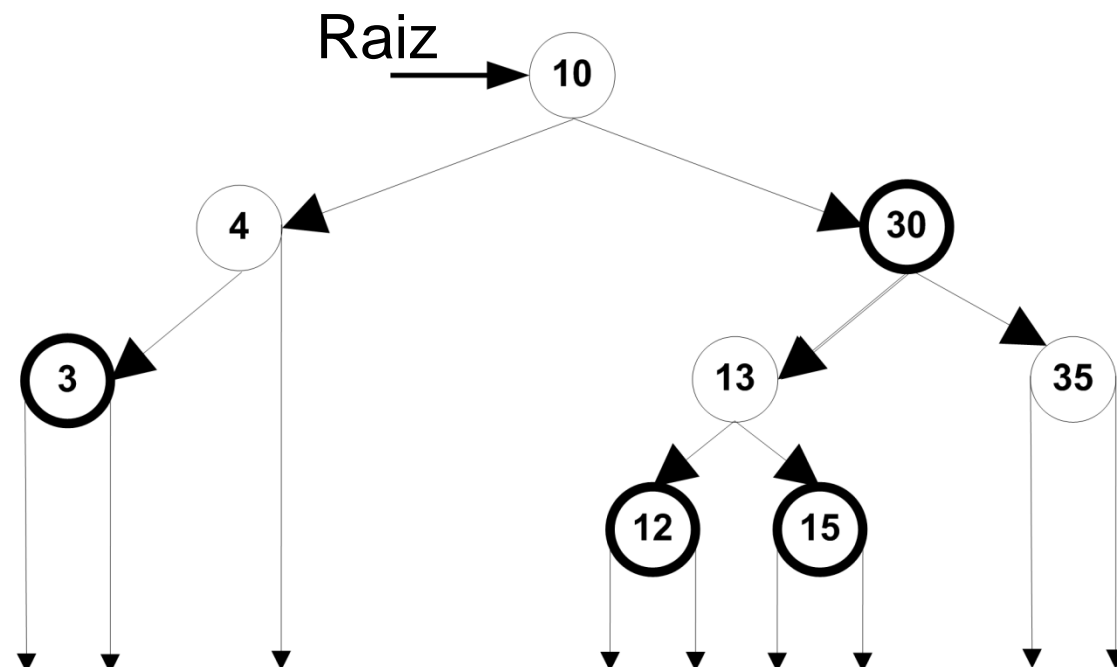
## Exemplo de Inserção

- Inserindo o 12



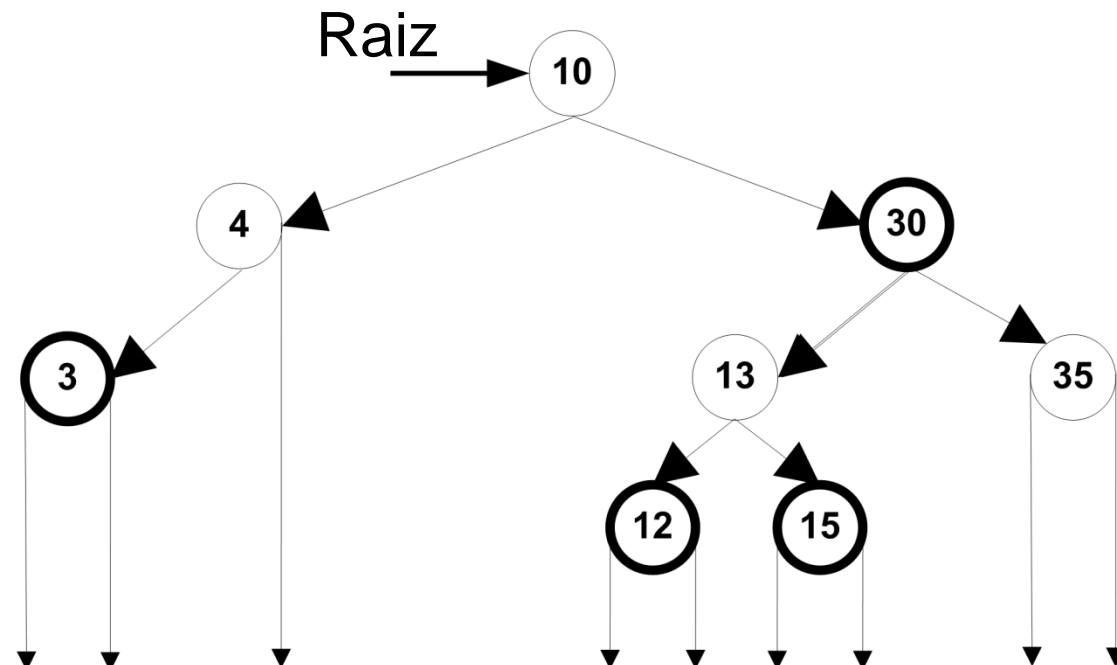
## Exemplo de Inserção

- Inserido o 12



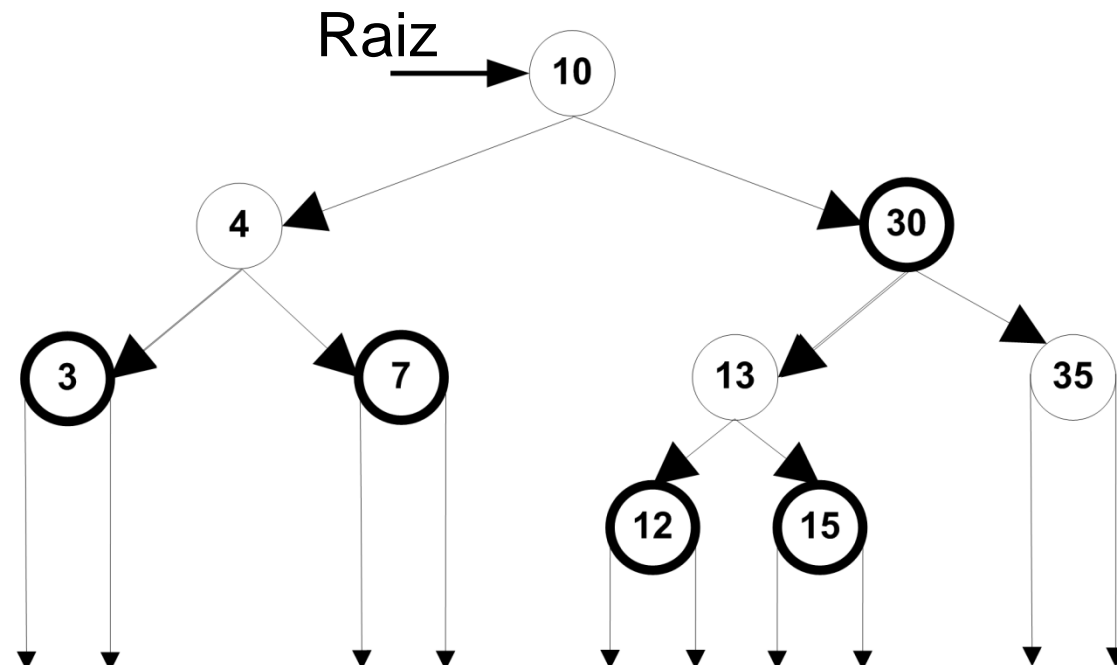
## Exemplo de Inserção

- Inserindo o 7



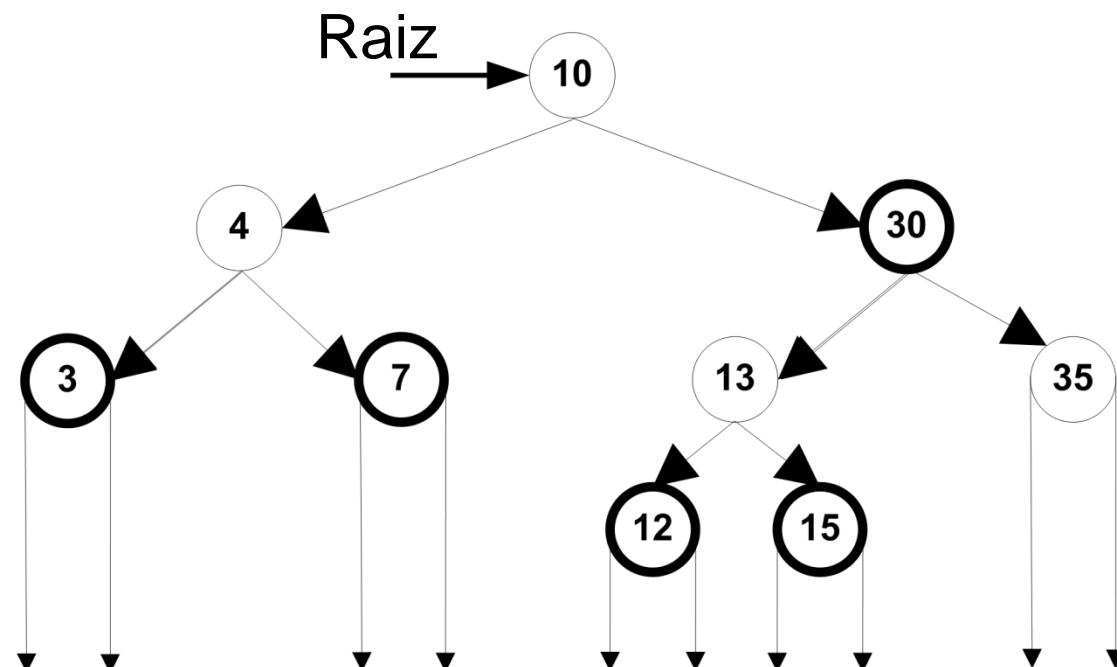
## Exemplo de Inserção

- Inserido o 7



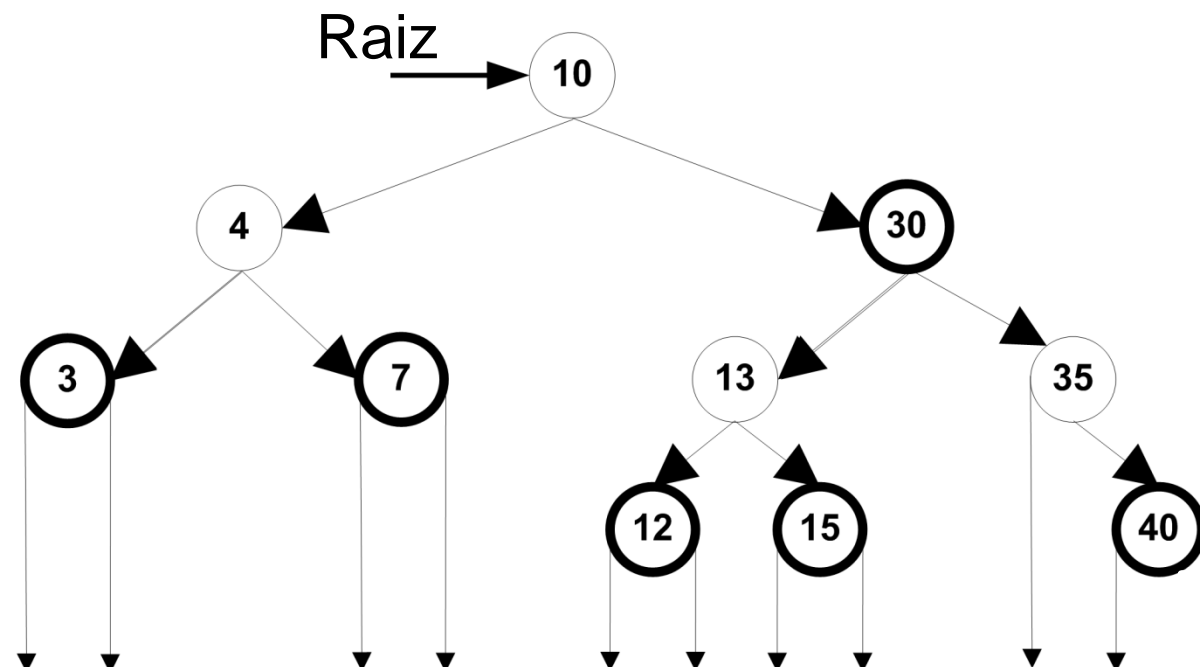
## Exemplo de Inserção

- Inserindo o 40



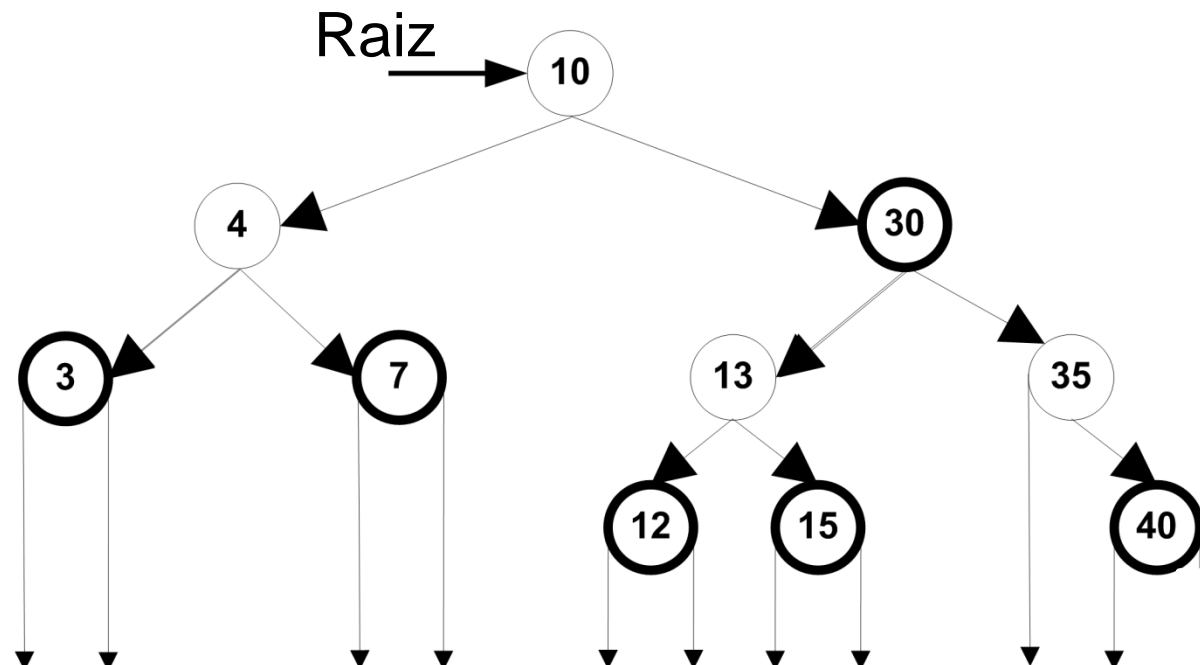
## Exemplo de Inserção

- Inserido o 40



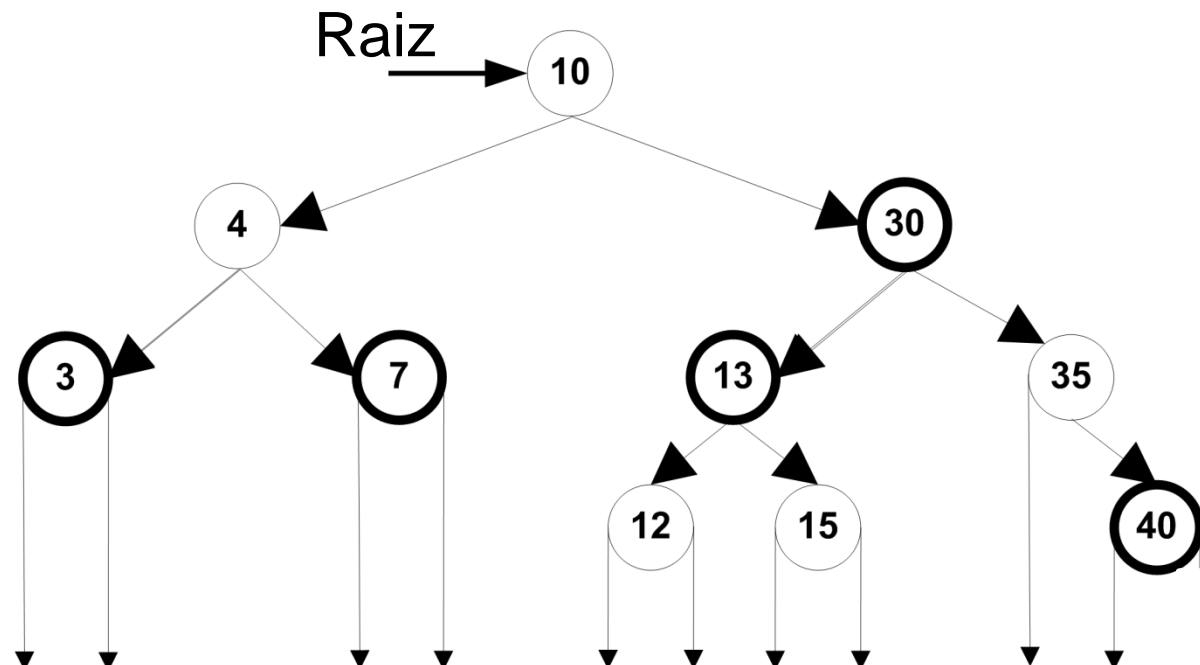
## Exemplo de Inserção

- Inserindo o 20, temos o 13 com dois filhos pretos



## Exemplo de Inserção

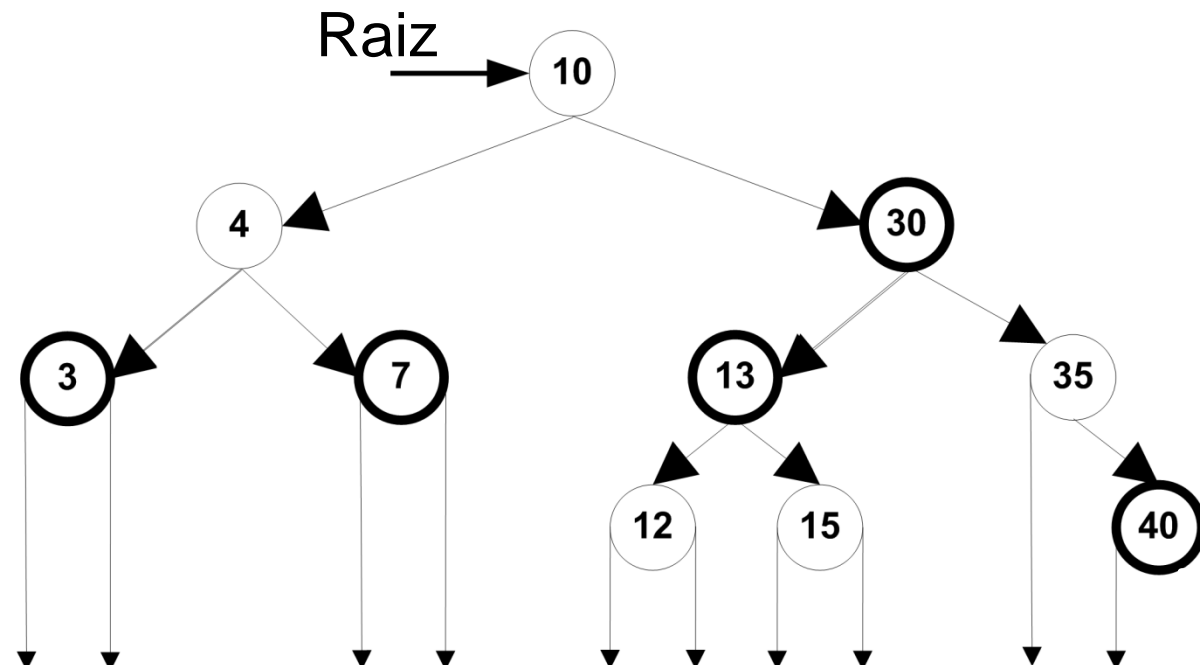
- Invertamos as cores do nó 13 e, conseqüentemente, ficamos como dois nós pretos consecutivos





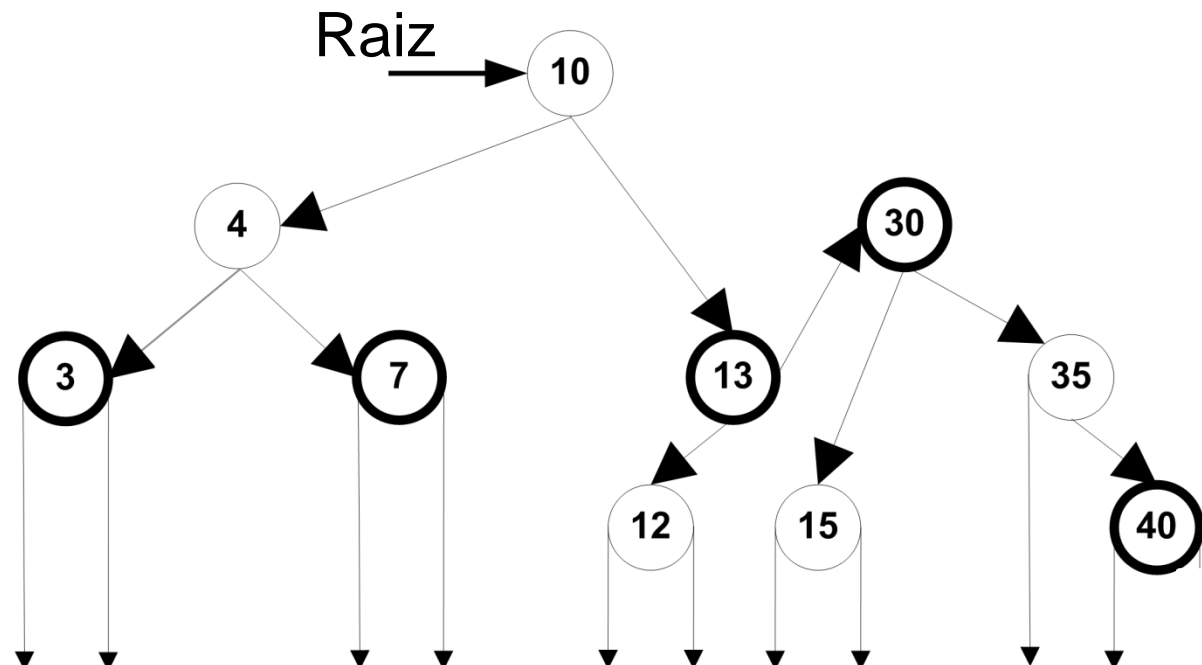
## Exemplo de Inserção

- Por isso, dado o alinhamento dos nós 10, 30 e 13, rotacionamos o 30 para a direita e o 10 para a esquerda



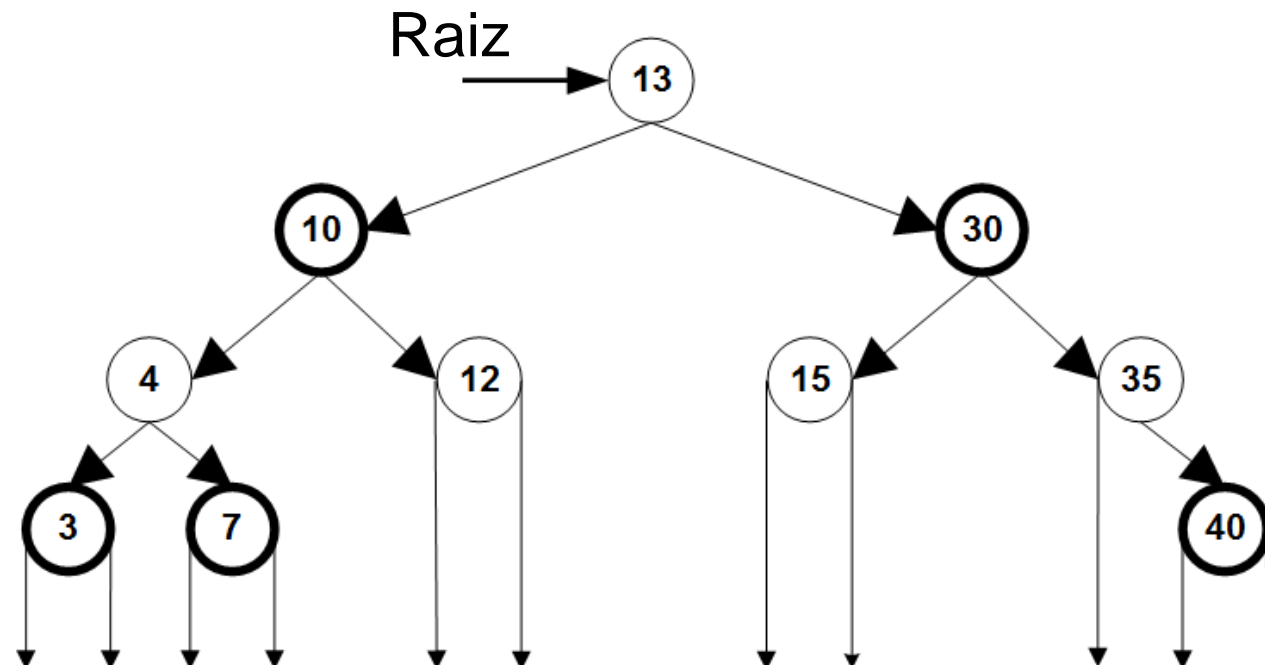
## Exemplo de Inserção

- Por isso, dado o alinhamento dos nós 10, 30 e 13, rotacionamos o 30 para a direita e o 10 para a esquerda



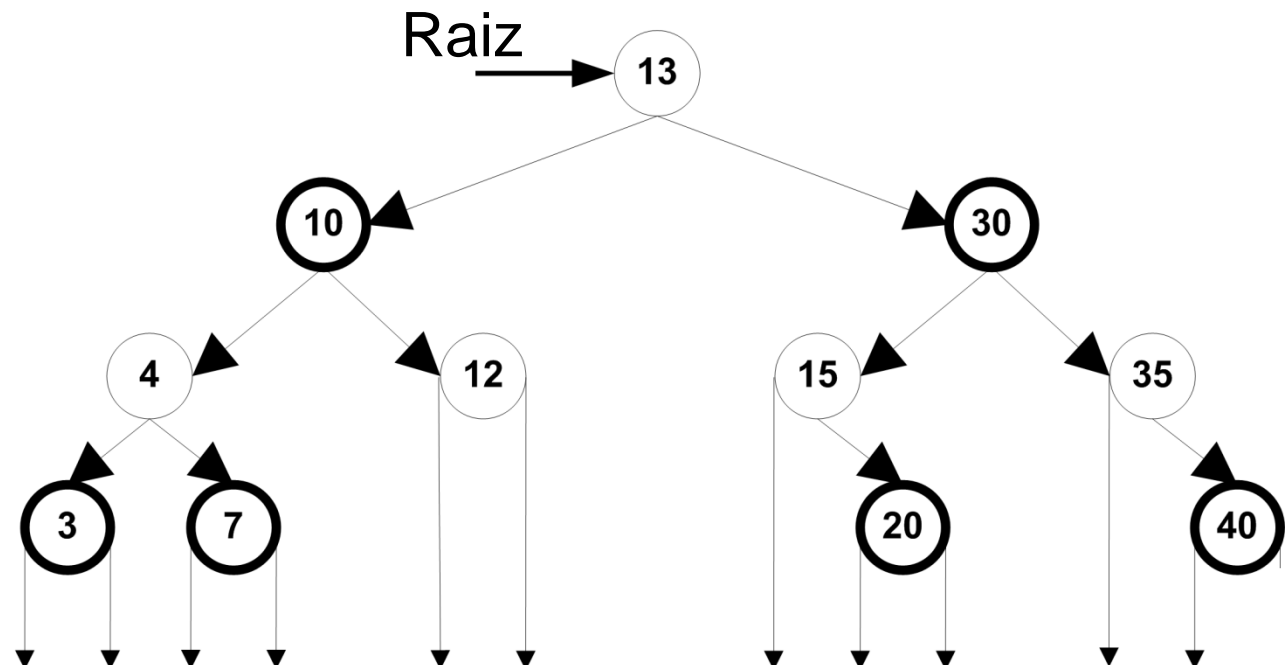
## Exemplo de Inserção

- Por isso, dado o alinhamento dos nós 10, 30 e 13, rotacionamos o 30 para a direita e o 10 para a esquerda



## Exemplo de Inserção

- Finalmente, inserimos o 20



# Exercício

- Crie uma árvore alvinegra através de inserções sucessivas dos números 1 a 20, respectivamente
- Crie uma árvore alvinegra através de inserções sucessivas dos números 20 a 1, respectivamente
- Para cada um dos três exercícios anteriores, verifique sua resposta usando nosso código para a árvore alvinegra

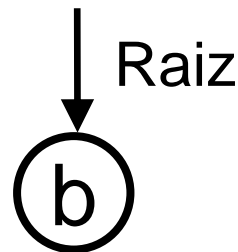
## Algoritmo em C-like

- Como testamos os valores do pai e avô de um nó, consideramos que nossa árvore tem pelo menos três elementos
- Se tivermos zero elementos ( $\text{raiz} == \text{null}$ ), inserimos o novo elemento na raiz



## Algoritmo em C-like

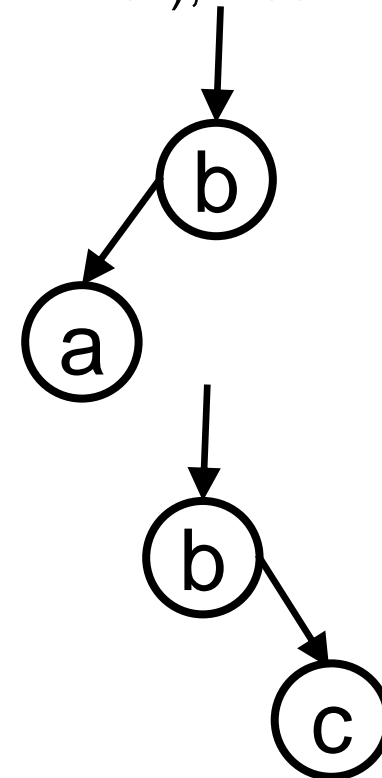
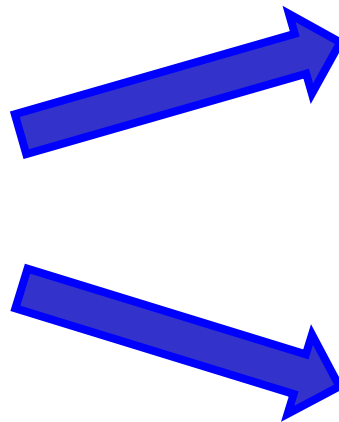
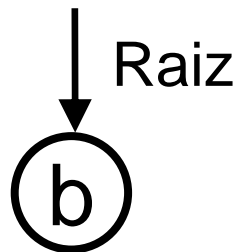
- Como testamos os valores do pai e avô de um nó, consideramos que nossa árvore tem pelo menos três elementos
- Se tivermos um elemento ( $\text{raiz.esq} == \text{null}$  AND  $\text{raiz.dir} == \text{null}$ ), inserimos o novo elemento à esquerda / direita da raiz



Quando o novo elemento  
fica à esquerda / direita?

## Algoritmo em C-like

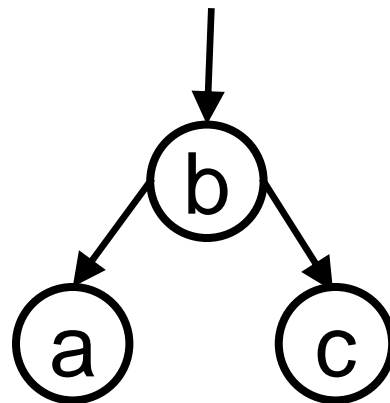
- Como testamos os valores do pai e avô de um nó, consideramos que nossa árvore tem pelo menos três elementos
- Se tivermos um elemento ( $\text{raiz.esq} == \text{null}$  AND  $\text{raiz.dir} == \text{null}$ ), inserimos o novo elemento à esquerda / direita da raiz





## Algoritmo em C-like

- Como testamos os valores do pai e avô de um nó, consideramos que nossa árvore tem pelo menos três elementos
- Se tivermos dois elementos ( $\text{raiz.esq} == \text{null}$  OR  $\text{raiz.dir} == \text{null}$ ), inserimos o novo elemento garantindo o alinhamento ( $a < b$  and  $b < c$ ). Para isso, eventualmente, trocamos os valores existentes de nó



# Algoritmo em C-*like*

Ver código em: [fonte/unidade06/alvinegra/](#)