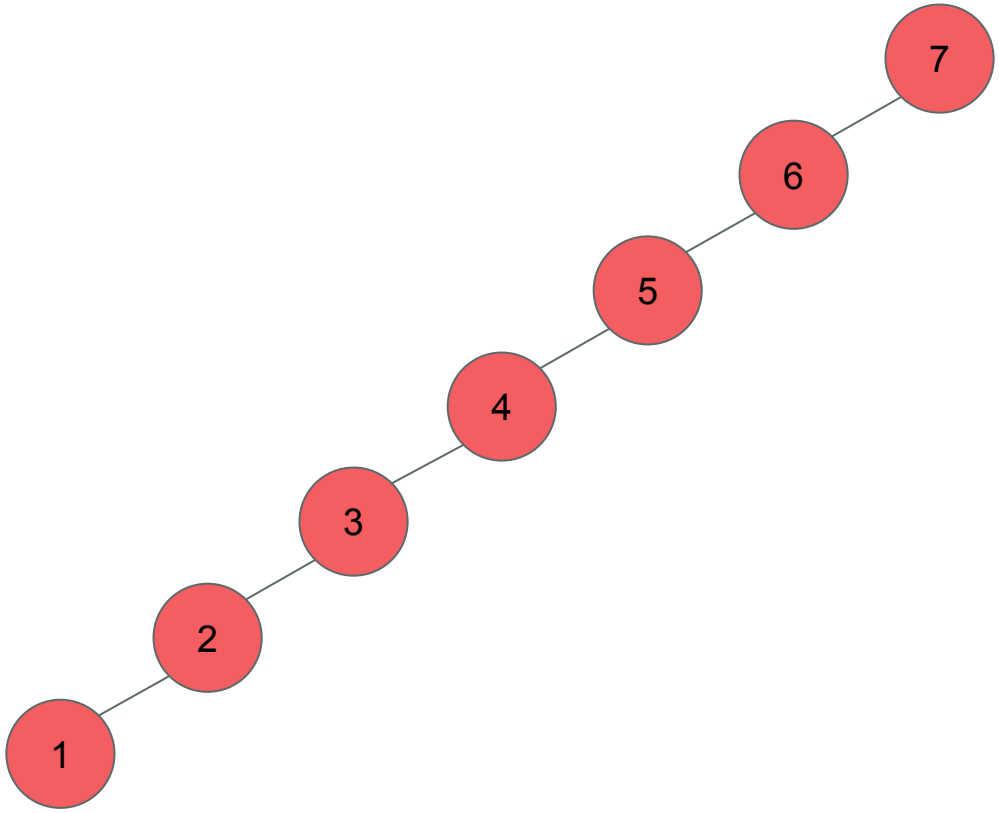
Aula 17 - Árvores Binárias de Busca Balanceadas pt. 2

Estruturas de Dados 2018/1
Prof. Diego Furtado Silva

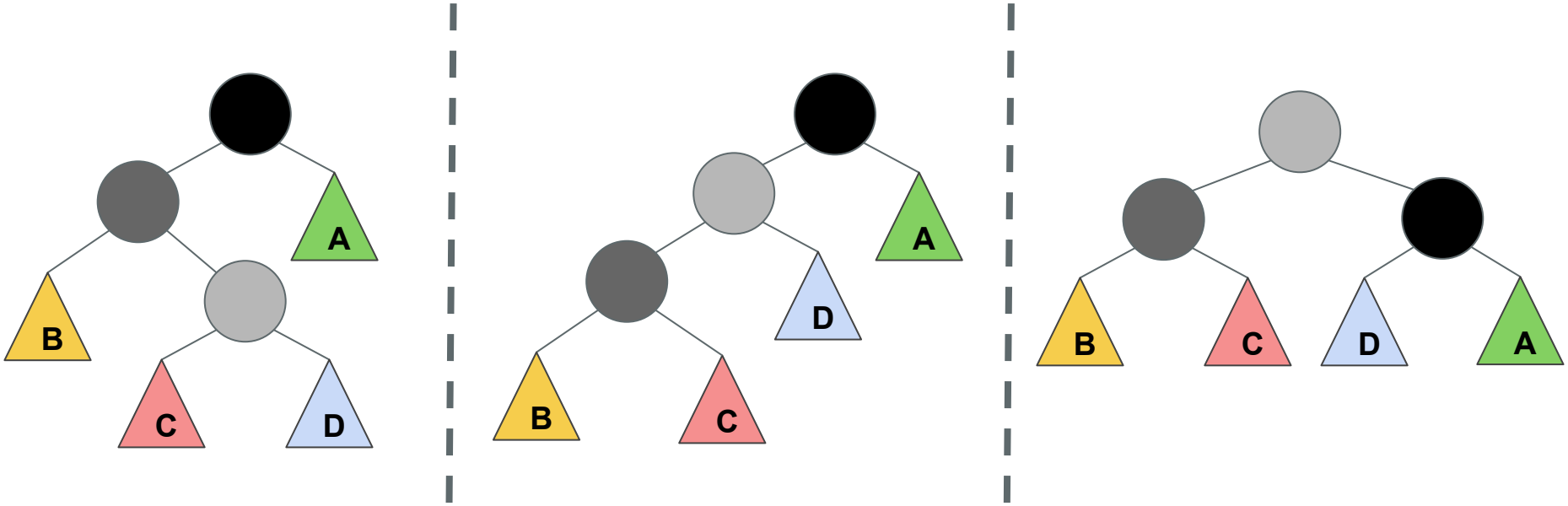
Relembrando - AVL

O objetivo é manter a árvore (aproximadamente) balanceada, evitando que a árvore se degenere em uma lista, como abaixo



Relembrando - AVL

Para isso, são feitas rotações simples ou duplas à direita ou à esquerda (exemplo abaixo: dupla à direita)



Relembrando - AVL

- O objetivo é manter o fator de balanceamento entre -1 e 1
  - $Fb = h_e - h_d$

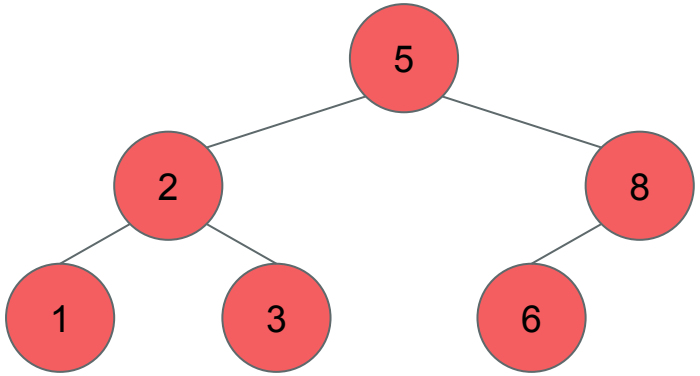
- Já vimos como a inserção interfere no balanceamento
  - Mantemos as alturas dos nós, para poder recalcular o fator de balanceamento após a inserção

- E a remoção?

AVL - Remoção

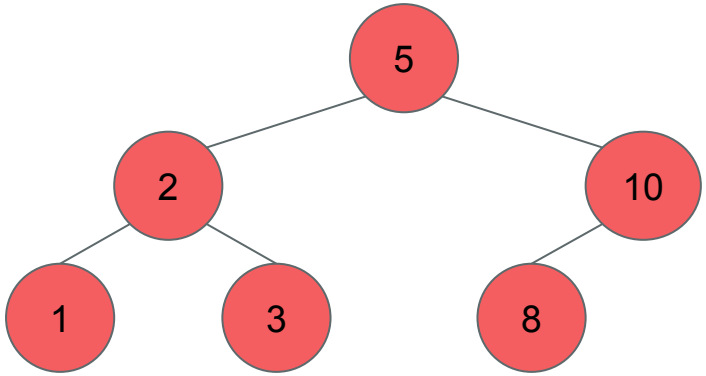
- Primeiro, usamos algoritmo semelhante ao da ABB (sem balanceamento)
- Caímos em três casos
  - Item está em um nó folha (sem filhos)
  - Item está em um nó com apenas um filho
  - Item está em um nó com os dois filhos

AVL - Remoção



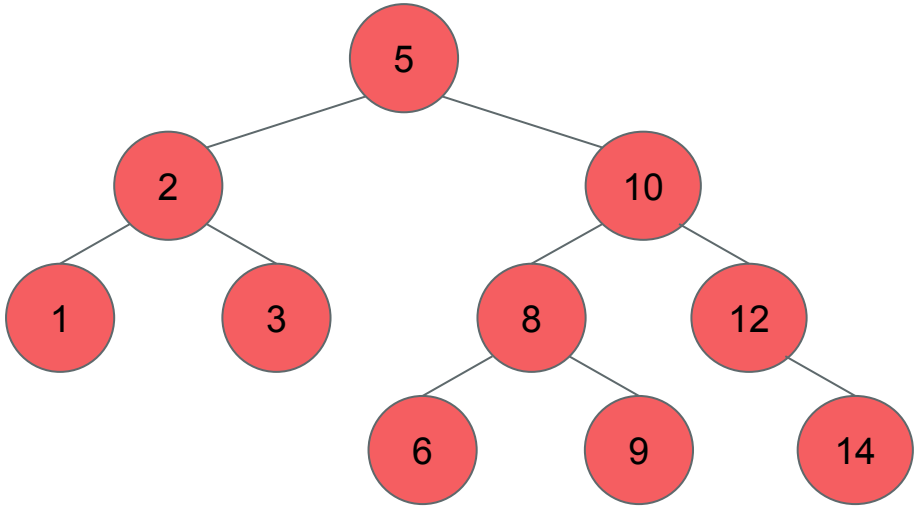
Remover a chave 6 (quadro)

AVL - Remoção



Remover a chave 10 (quadro)

AVL - Remoção



Remover a chave 10

AVL - Remoção

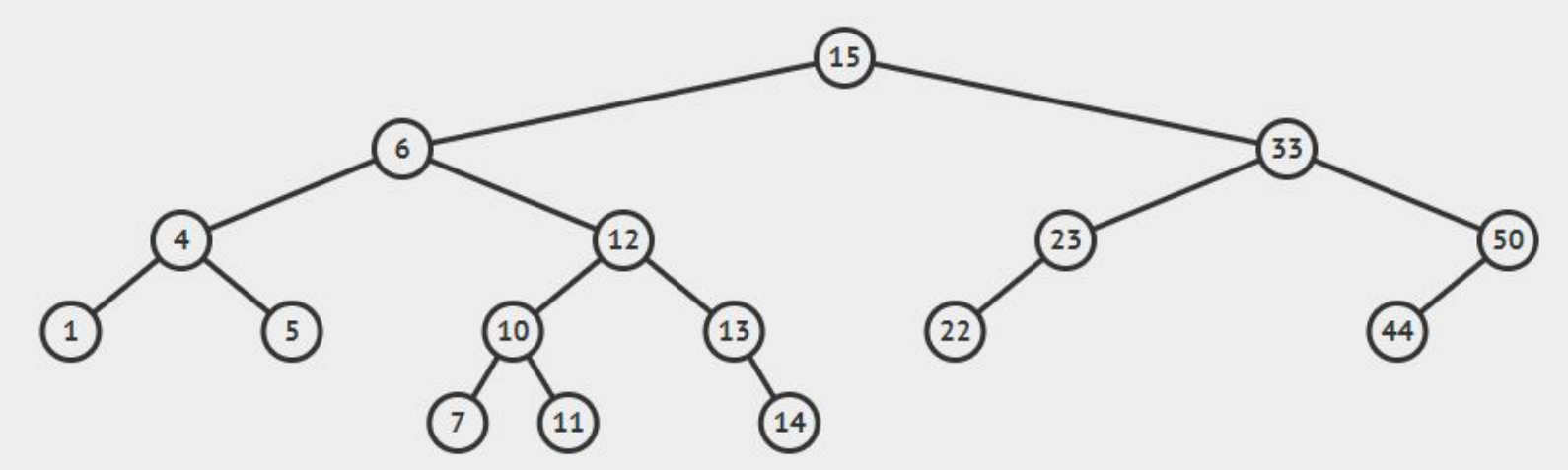
Qualquer remoção **pode** diminuir a altura de uma subárvore

- Portanto, pode causar desbalanceamento
- Devemos fazer o ***retrace*** da folha (que foi removida) até a raiz revendo o rebalanceamento
- É possível que seja necessário fazer várias rotações até a raiz

Exemplos a seguir utilizam o visualgo.net

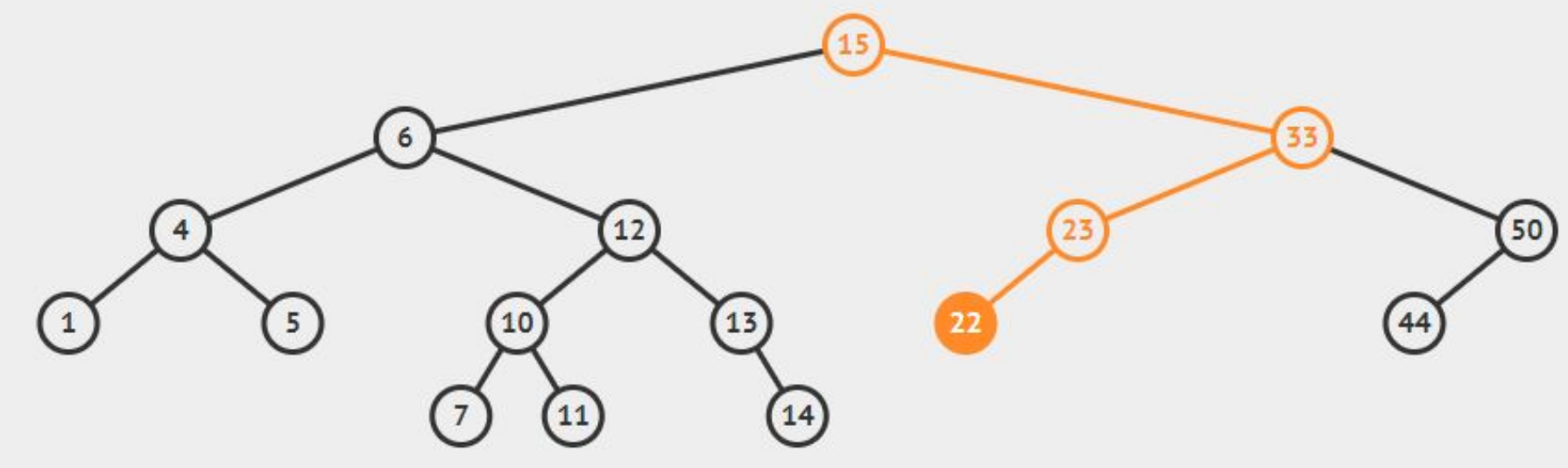
AVL - Remoção

Inserir 15,6,33,4,12,23,50,1,5,10,13,22,44,7,11,14

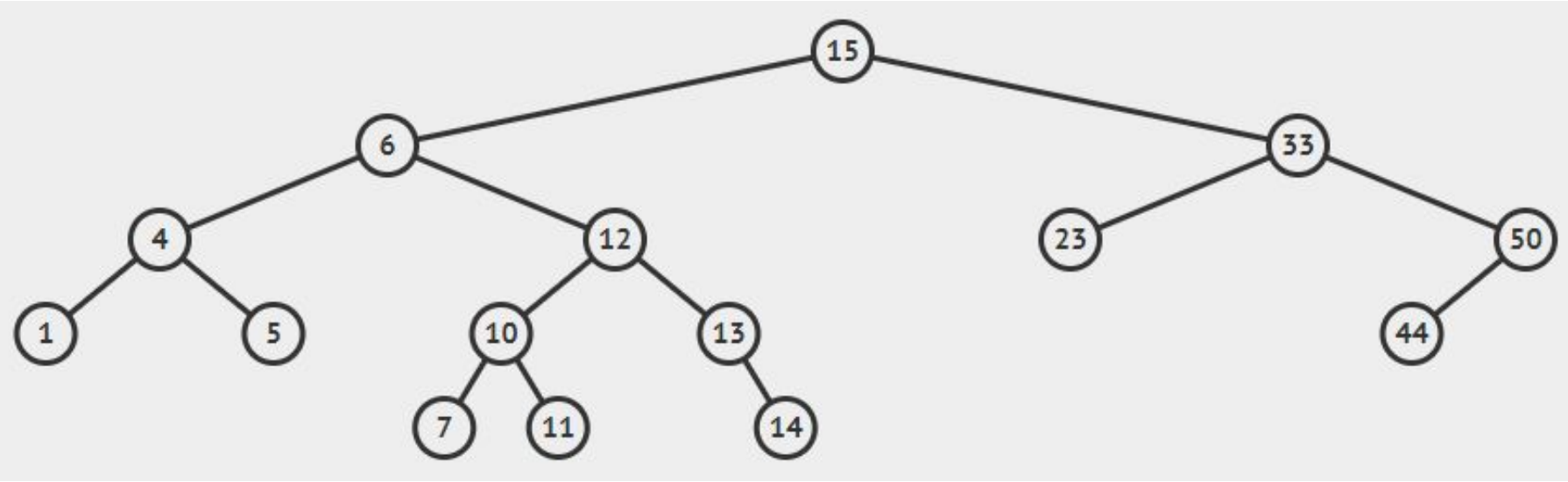


Remover 22 (ainda não causa desbalanceamento) e 23

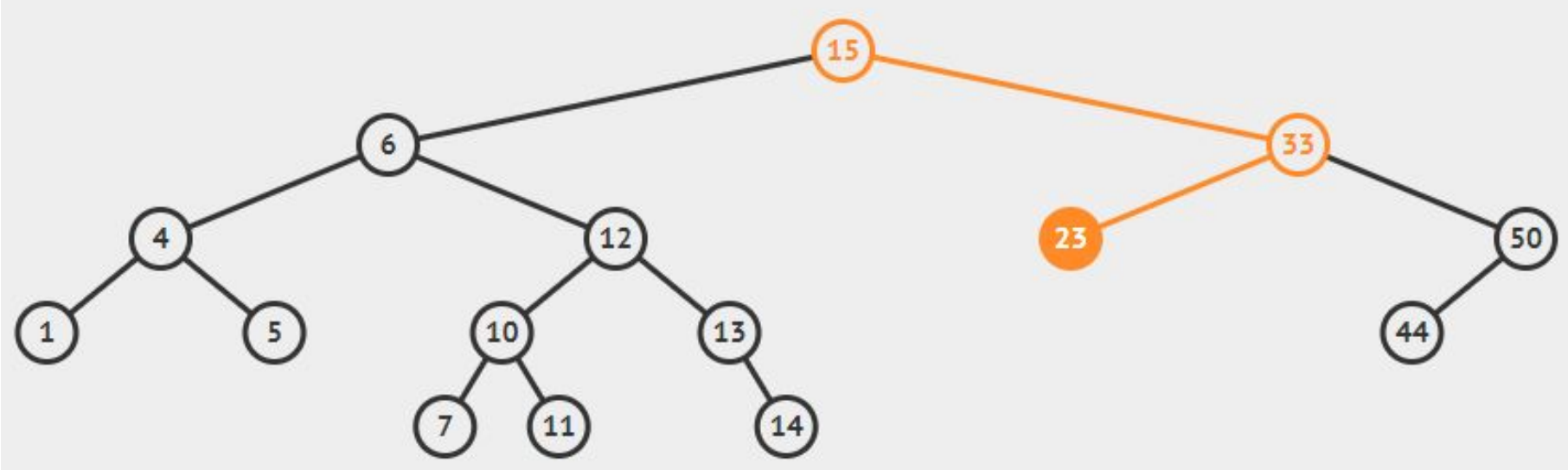
AVL - Remoção



AVL - Remoção

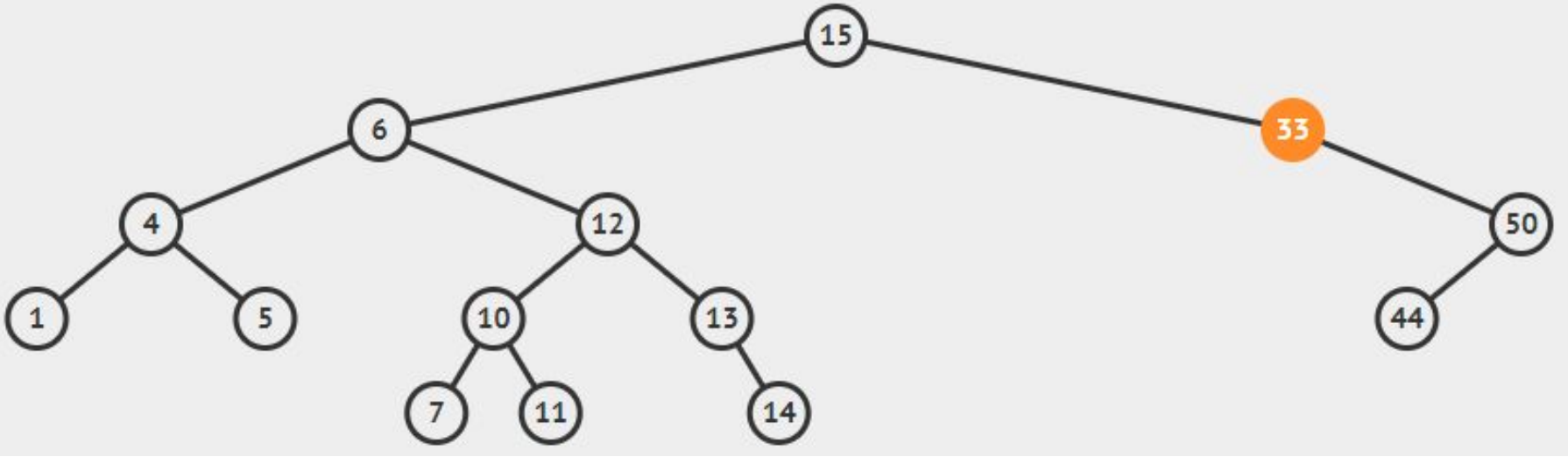


AVL - Remoção



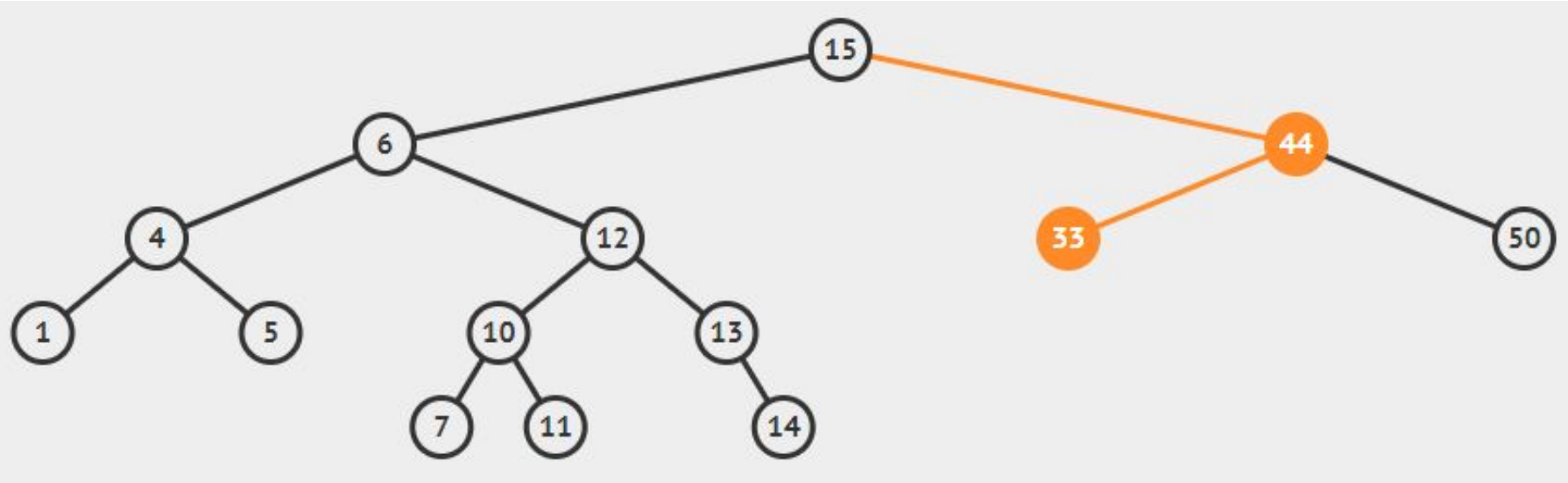
AVL - Remoção

O nó em destaque (e apenas ele) fere |Fb| ≤ 1



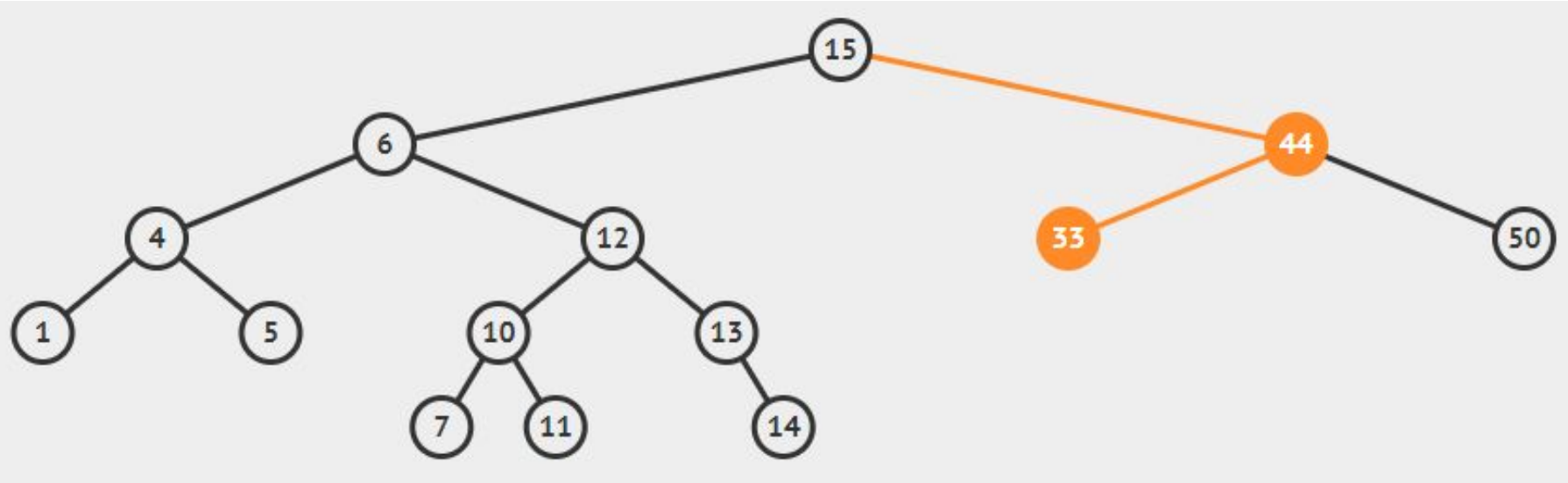
AVL - Remoção

Após a rotação, o nó raiz passa a não ter um Fb desejável



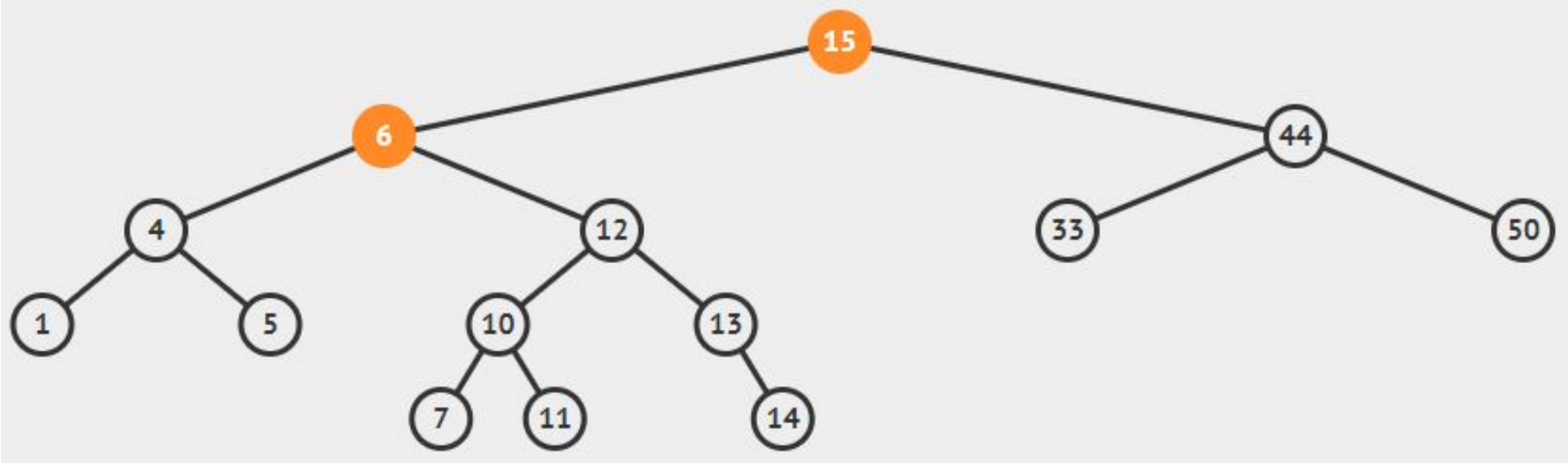
AVL - Remoção

Após a rotação, o nó raiz passa a não ter um Fb desejável

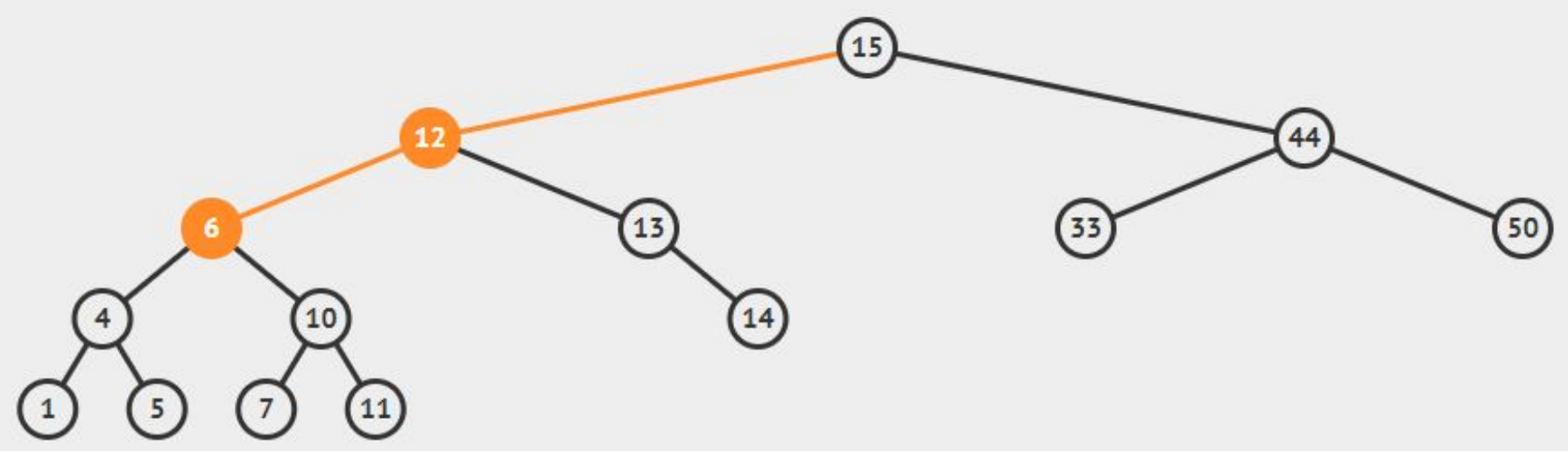


AVL - Remoção

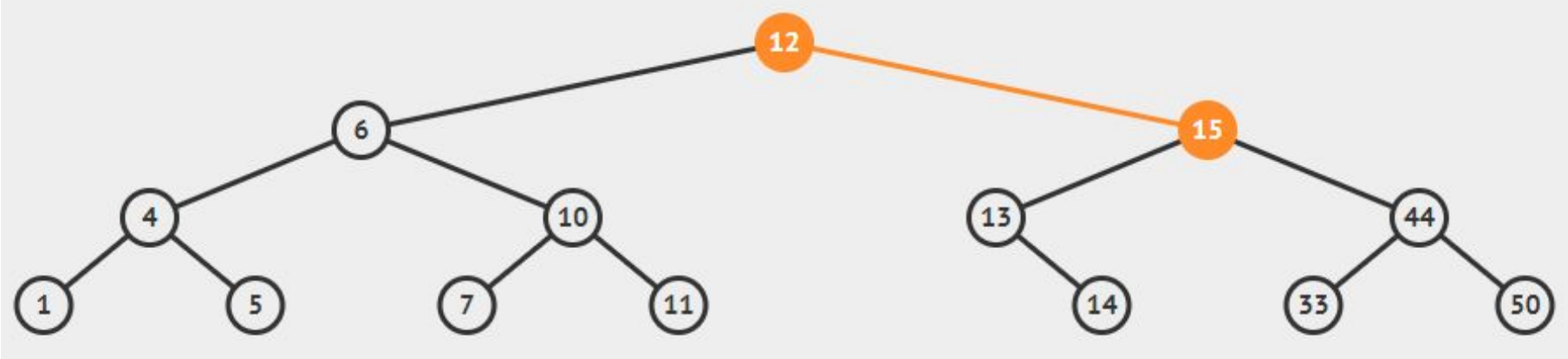
Fazermos uma dupla à direita na raiz



AVL - Remoção

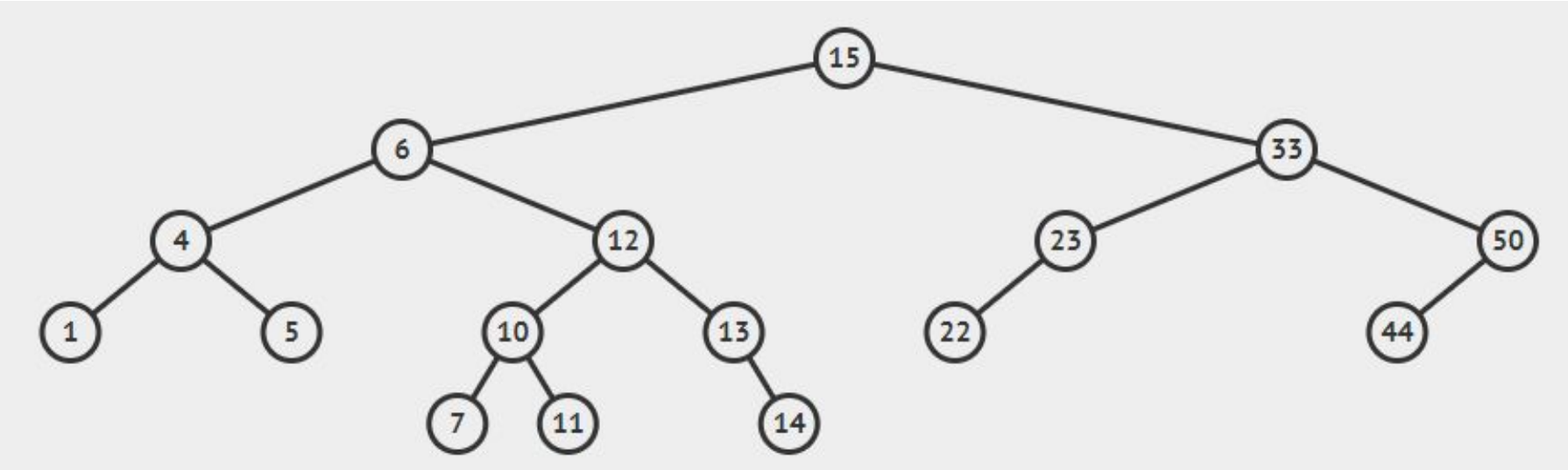


AVL - Remoção



AVL - Remoção

Exercício: a partir dessa AVL, remover 12,1,4,5,44,50



AVL - implementação

E a implementação?

Vale 1 ponto extra na prova!

- Vocês têm que implementar alguma coisa das aulas

- Eu tenho que implementar menos coisas das aulas (trabalho de vocês)

- Sem valer nada, ninguém faz

- Podem/devem usar meu código de ABB, então tá fácil

Aula 17.5 - Outras árvores

Estruturas de Dados 2018/1
Prof. Diego Furtado Silva

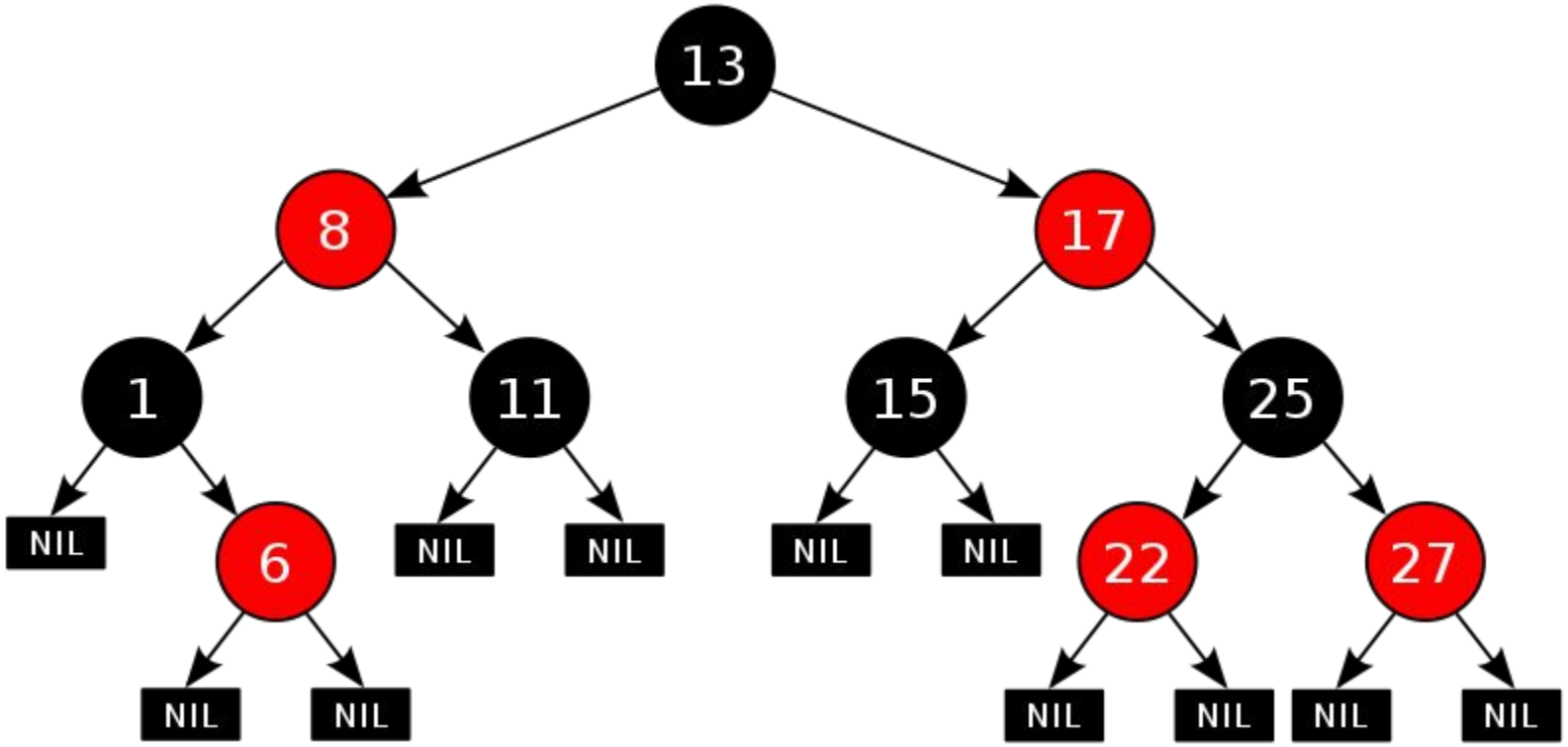
Árvore Rubro-Negra

Assim como a AVL, a rubro-negra (vermelho e preta) também é uma árvore binária de busca balanceada

- Todo nó recebe uma cor (preto ou vermelho)
- A raiz é preta
- Todos os nulos (abaixo das folhas) são pretos
- Ambos filhos de vermelhos são pretos
- Todos os caminhos da raiz (da árvore ou subárvore) até alguma folha possuem o mesmo número de nós pretos

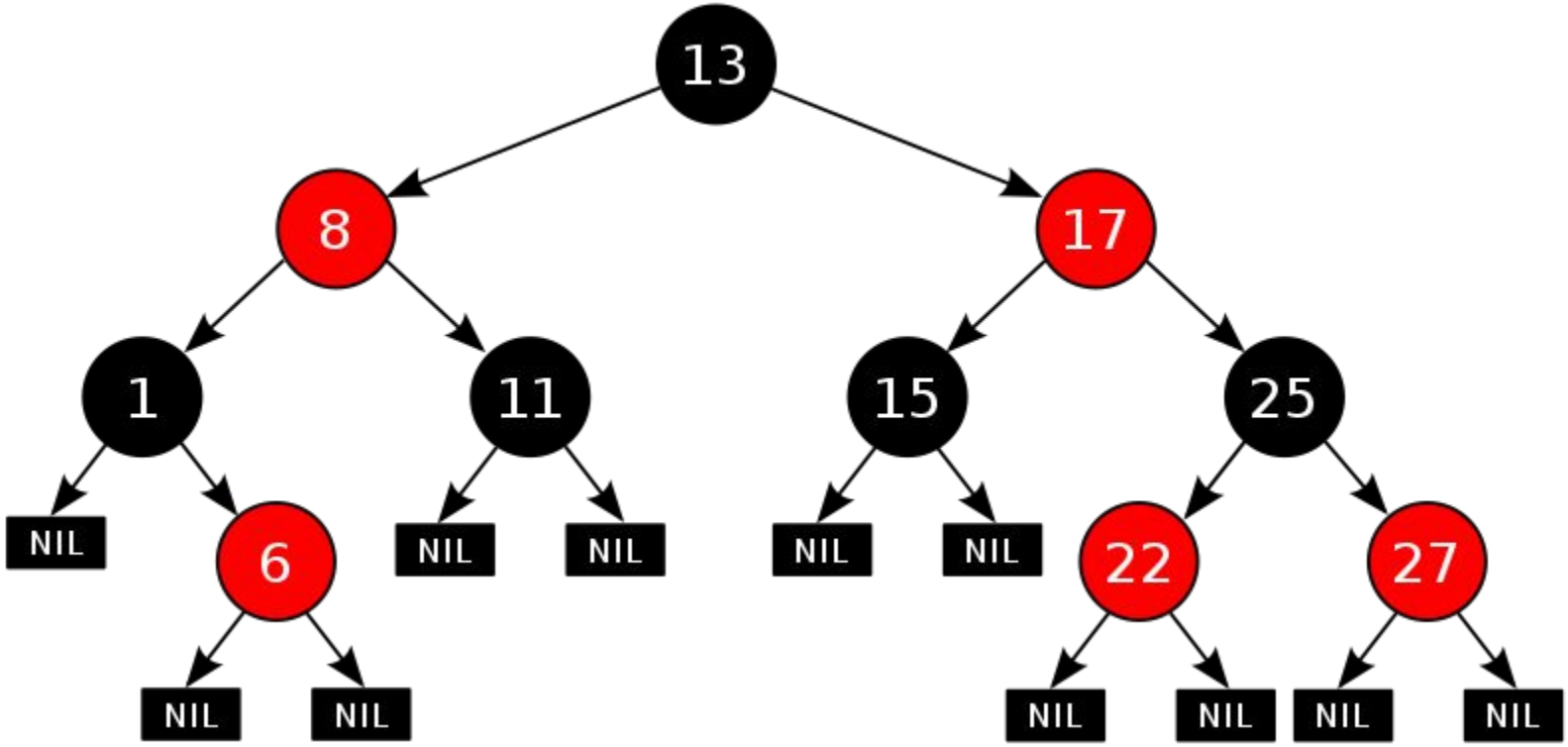
Árvore Rubro-Negra

Isso faz com que o tamanho do caminho mais longo da raiz até alguma folha não seja maior do que o tamanho do mais curto



Árvore Rubro-Negra

Inserções são vermelhas e podem causar rotações e/ou troca de cores de nós para manter as propriedades



ABBs balanceadas

Comparativo entre ABBs e ABBs balanceadas

	Inserção (média / pior caso)	**Remoção (média / pior caso)**	**Busca (média / pior caso)**
ABB	O(log n) / O(n)	O(log n) / O(n)	O(log n) / O(n)
AVL	O(log n) / O(log n)	O(log n) / O(log n)	O(log n) / O(log n)
Rubro-negra	O(log n) / O(log n)	O(log n) / O(log n)	O(log n) / O(log n)

ABBs balanceadas

Na prática, AVL tende a fazer mais rotações quando um item é inserido ou removido, mas provê um melhor balanceamento (e, portanto, buscas mais eficientes).

Alturas das árvores:

AVL: $O(1.44 * \log(N+2))$
Rubro-negra: $O(2 * \lg(N+1))$

Quando usar cada uma delas?

ABBs balanceadas

Além de AVL e rubro-negras, há diversas outras ABBs balanceadas:

- Weaky AVL (WAVL)
- Árvore AA
- Árvore (a,b)
- ...

Há também estruturas randomizadas

- Treap
- Skip-lists
- ...

Outras árvores

Há muitas árvores que são *n*-árias

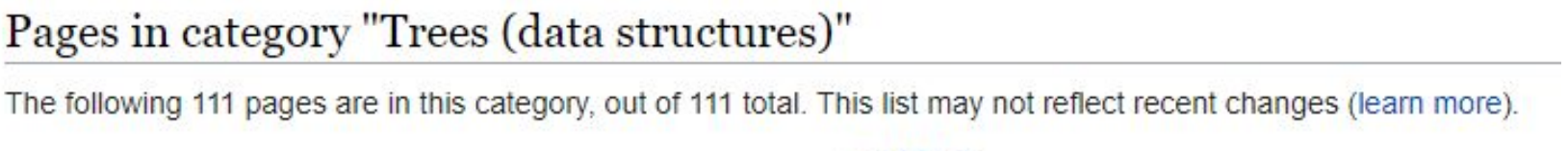
- Podem possuir até *n* filhos (não apenas 2)

Há árvores específicas de domínio

- Árvores para *strings* (trie, árvore de sufixos, etc)
- Árvores para indexar espaço (R, kd, etc)
- Árvores para intervalos
- Etc
- Etc

Outras árvores

Apenas na Wikipedia (em inglês), existem **111** entradas na categoria **Trees (data structures)**


Pages in category "Trees (data structures)"

The following 111 pages are in this category, out of 111 total. This list may not reflect recent changes (learn more).

Por hoje é só

E pelo semestre também (digo… quase isso)