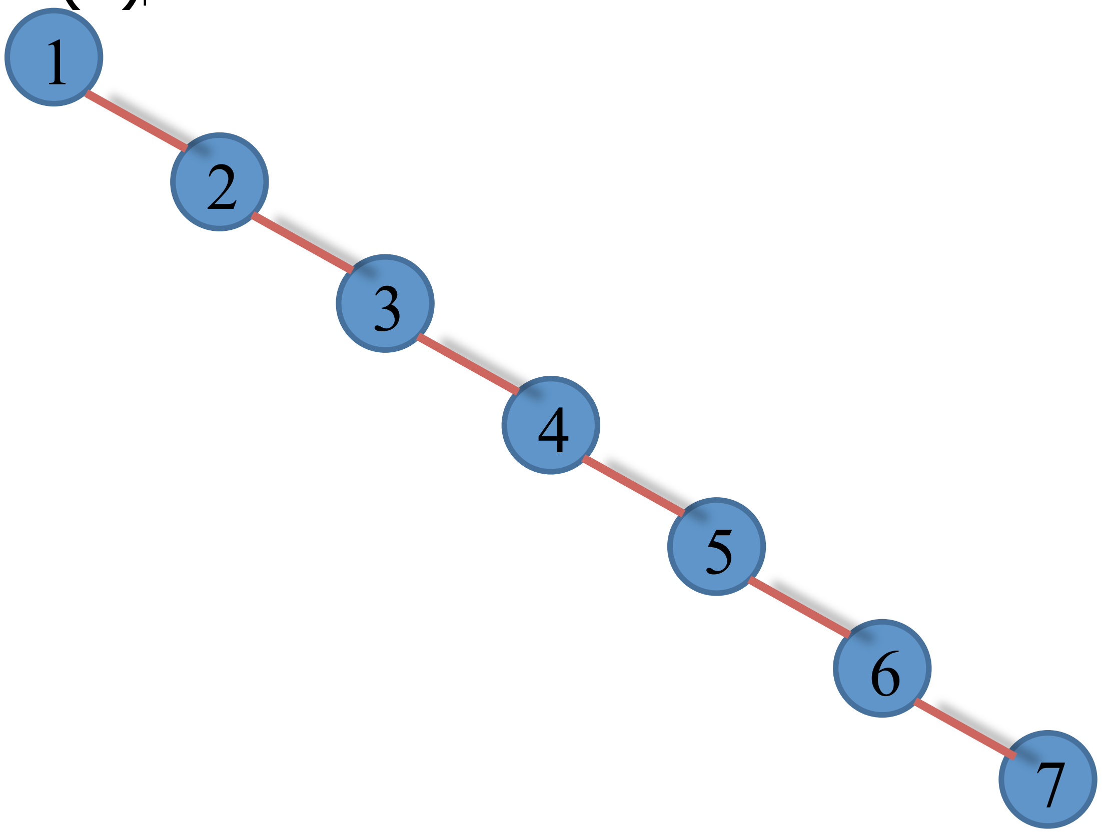
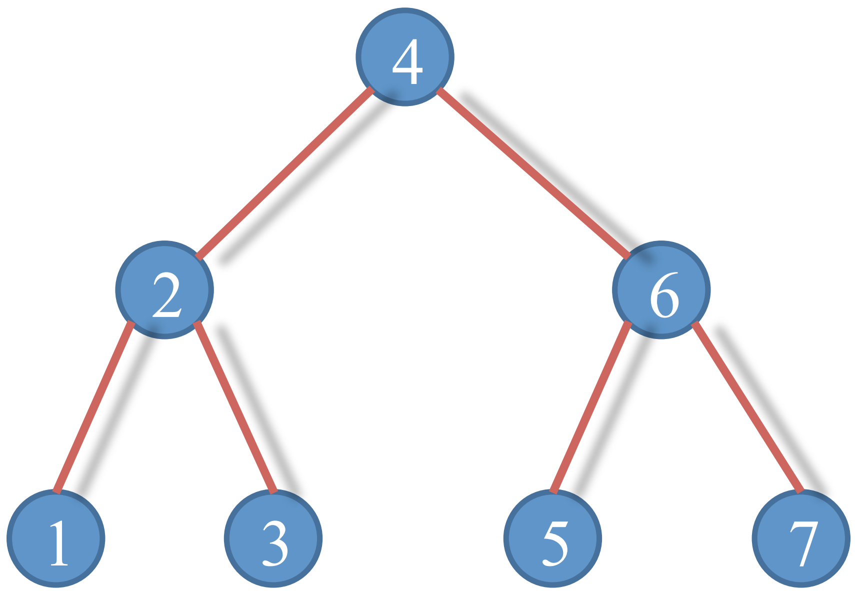
Árvore Binária - altura máxima

- A: Inserção de 1, 2, 3, 4, 5, 6 e 7
- Pior caso: O(n)

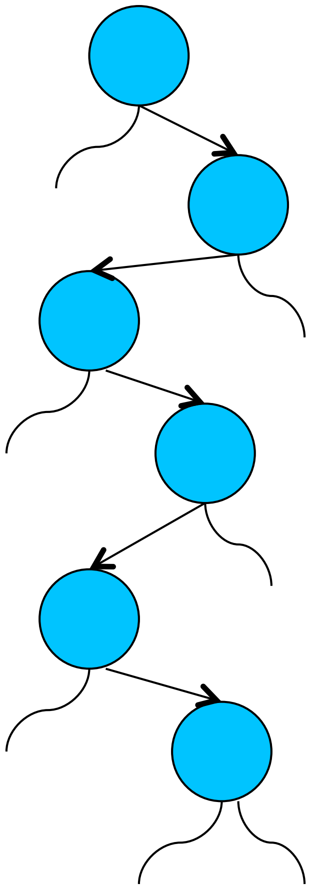


Árvore Binária - altura mínima

- Inserção de 4, 2, 6, 1, 3, 5 e 7, nesta ordem
- Pior caso: O(*log* n)



Árvore Binária de altura máxima



Para árvores com n nós:

- altura máxima: cada nó não folha só possui um filho - ziguezague
- sua altura é n-1

Árvore Binária - altura mínima

- Seja T′ uma árvore binária de altura mínima
  - Se T′ não é completa, retira-se uma folha w de seu último nível e coloca-se como filho de uma folha de nível superior (acima do penúltimo)
  - Repete-se a operação até não ser possível mais realizá-la
  - A árvore resultante T″ é completa
- Se a altura de T″ for menor que a de T′ → esta não seria mínima.
- T″ não pode ser de altura maior → nenhum nó foi retirado

T′ e T″ possuem a mesma altura

Árvore Binária - altura mínima

nível 0 – (somente a raiz) contém um nó

nível 1 – contém no máximo 2 nós

.....

no nível L - pode conter no máximo 2^L nós

árvore binária cheia de altura d tem exatamente 2^L nós em cada nível $0 <= L <= d$

Árvore Binária

O total de nós n em uma árvore binária cheia (que seria o máximo) de altura d é a soma do número de nós a cada nível

$$n = 2^0 + 2^1 + 2^2 + \dots\dots + 2^d = \sum_{j=0}^{d} 2^j$$

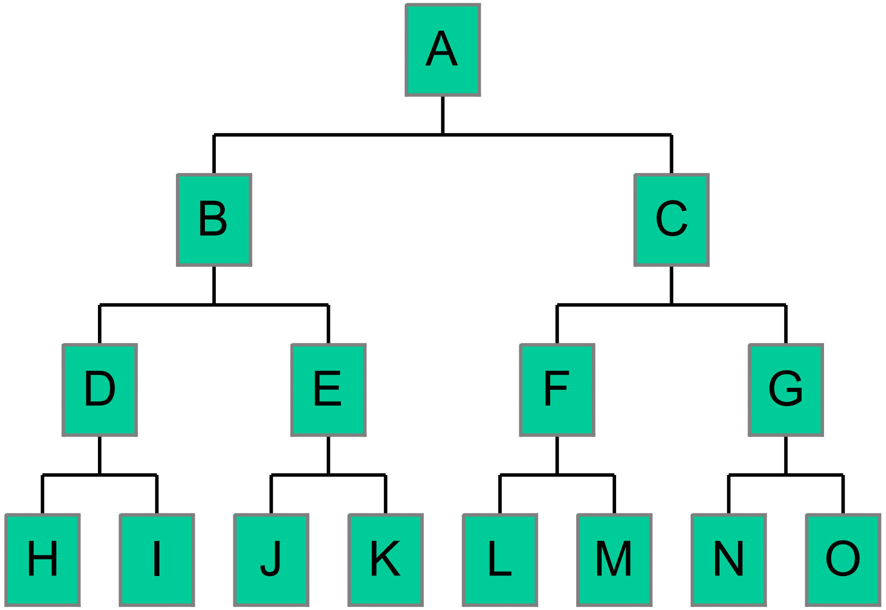
$$n = 2^{d+1} - 1 \rightarrow d = \log(n+1) - 1$$

pode-se mostrar também por indução!

OBS.: número de folhas de uma árvore cheia com n nós

$$2^d = 2^{\log(n+1)-1} = \frac{2^{\log(n+1)}}{2} = \frac{n+1}{2}$$

Árvore Binária Cheia



Árvore Binária Cheia de altura 3

→ $2^{3+1}-1 = 15$ nós

Árvore Binária Cheia

É a árvore binária com o máximo de nós para uma dada altura, mas

a distância da raiz é pequena

Lema: Seja T uma árvore binária completa com $n>0$ nós. Então, T possui altura h mínimo. Além disso, $h = O(\log n)$

Árvores AVL

- importante manter balanceadas árvores binária de busca

- Árvore Balanceada ou **AVL**:
  - a altura de uma árvore binária é o maior nível de suas folhas
  - A altura de uma árvore vazia é definida como -1

Árvores AVL - definição

- *árvore AVL* é uma árvore binária de busca onde:
  - para cada vértice, a altura das duas sub-árvores não difere por mais de um:

$$\boxed{|altura\ (subdir\ (v)) - altura\ (subesq(v))| <= 1}$$

Árvores AVL - definição

- toda árvore completa é AVL e toda AVL é completa?

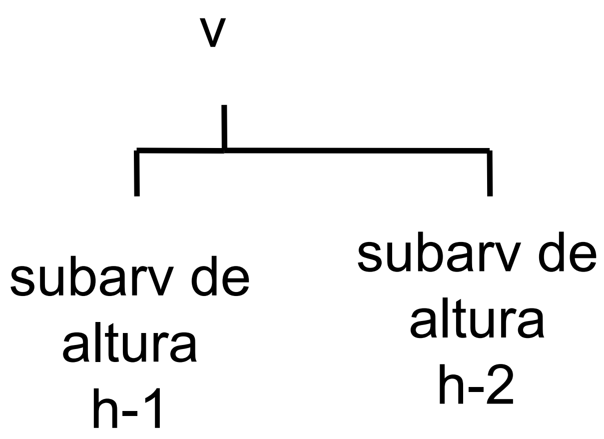
- se a inequação acima acontece para um vértice v, v é dito *regulado*, senão, *desregulado*

Árvore AVL de altura h

- *Qual a altura da árvore AVL?*
  - tem altura mínima?

Árvore AVL de altura h

- Vamos trabalhar no número mínimo de nós
- sem perda de generalidade, por definição:



Árvore AVL de altura h

- Suponha *altura(subesq(v)) = h-1*

Qual o número mínimo de nós?

- Como queremos analizar o número mínimo de nós, então *altura(subd(v)) = h-2*
- árvore AVL construída recursivamente:

Seja T_h uma árvore AVL de altura h

h = 0 T_h tem 0 nós: $| T_h | = 0$

h = 1 $| T_h | = 1$

h = 2 $| T_h | = 1 + | T_{h-1} | + | T_{h-2} |$

Qual o número mínimo de nós?

h	$\mid T_h \mid$
0	0
1	1
2	2
3	4
4	7
5	12
6	20
7	33

Qual o número mínimo de nós?

- Por observação:
  - o *h*-ésimo elemento da sequência de Fibonacci é:

$$F_h = 0 \qquad \text{se } h = 0$$
$$F_h = 1 \qquad \text{se } h = 1$$
$$F_h = F_{h-1} + F_{h-2} \qquad \text{se } h > 1$$

análogo a $|T_h|$, mas diferindo de 1: $|T_h| = F_h + 1$

  - pode-se provar que (exercício 5.3)

termo **< 1** se **h>0**

$$F_h = 1/\sqrt{5}\,[(\,(1+\sqrt{5})^h/2) - ((1-\sqrt{5})^h/2\,)]$$

$$|T_h| = F_h + 1$$

Qual o número mínimo de nós?

$|T_h| > 1/\sqrt{5}\ [((1+\sqrt{5})^h/2) - ((1-\sqrt{5})^h/2)] - 1$

fazendo $a = ((1+\sqrt{5})/2)$ temos

$|T_h| > 1/\sqrt{5}\ a^h - 1$

$|T_h| + 1 > 1/\sqrt{5}\ a^h$

$\log_a (|T_h| + 1) > \log_a (a^h/\sqrt{5})$

$\log_a (|T_h| + 1) > h \log_a \sqrt{5}$ (mudando para base 2)

$\log_2 (|T_h| + 1)/\log_2 a > h \sqrt{5}$

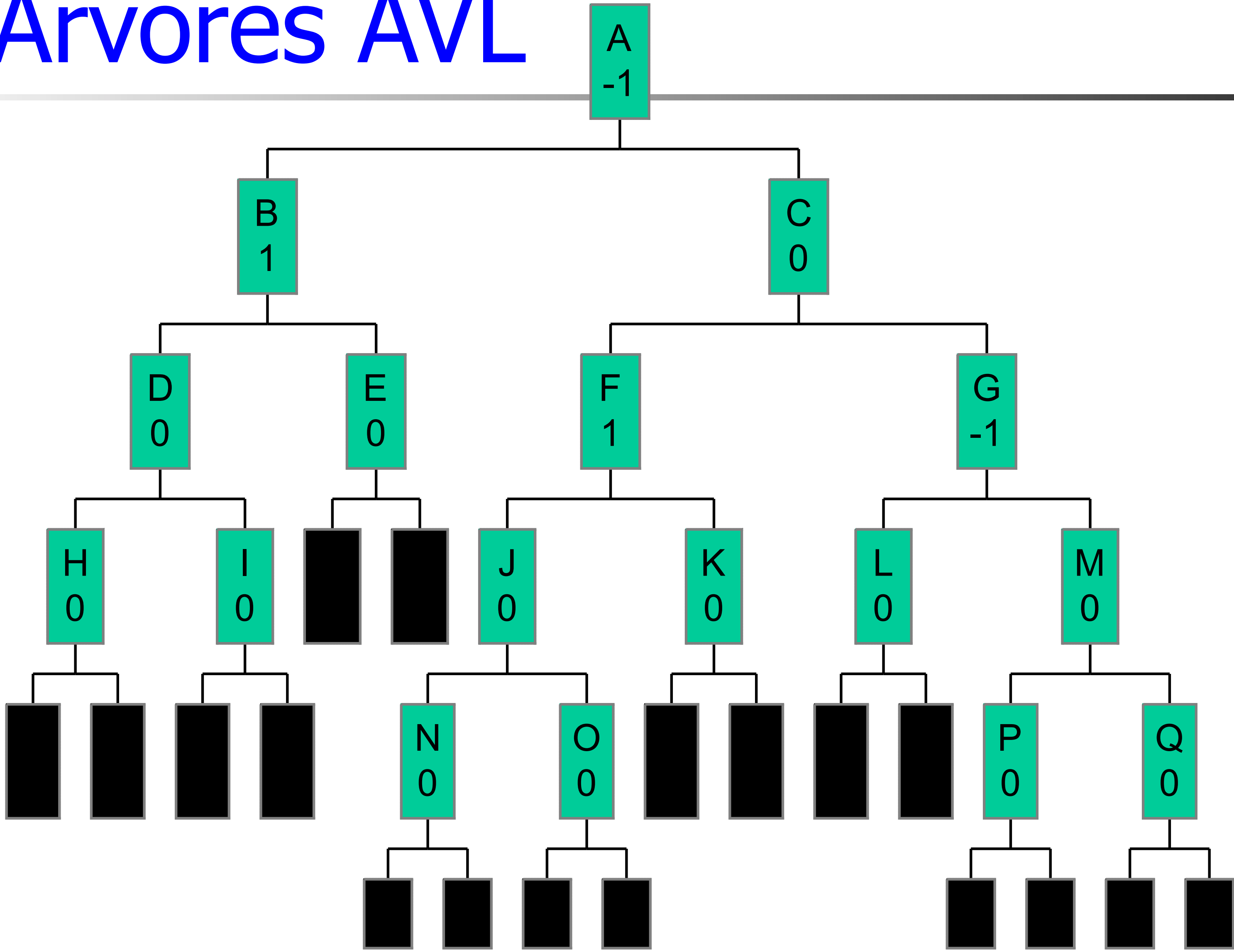
→ $h = O(\log n)$

Árvores AVL

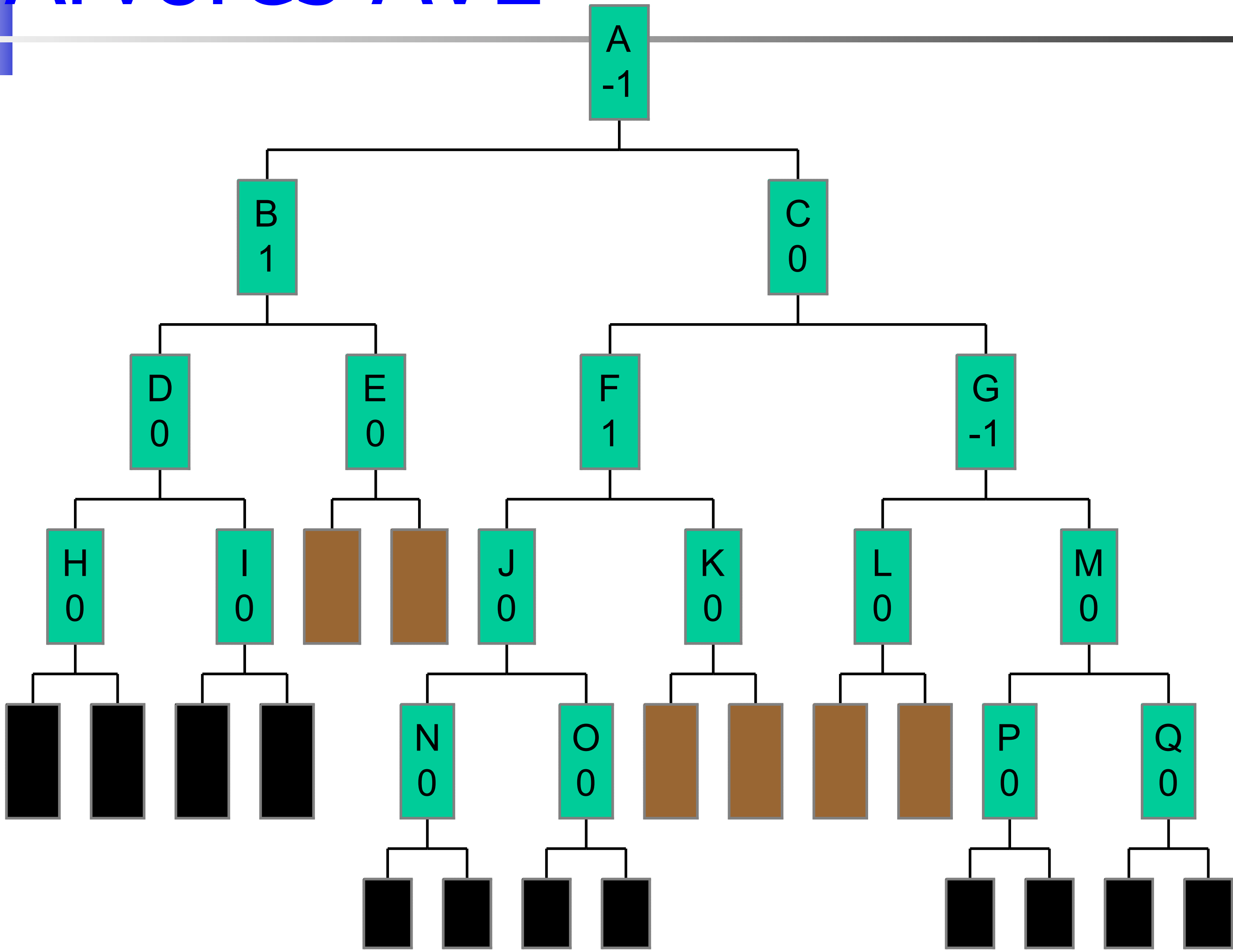
- *balanço* de um nó
  - altura da sub-árvore da esquerda menos a altura da sub-árvore direita
- O balanço de cada nó em uma árvore binária AVL balanceada
  - -1, 0 ou 1
- Seja T uma árvore binária balanceada. Ao inserir um nó *q* em T
  - a árvore resultante pode ou não ser balanceada
- toda vez que um nó qualquer de T se torna *desregulado*

  rebalancear, através de rotações
- cuidado: manter a ordenação entre as chaves

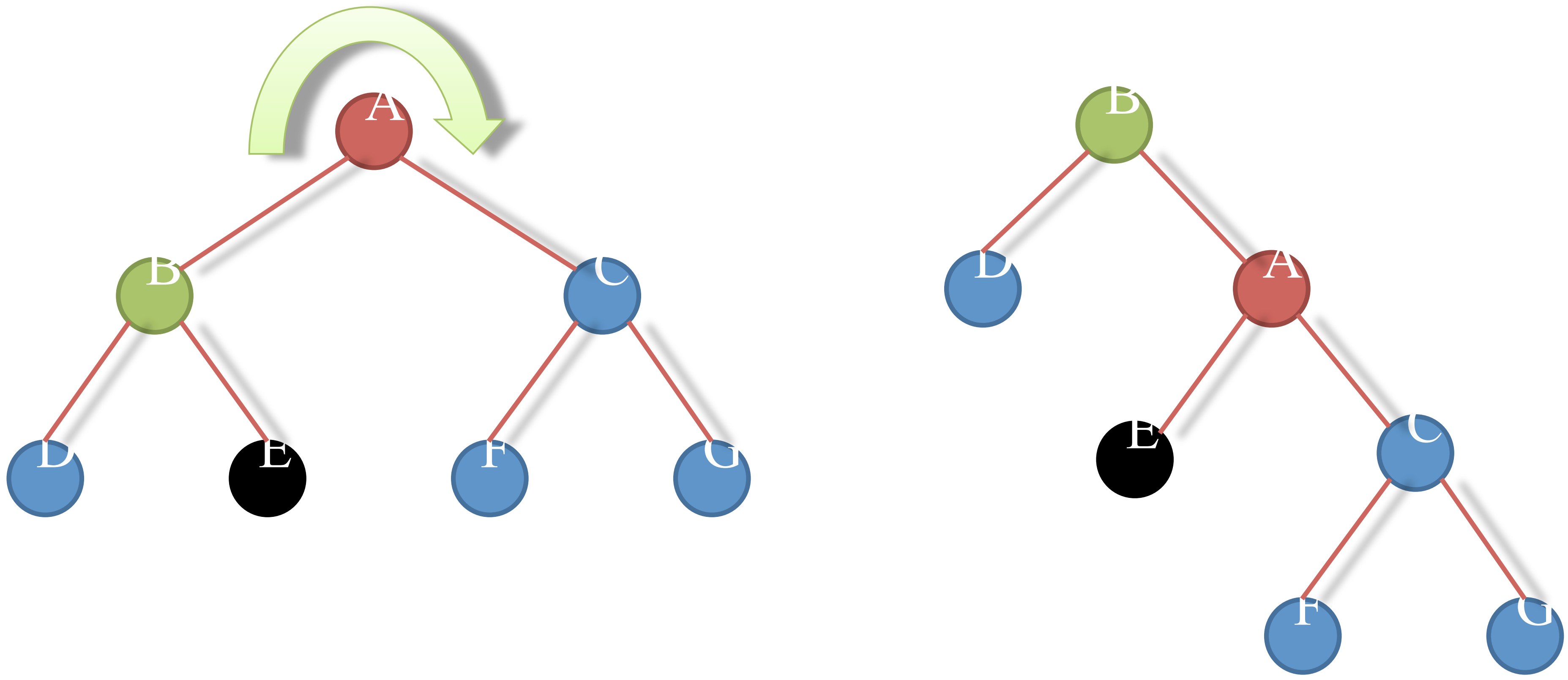
Árvores AVL



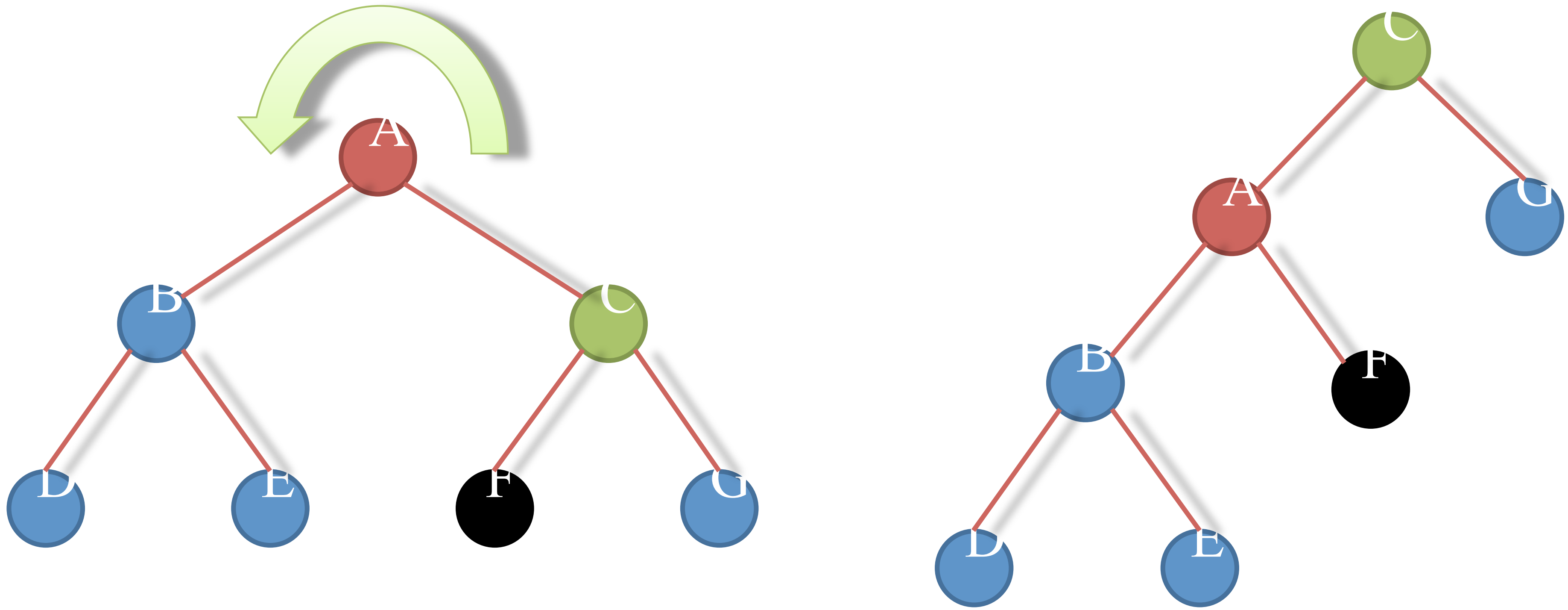
Árvores AVL



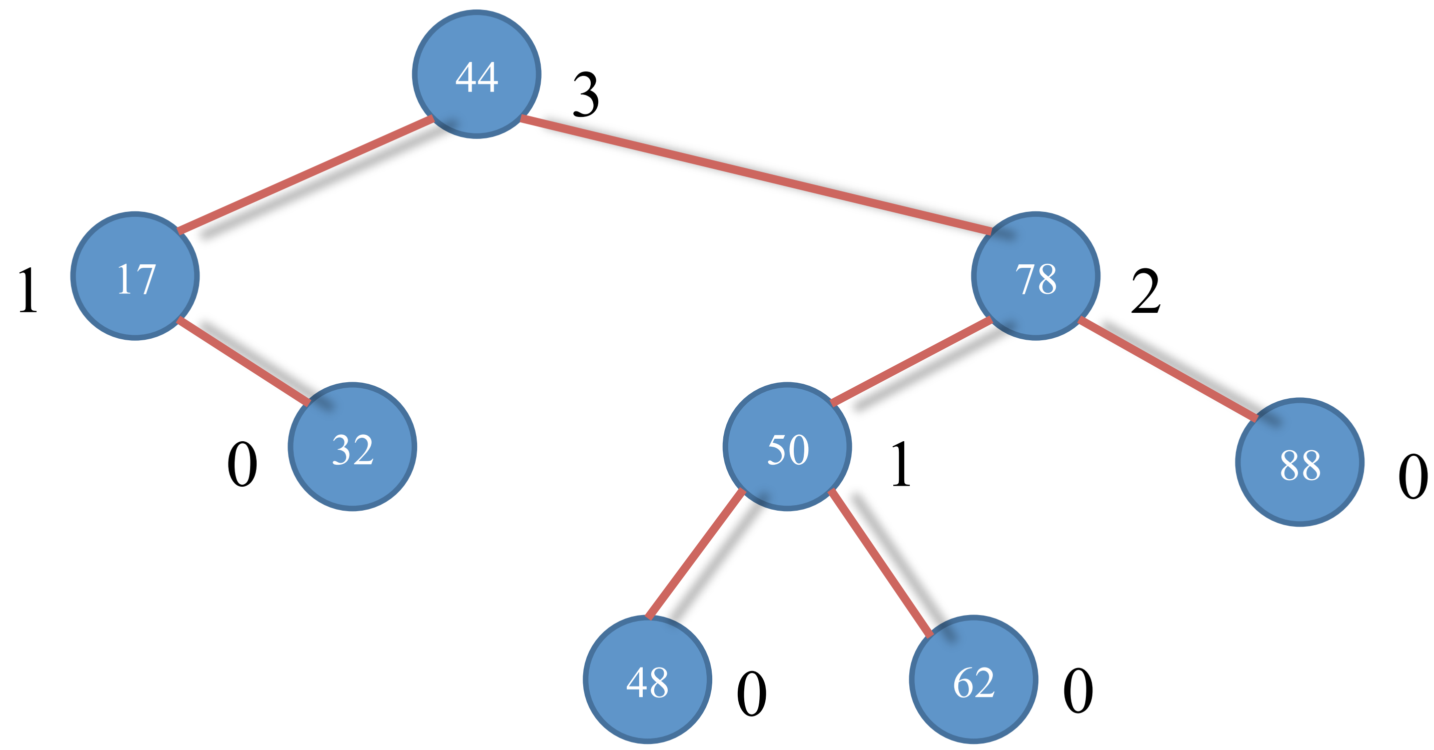
Árvores AVL – rotações



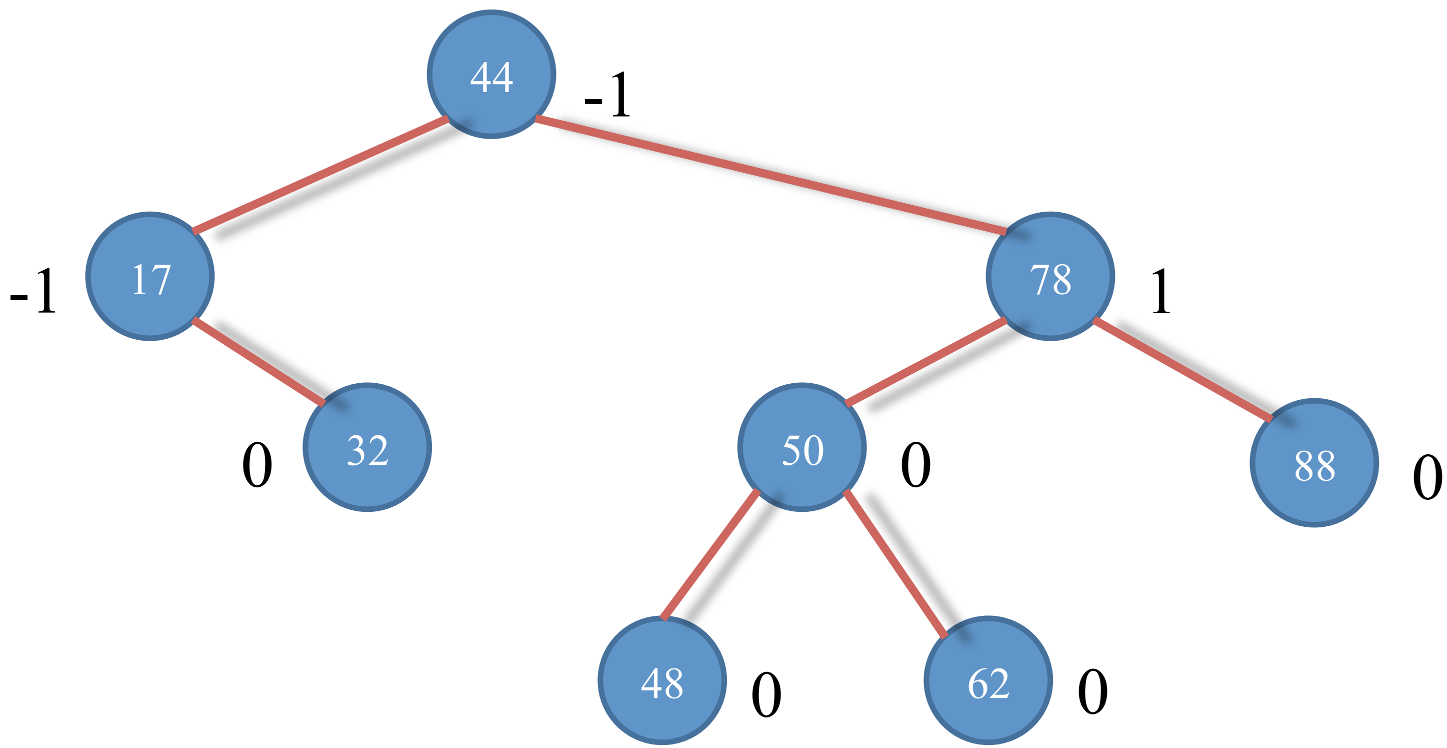
Árvores AVL – rotações



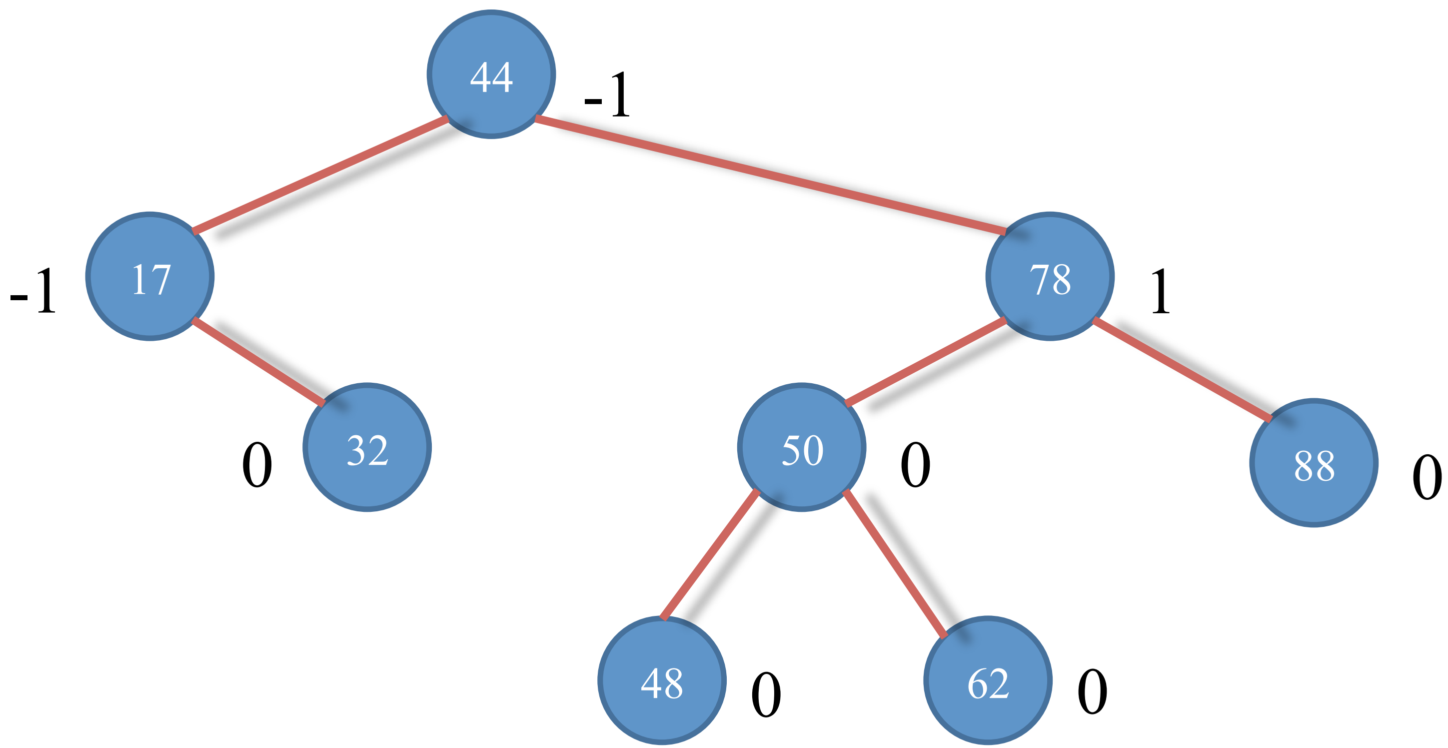
Árvores AVL - exemplos



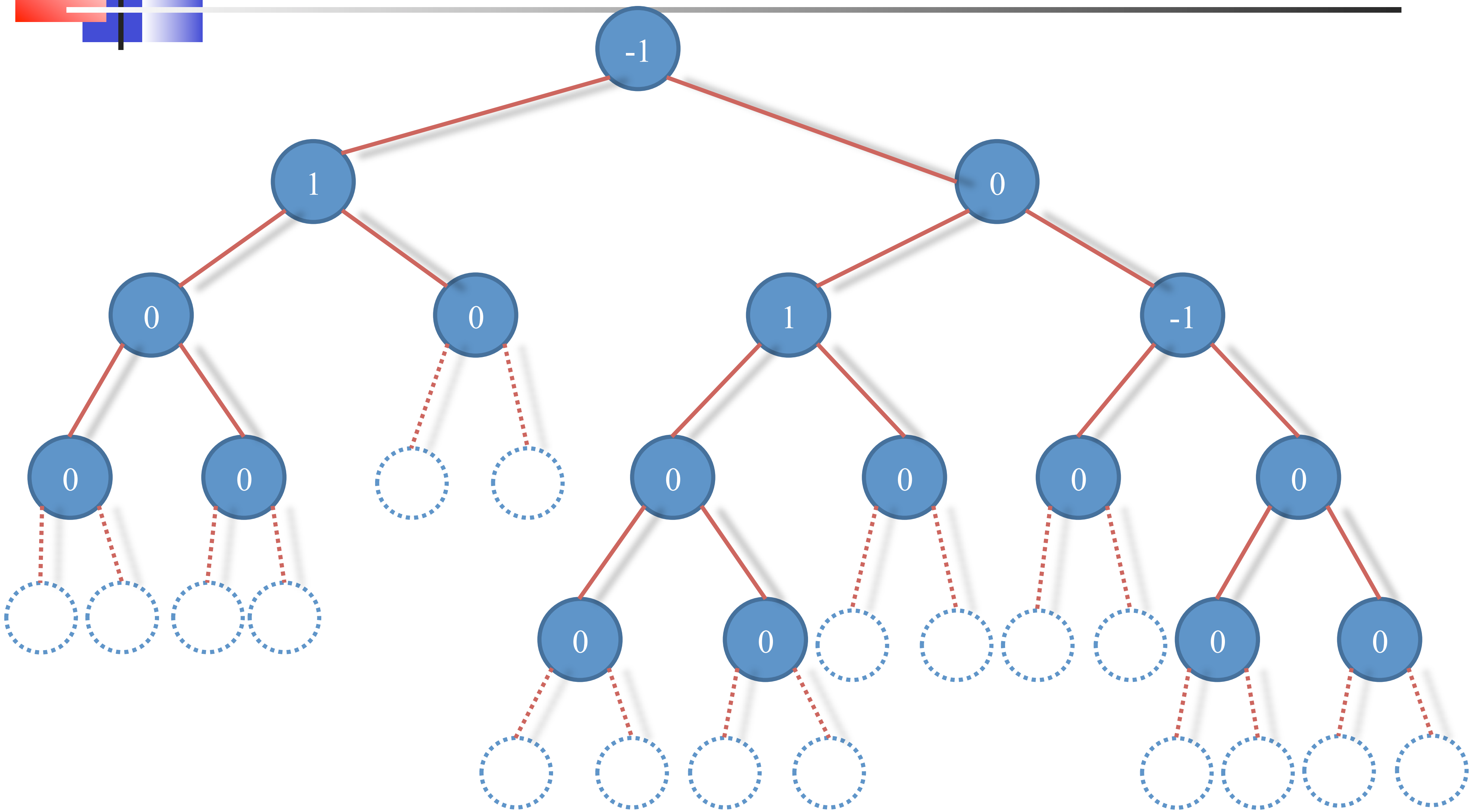
Árvores AVL – fator de balanceamento



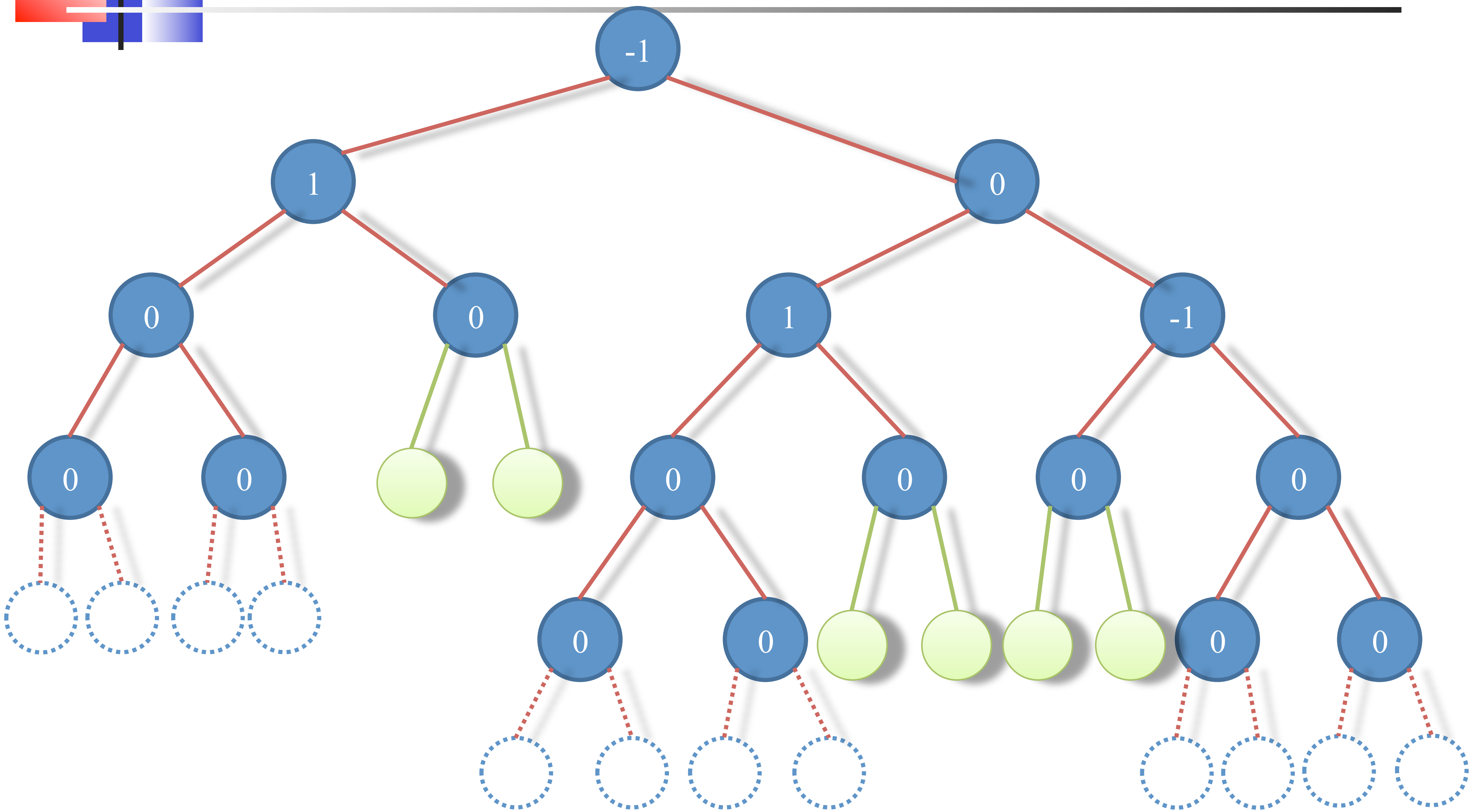
Árvores AVL – fator de balanceamento



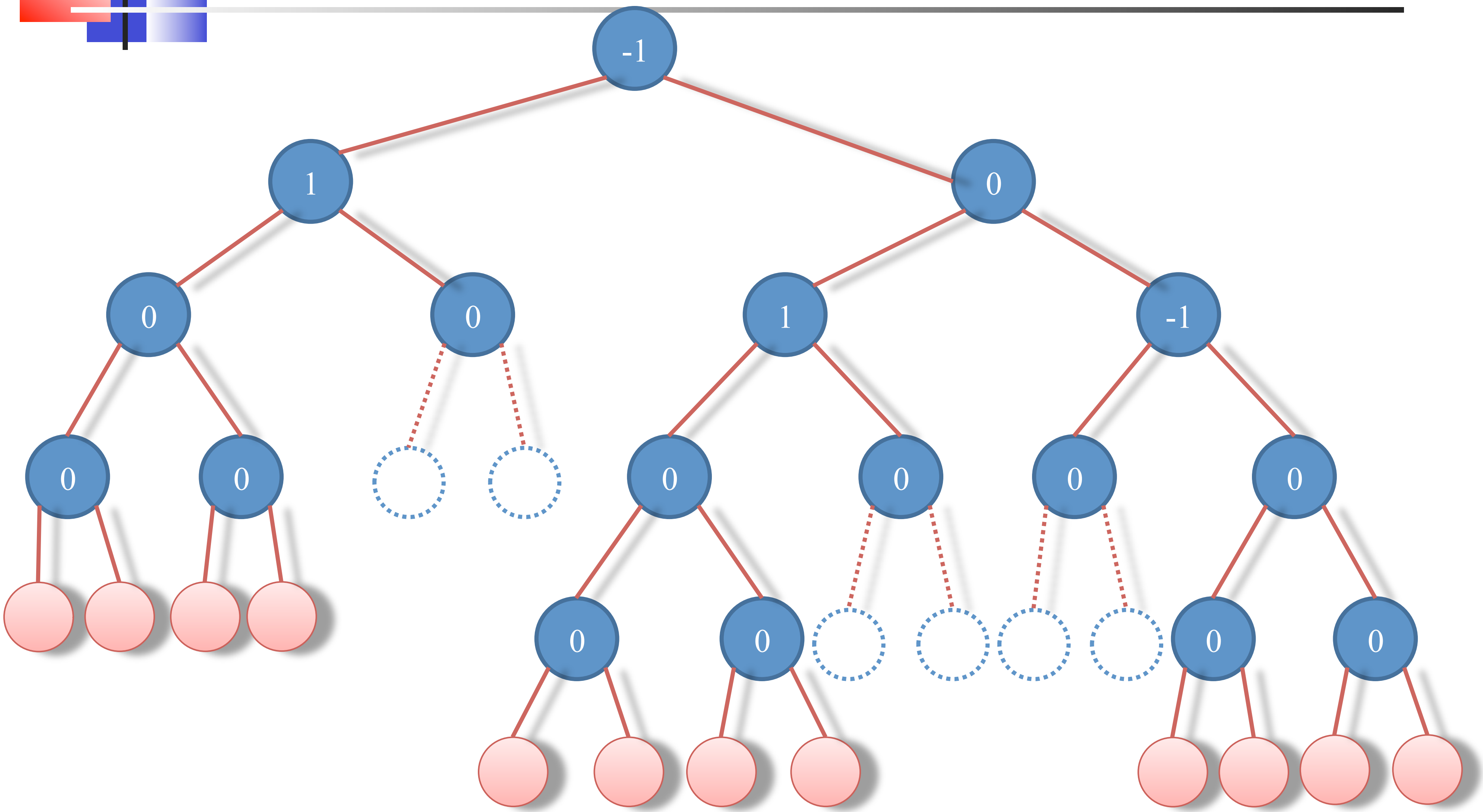
Árvores AVL – inserções



Árvores AVL – inserções - balanceada

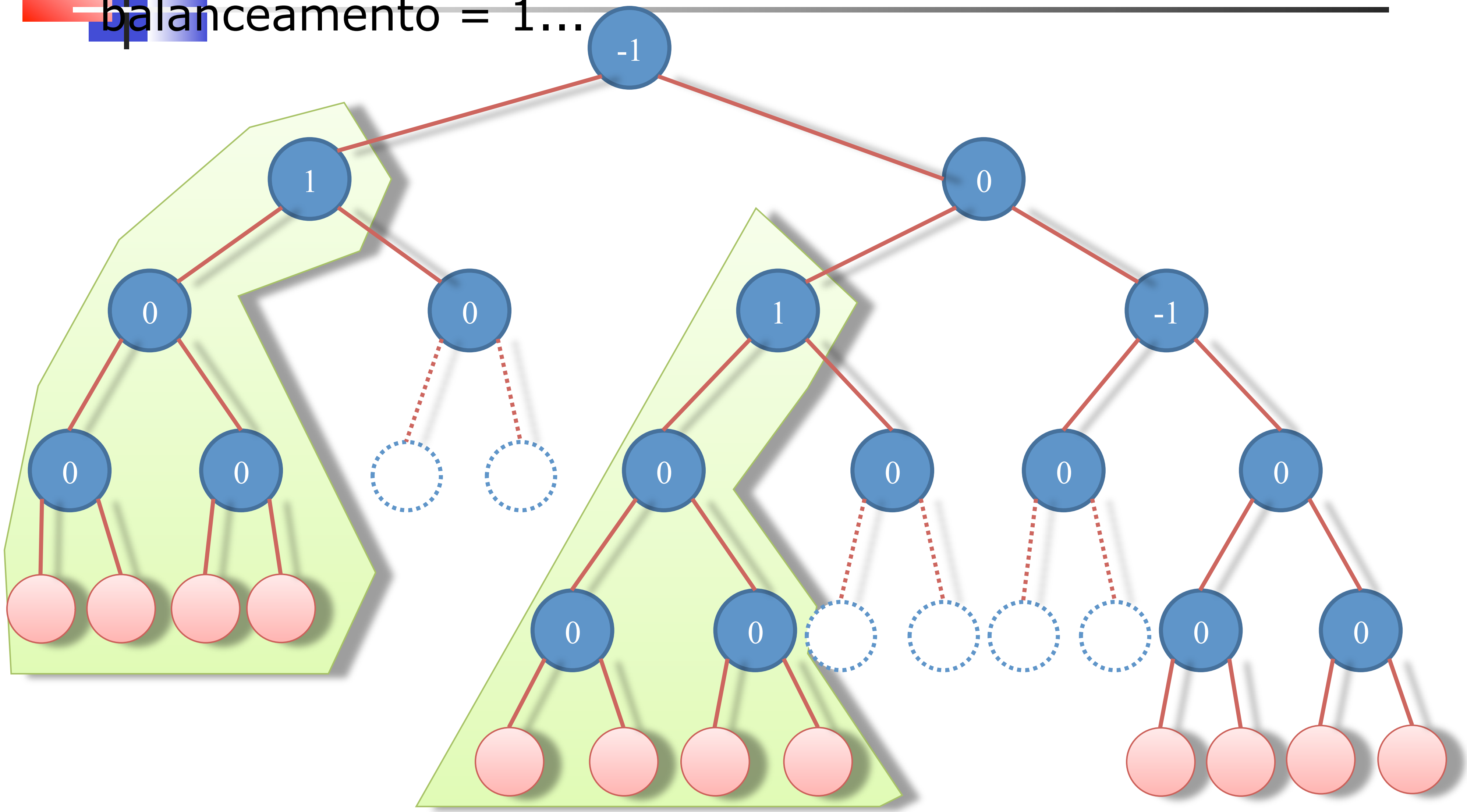


Árvores AVL – inserções - desbalanceada

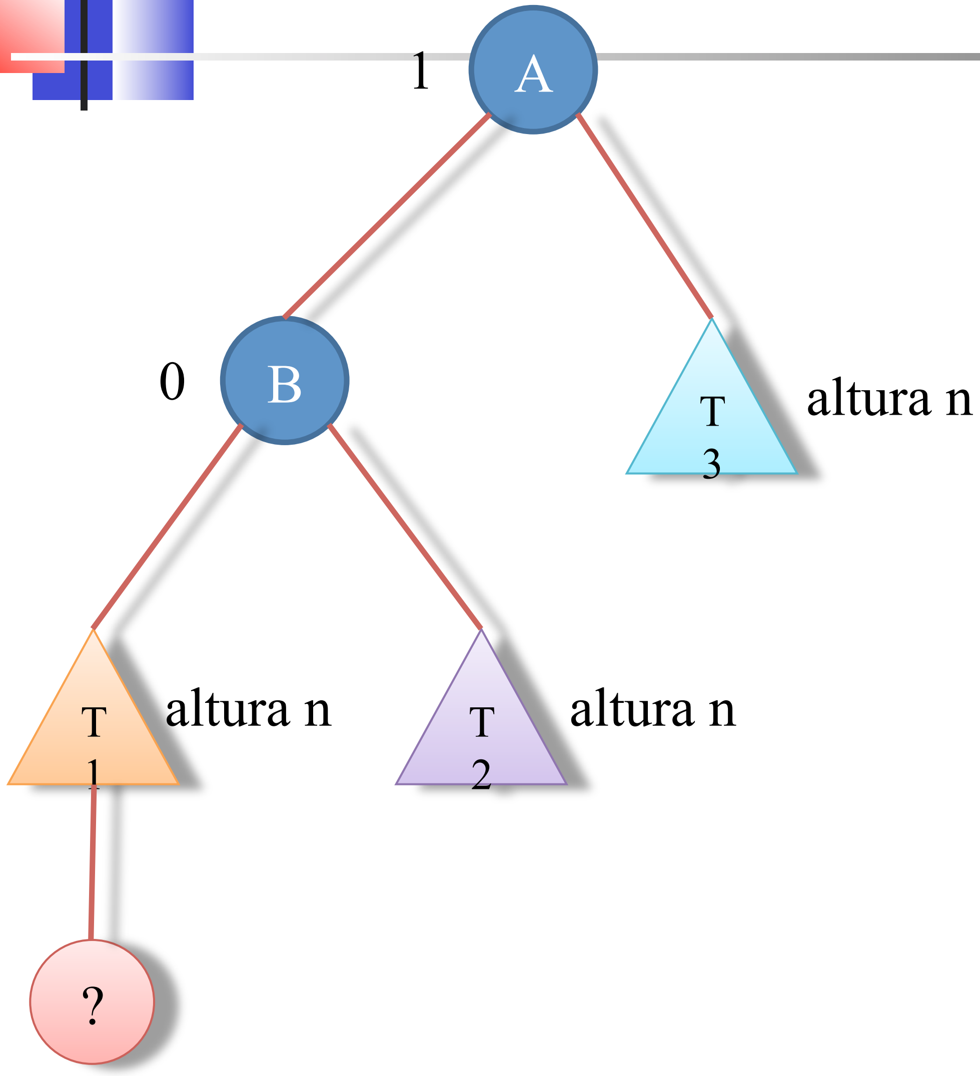


Árvores AVL – inserções - desbalanceada

Apenas se o nó inserido for à esquerda de um nó com balanceamento = 1...

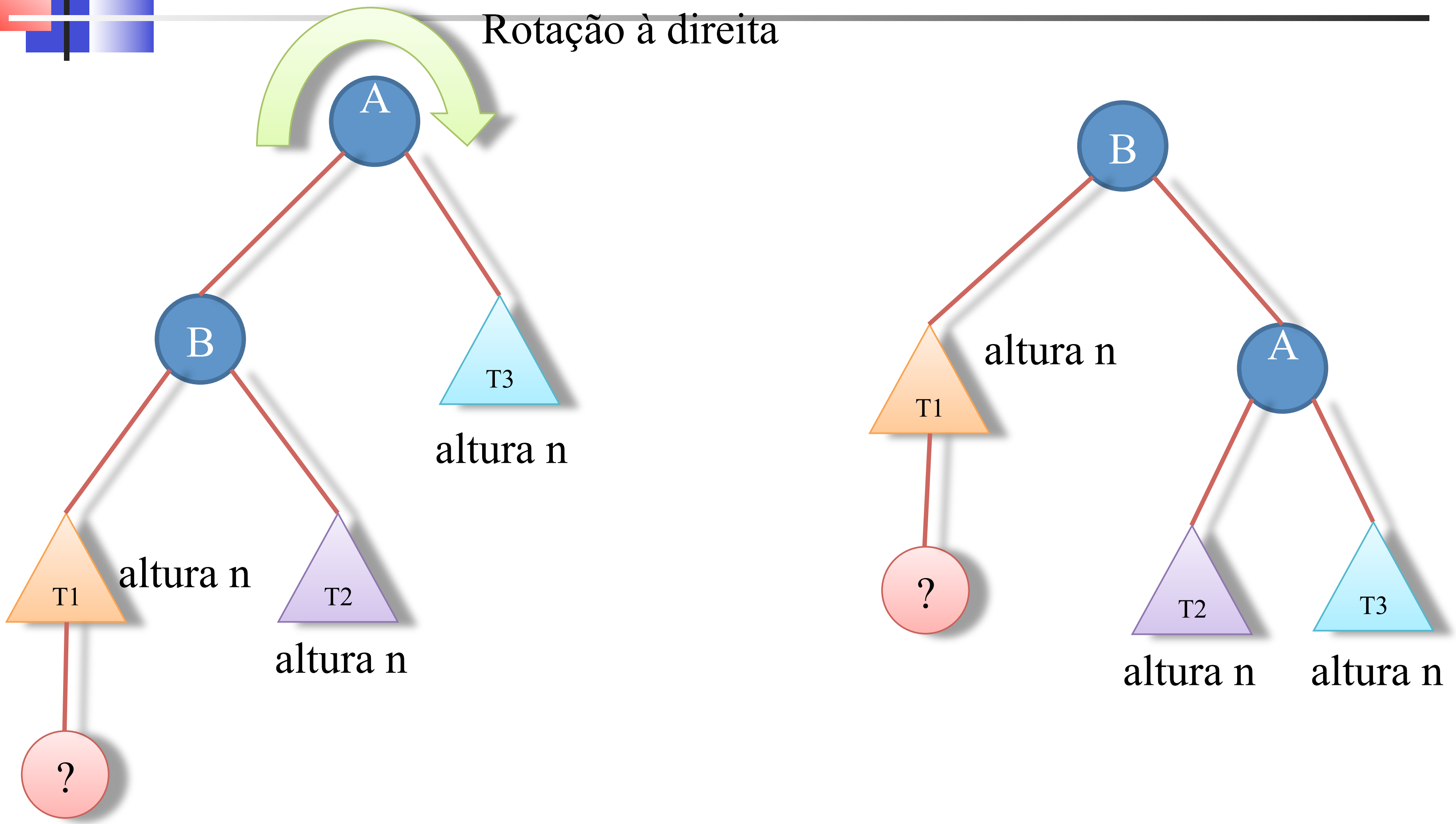


Caso 1: Raiz com bal = 1, nó na SAE

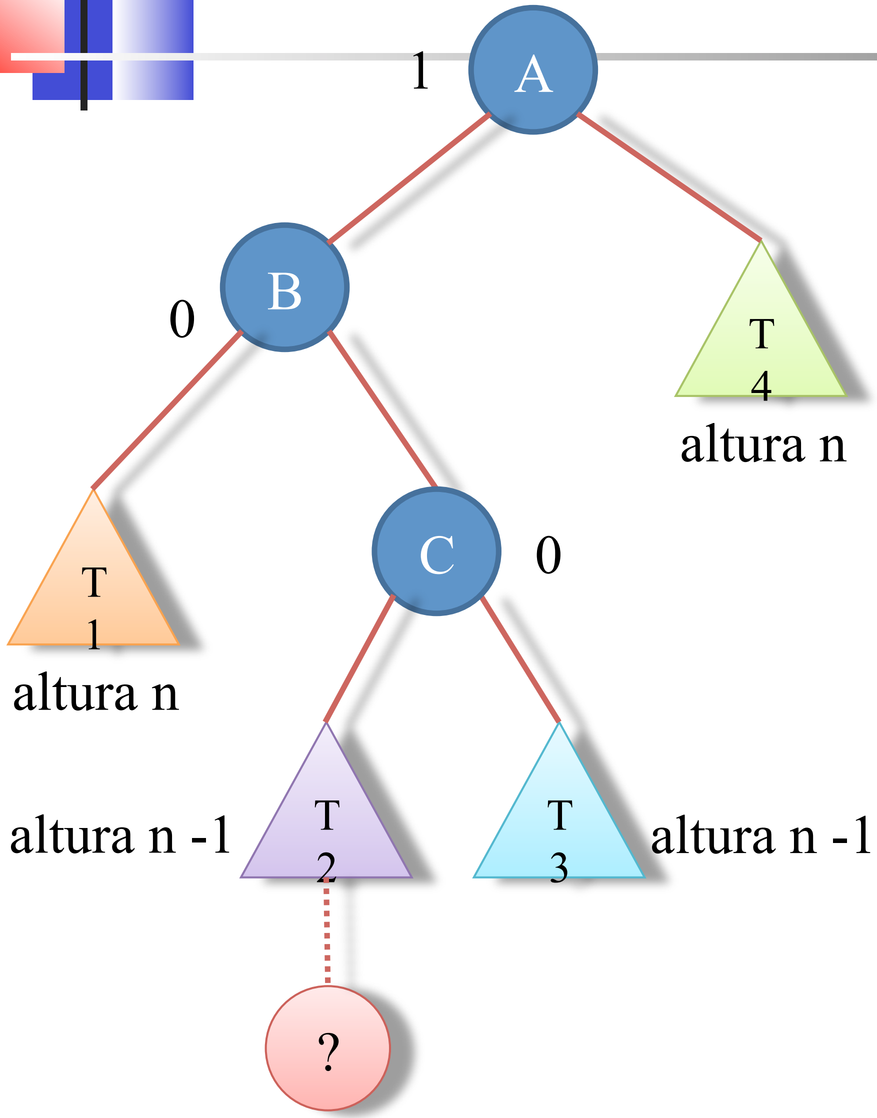


É necessária uma rotação à direita para balancear a árvore...

Caso 1: Raiz com bal = 1, nó na SAE



Caso 2: Raiz com bal = 1, nó na SAD

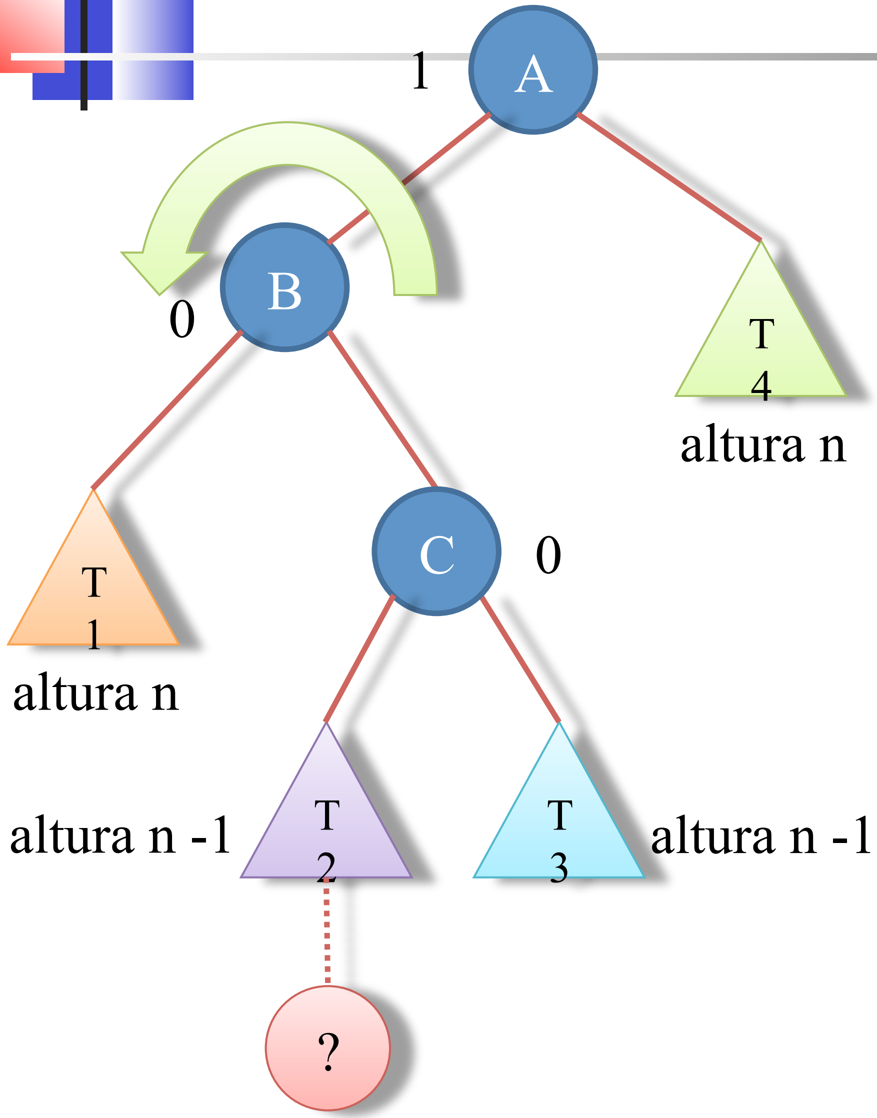


Rotação dupla:

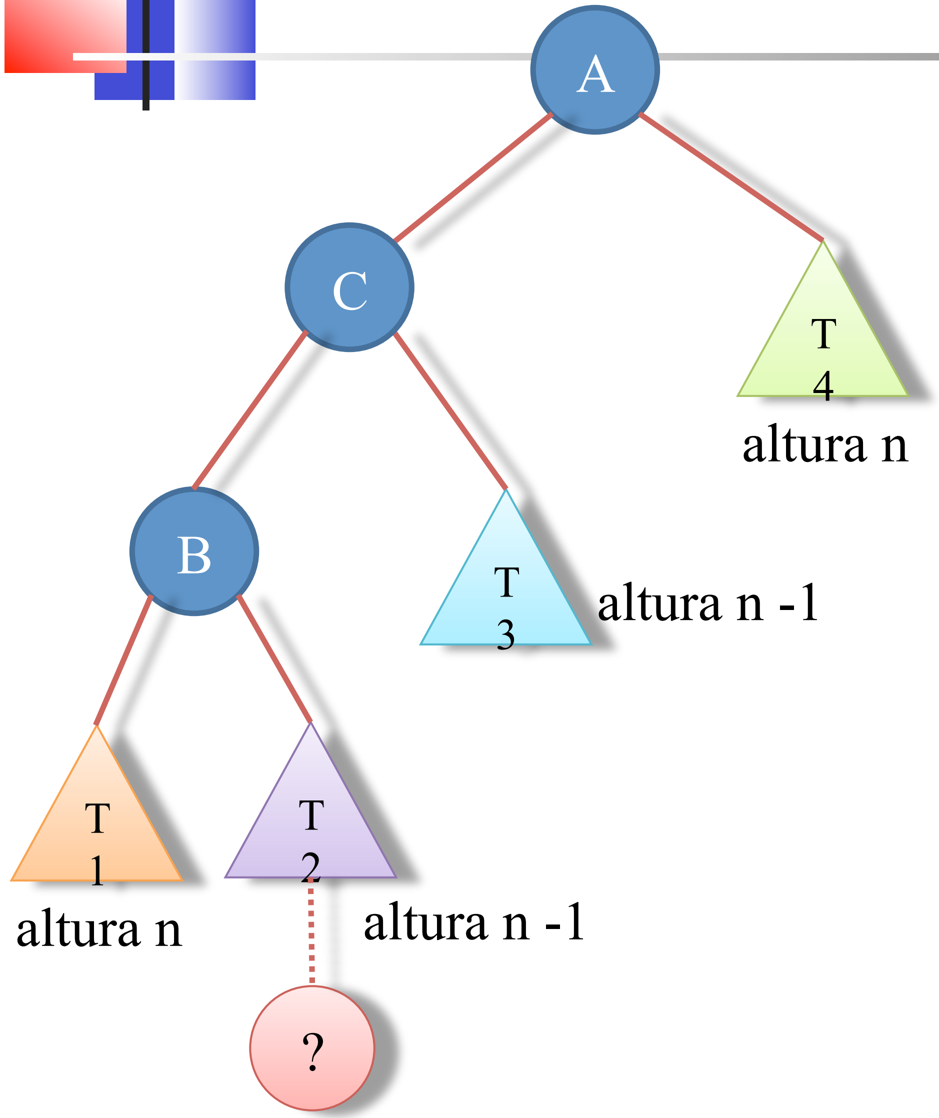
- rotação à esquerda na sub-árvore enraizada em B.

- uma rotação à direita na sub-árvore enraizada em A.

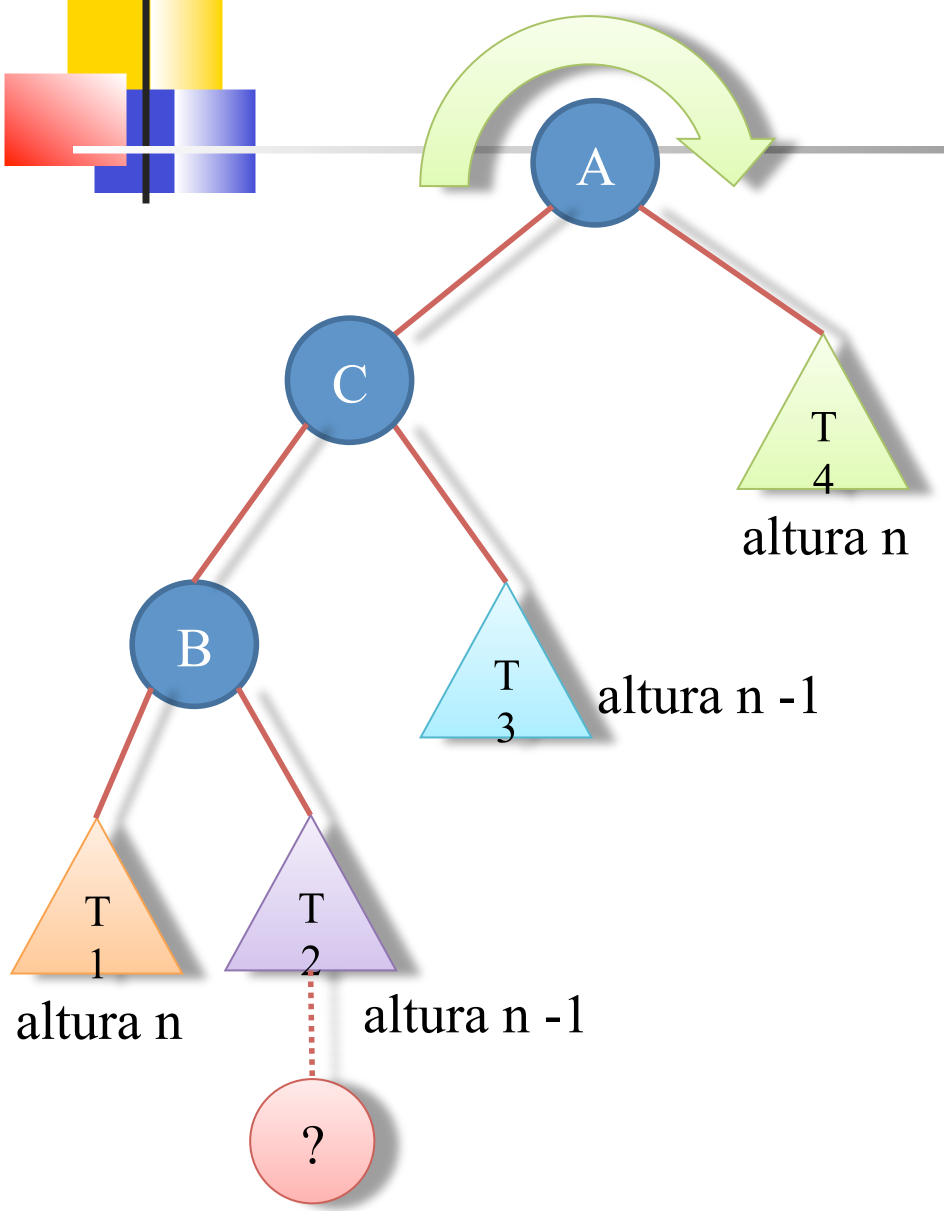
Caso 2: Raiz com bal = 1, nó na SAD



Caso 2: Raiz com bal = 1, nó na SAD



Caso 2: Raiz com bal = 1, nó na SAD



Caso 2: Raiz com bal = 1, nó na SAD