Árvores AVL

Definição; Uso e operações de rotação para restauro do equilíbrio
 Casos 1 e 4: Rotações simples.
 Casos 2 e 3: Rotações duplas.

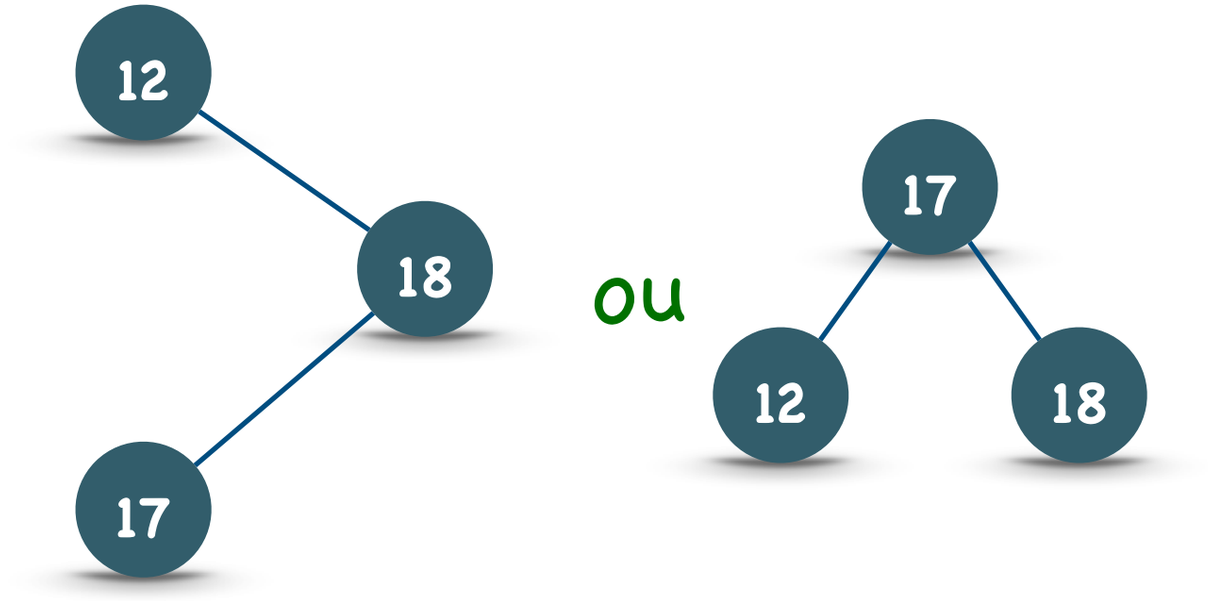
13-Maio-2021

Equilíbrio

- Problema? Seja n o número de nós da árvore
  - Normalmente operações $O(log(n))$
  - No entanto, no pior caso temos um embaraçoso: $O(n)$
    - Inserção de valores já ordenados
    - Sucessão de pares insere/remove
- Problema? Causa?
  - desequilíbrio

Equilíbrio

- Solução?
- Estrutura de dados(ABP?/outra) que se auto-equilibre
  - Nas operações que possam estragar o equilíbrio tentar restaurá-lo
- É essa a ideia das árvores AVL porque já sei que



são árvores equivalentes(têm os mesmos nós!)

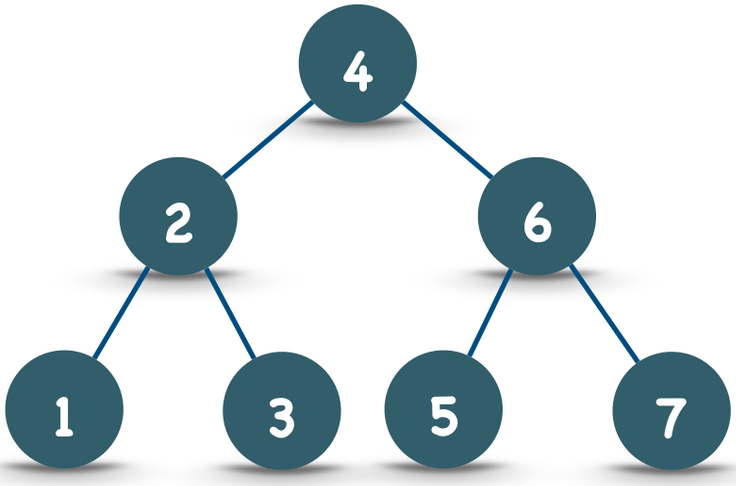
mas …têm alturas diferentes

Altura da árvore?

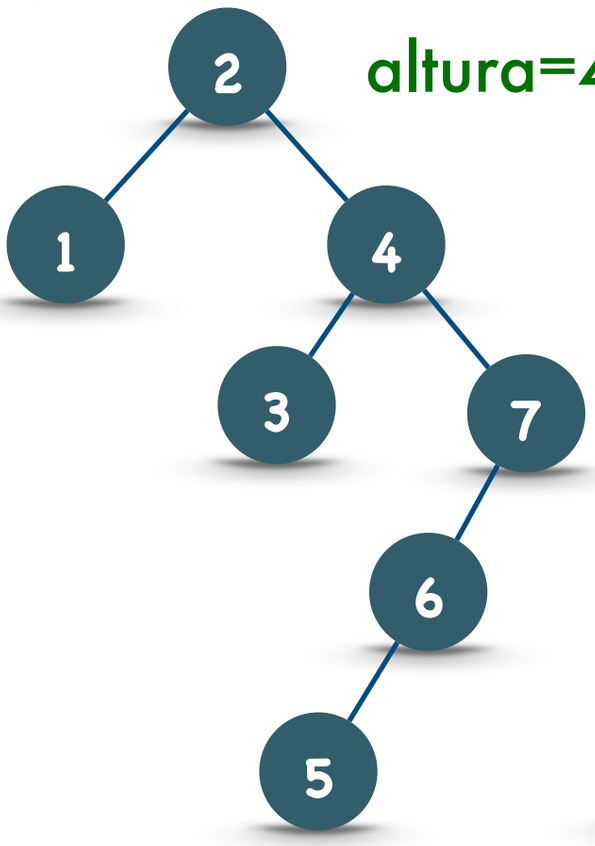
- input 4;2;6;1;3;5;7
- input 2;1;4;3;7;6;5
- input 7;6;5;4;3;2;1

Altura das árvores?

altura=2



altura=4



altura=6

Árvores AVL

- Ideia de Adelson-Velskii e Landis (1962)
  - Árvore binária com uma condição de equilíbrio
  - Procurar condições de equilíbrio (relativamente) fáceis de manter
- Objectivo: altura da árvore $O(log(n))$

Árvores AVL

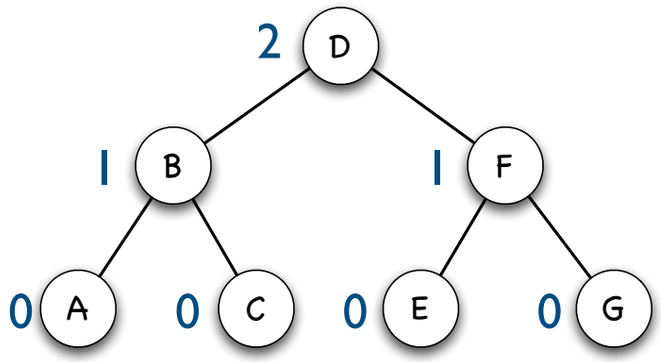
- Recurso: associar a cada nó uma função altura que retorna a altura do nó

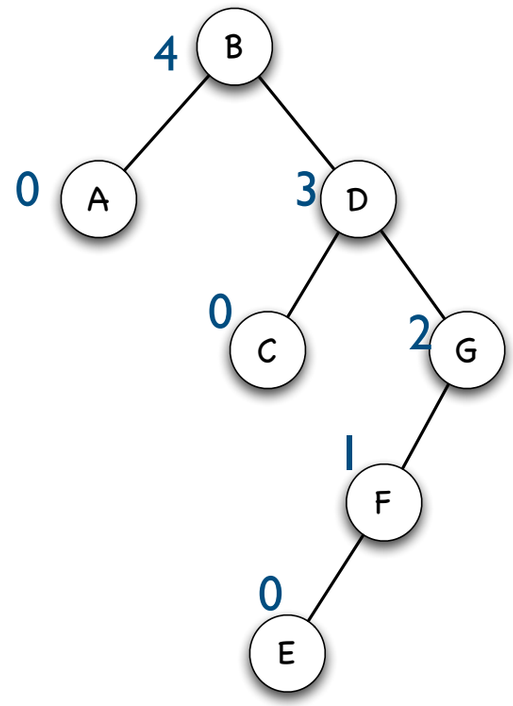
$$Altura(A) = \begin{cases} -1, & vazio(A) \\ 1 + max(Altura(Esq), Altura(Dir)), & otherwise \end{cases}$$

- Fazer com que as sub-árvores esquerda e direita nunca tenham alturas muito diferentes

Altura

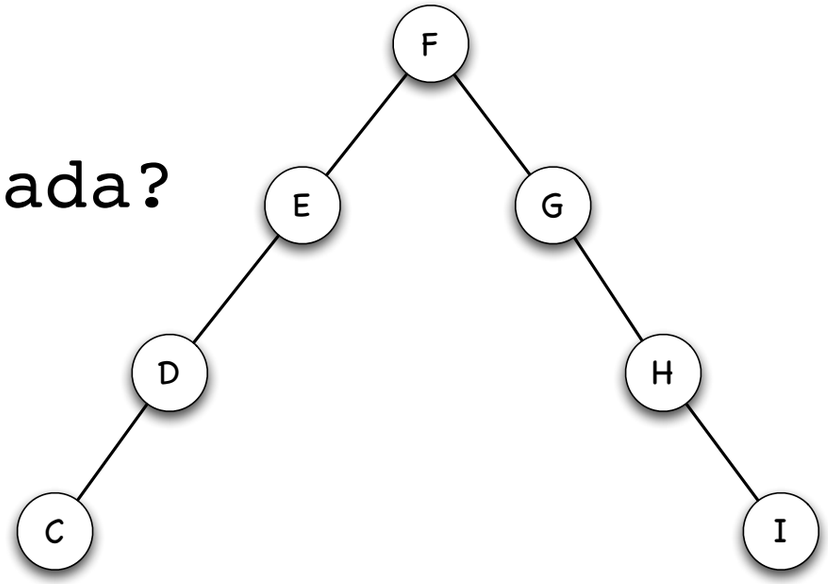
- Como calcular a altura das árvores usando a informação nos nós?
  - input D;B;F;A;C;E;G
  - input B;A;D;C;G;F;E





Árvores AVL

- Como obter árvores equilibradas?
  - Garantir que a raíz é equilibrada?
    - Contra-exemplo



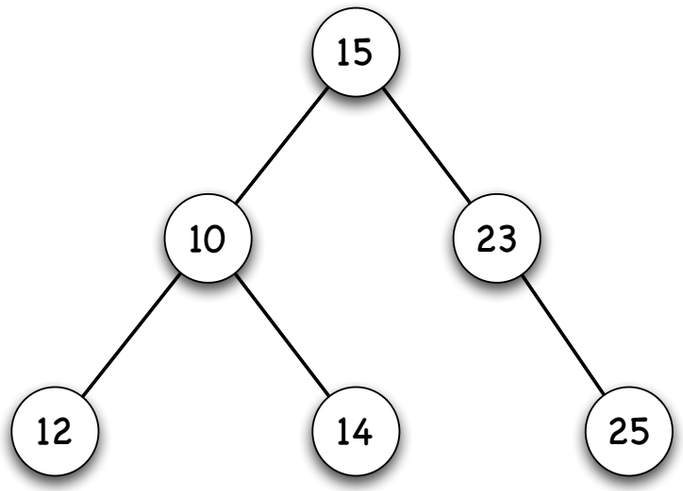
- Exigir que a altura das sub-árvores não difiram mais de 1 unidade
  - É a condição de equilíbrio das árvores AVL
  - Requer tratamento especial nas operações de inserção e remoção

Árvores AVL

- Com esta condição:
  - Pode-se demonstrar que
    - Altura máxima duma árvore AVL (i.e. que satisfaça a condição de equilíbrio) é no máximo
      - $1.44log(N+2) - 0.328$
    - Em média será $log(N+1) + .25$
- Na prática, ficará sempre $O(log(N))$, mesmo no pior caso

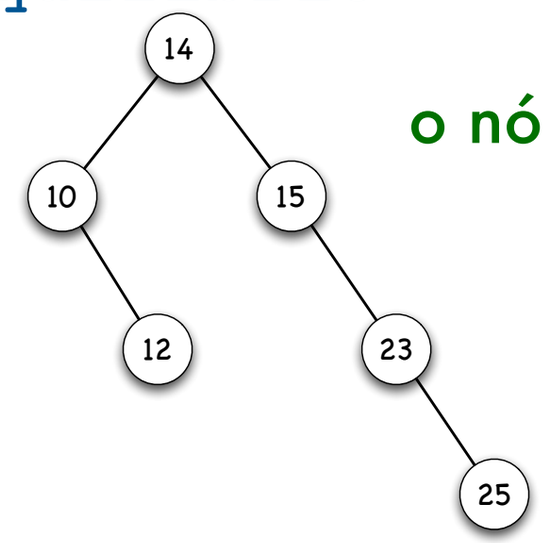
Árvores AVL

- É uma AVL?



As árvores têm exactamente os mesmos nós

- Exemplo duma ABP, mas não uma AVL: um nó que viola a restrição de equilíbrio



o nó 15 viola a restrição de equilíbrio

Árvores AVL

- Com profundidade máxima O(log N)
  - Operações que não modificam serão T=O(log N)
  - Operações que modificam (insere, remove)… ainda não se sabe
    - Se usarmos "remoção preguiçosa" , a operação remove também será garantidamente O(log N)
- Resta o "osso": a operação de inserção

Árvores AVL

- Insere-se como numa ABP "normal"
  - Há que manter a informação de altura
    - Pode ser explícito no nó (=O(1)), em detrimento duma função que avalie.

- A inserção pode causar um desequilíbrio
  - Ao inserir um valor, só os nós no caminho entre a raiz e o novo nó é que poderão ter a sua condição de equilíbrio alterada
  - Há que restabelecer o equilíbrio antes de terminar a inserção

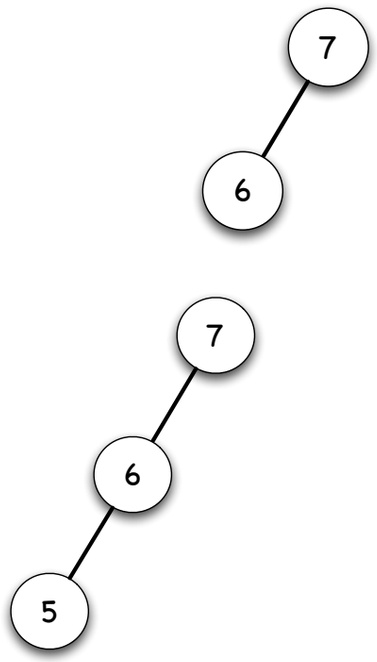
Árvores AVL

- Ao inserir um novo nó, fá-lo-emos numa folha
- É preciso reavaliar a altura dos nós acima
  - Faz-se de baixo para cima
  - Se não houver nenhum que viole a condição de equilíbrio, já está
  - Se houver um nó em que a condição não seja satisfeita:
    - Há que reequilibrar a sub-árvore cuja raiz é esse nó
    - Não é preciso fazer mais nada acima

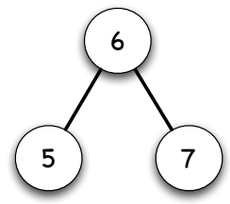
Árvores AVL

- Inserir 5 em

- obtemos



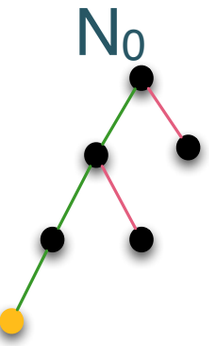
- está desequilibrada.Como obter uma árvore com os mesmos nós mas equilibrada. Isto é a AVL ?



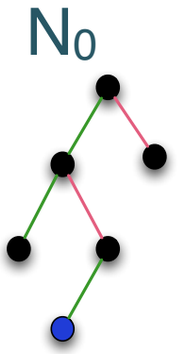
Árvores AVL

- Ao reequilibrar, estamos a fazê-lo numa árvore que era anteriormente equilibrada

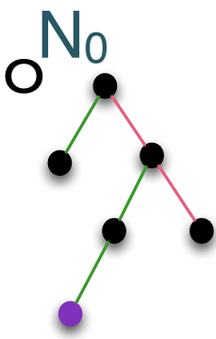
- Seja N_0 o nó onde detectámos o desequilíbrio



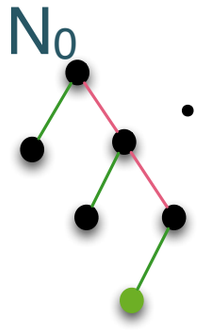
  - caso **1**: Inserção na sub-árvore **esquerda** do filho **esquerdo** de N_0



  - caso **2**: Inserção na sub-árvore **direita** do filho **esquerdo** de N_0



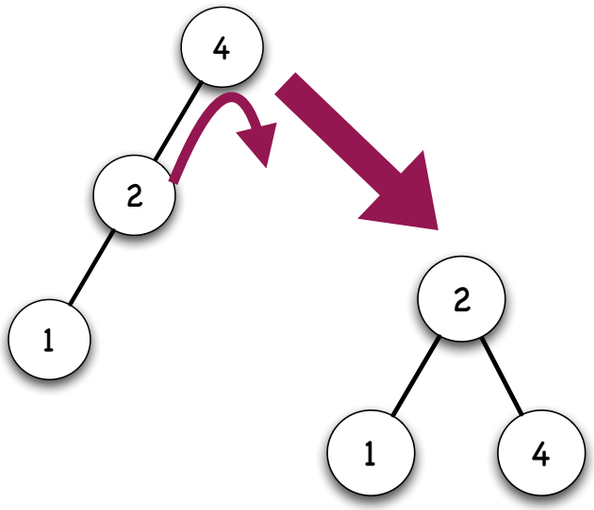
  - caso **3**: Inserção na sub-árvore **esquerda** do filho **direito** de N_0

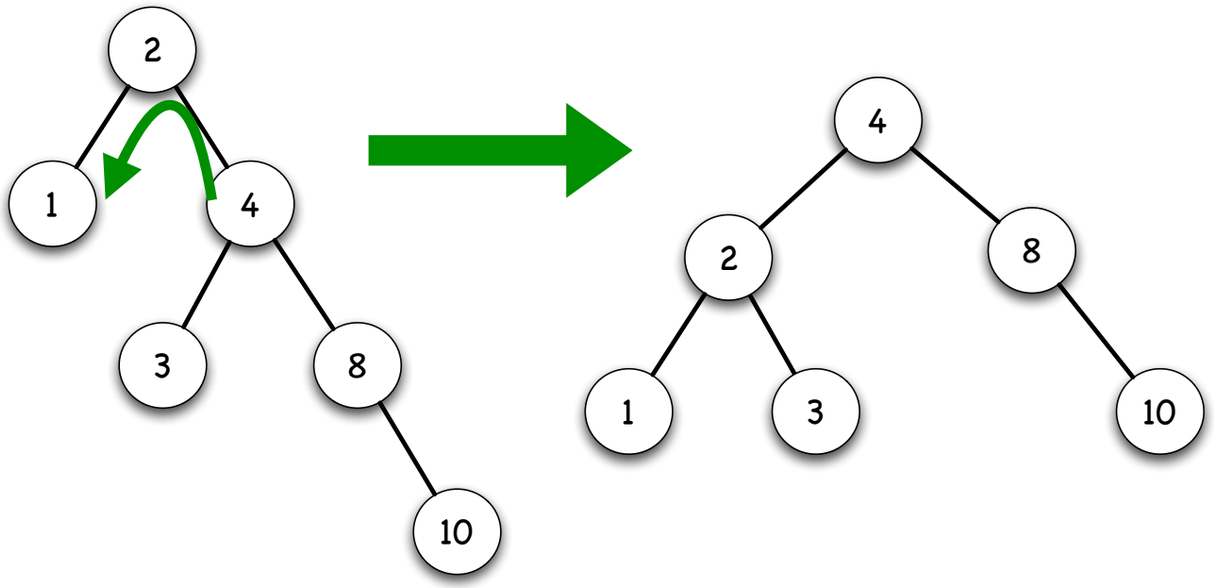


  - caso **4**: Inserção na sub-árvore **direita** do filho **direito** de N_0

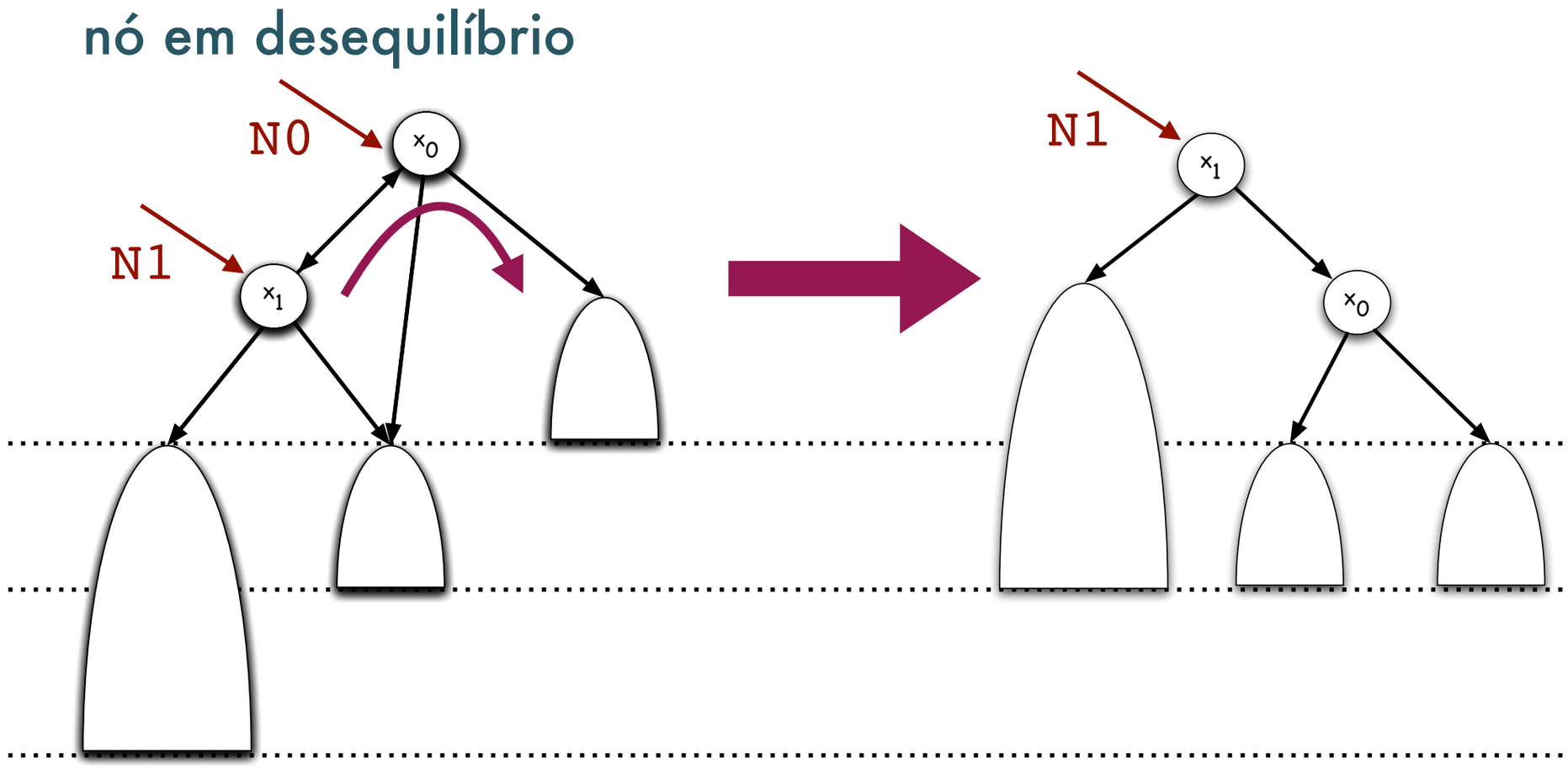
Árvores AVL

- Casos 1 e 4 (tudo à **esquerda**, tudo à **direita**)
  - Resolvem-se com uma rotação simples;





CASO 1 TUDO À ESQUERDA

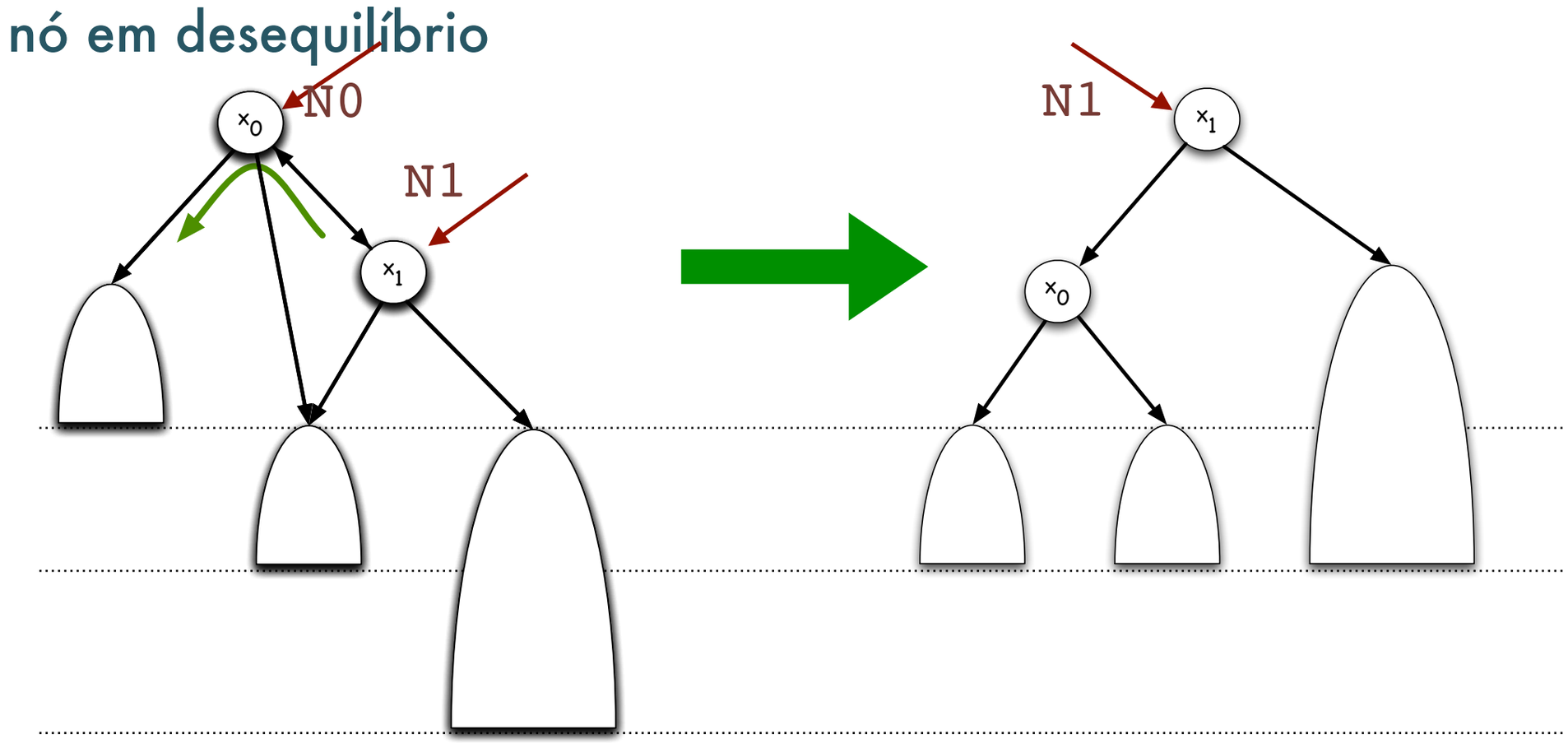



```
N1 = N0->Left;
N0->Left = N1->Right;
N1->Right = N0;
return N1;
```

Árvores AVL

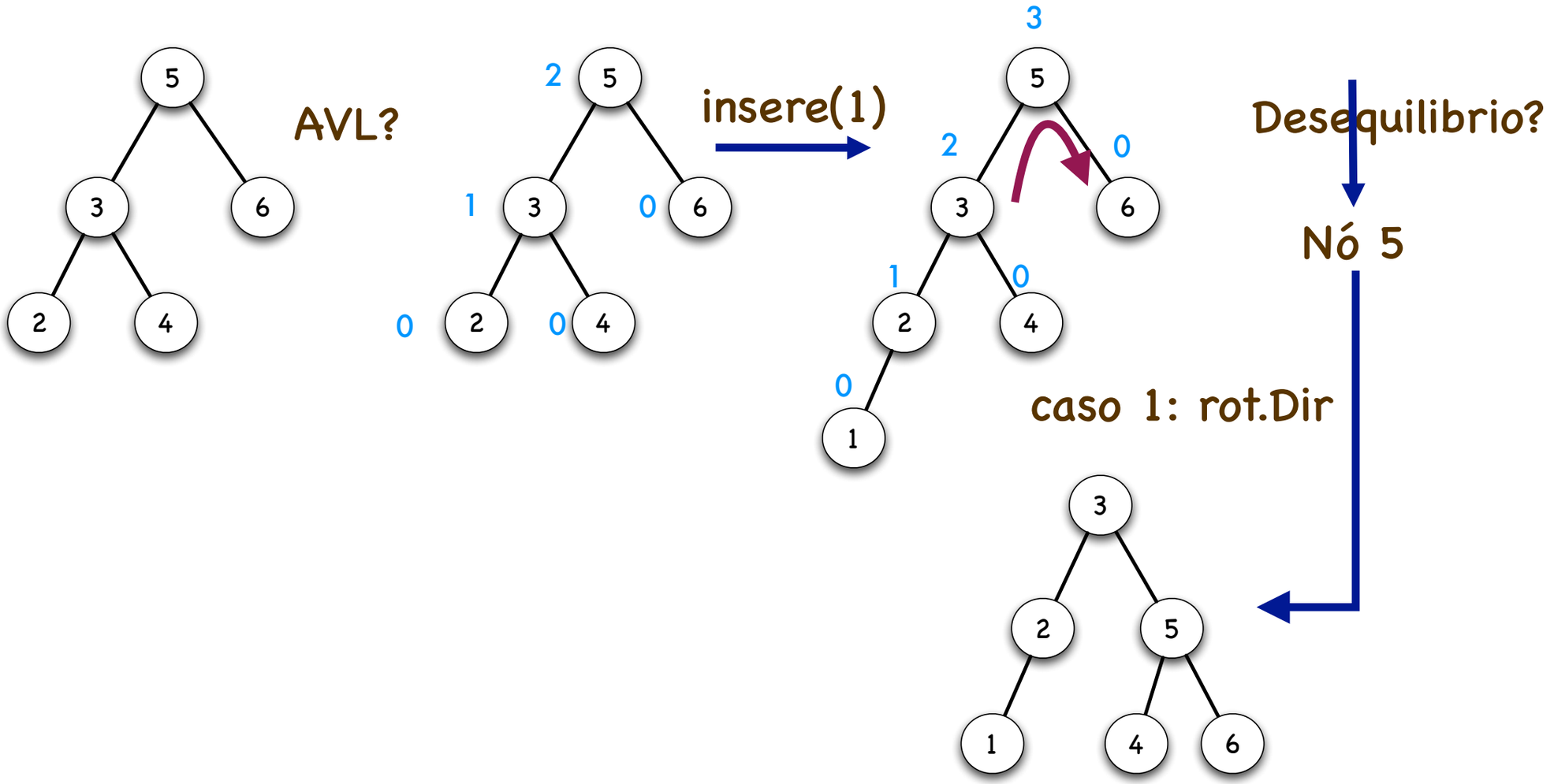
```
static Position RSD( Position N0 ){
    Position N1;
    printf("Nó desequilibrio: %d\n",N0->Element);
    printf("Rotação simples direita - caso 1 EE\n");


    N1 = N0->Left;
    N0->Left = N1->Right;
    N1->Right = N0;

    N0->Height = Max( Height( N0->Left ), Height( N0->Right ) ) + 1;
    N1->Height = Max( Height( N1->Left ), N0->Height ) + 1;

    return N1;  /* New root */
}
```

Caso 4 - Tudo à direita

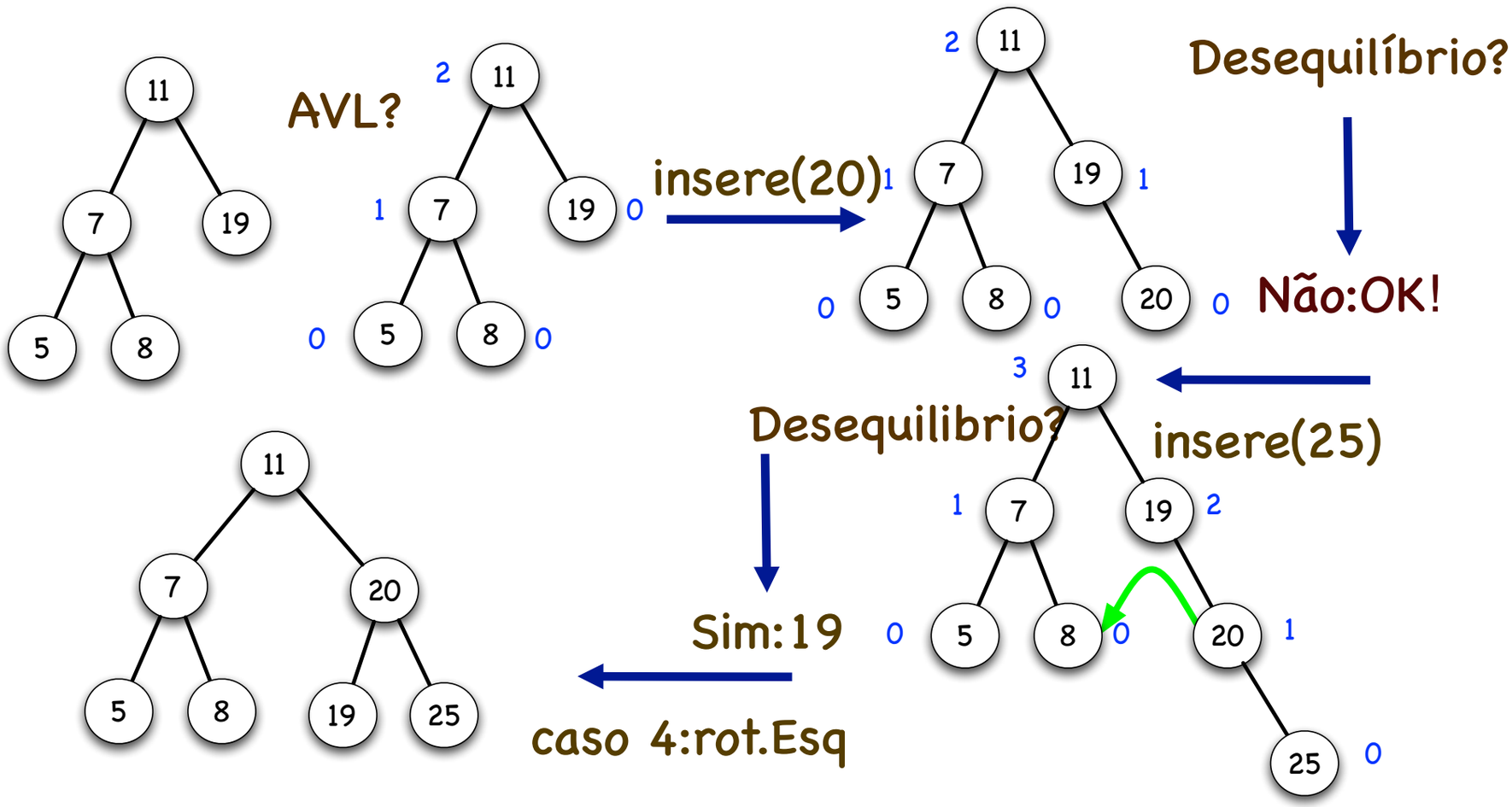



```
N1 = N0->Right;

N0->Right = N1->Left;

N1->Left = N0;

return N1;
```

Exemplo

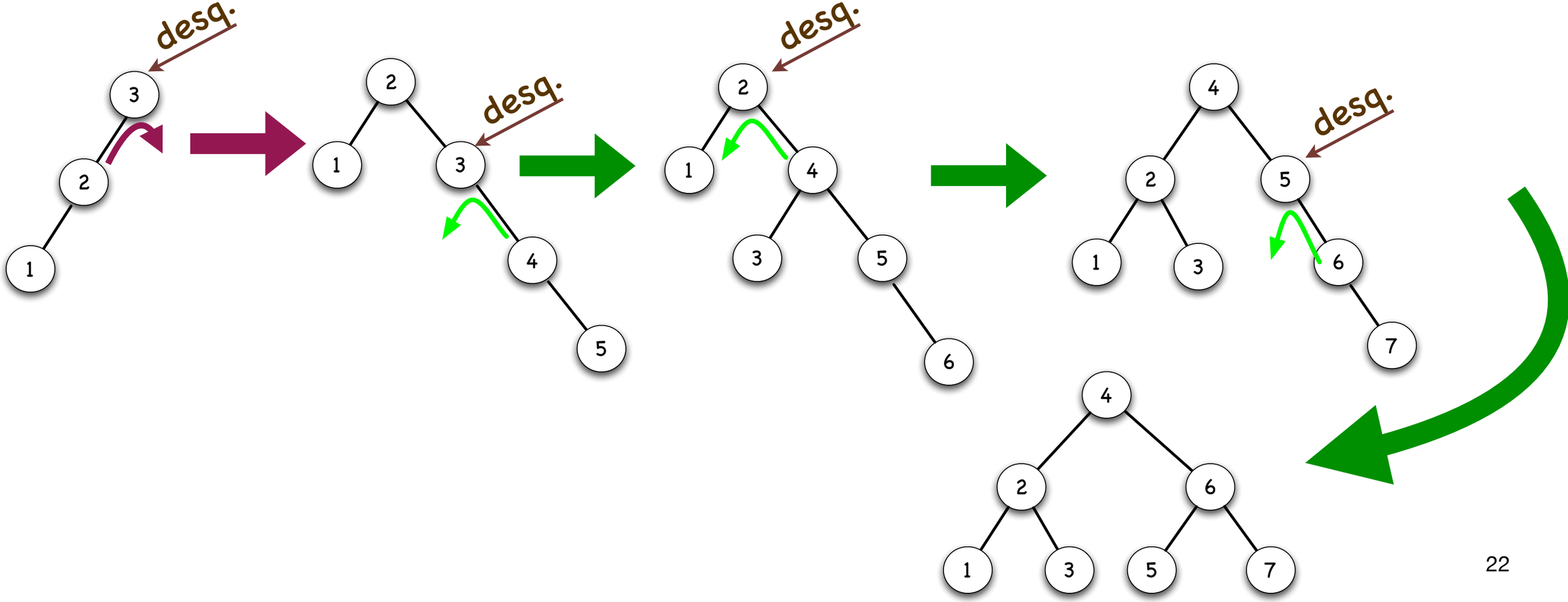

5
3
6
2
4
AVL?
2
5
1
3
0
6
0
2
0
4
insere(1)
3
5
2
0
3
6
1
0
2
4
0
1
Desequilibrio?
Nó 5
caso 1: rot.Dir
3
2
5
1
4
6

Exemplo


AVL?
insere(20)
Desequilíbrio?
Não:OK!
insere(25)
Desequilibrio?
Sim:19
caso 4:rot.Esq

Casos Exteriores

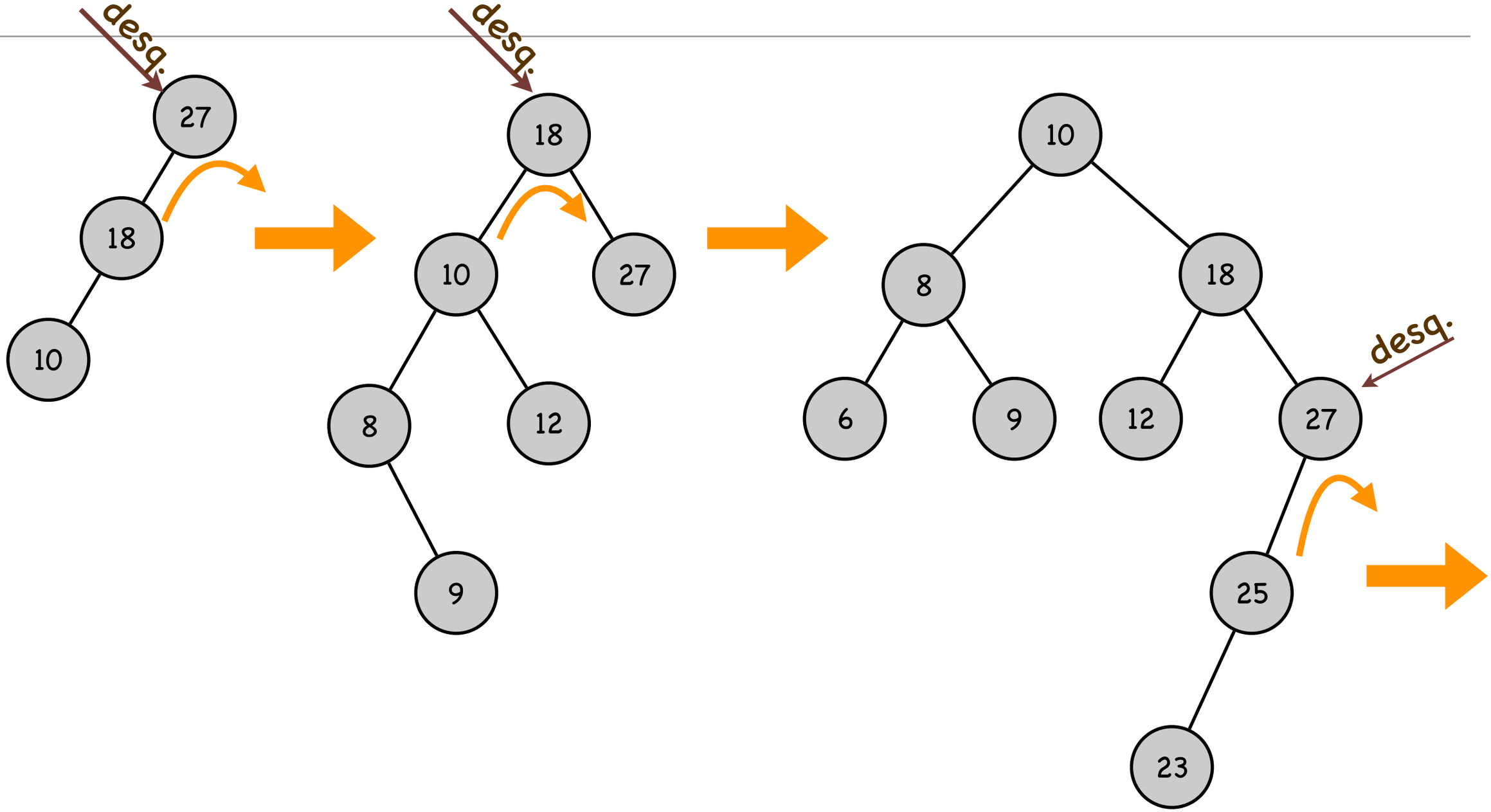
- Como exemplo, considere-se a inserção numa árvore AVL vazia de:
  - 3, 2, 1, 4, 5, 6, 7



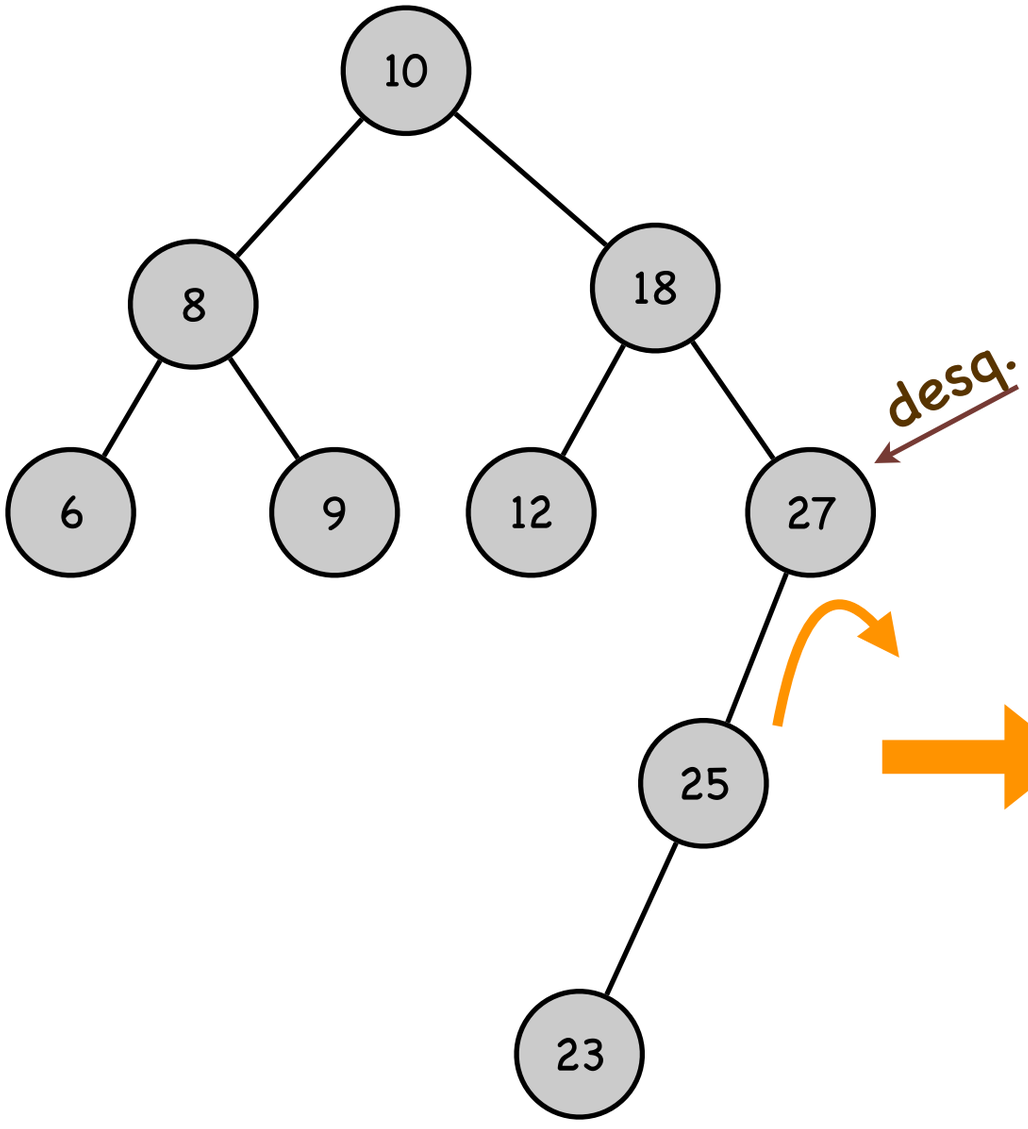
AVL - CASOS EXTERIORES

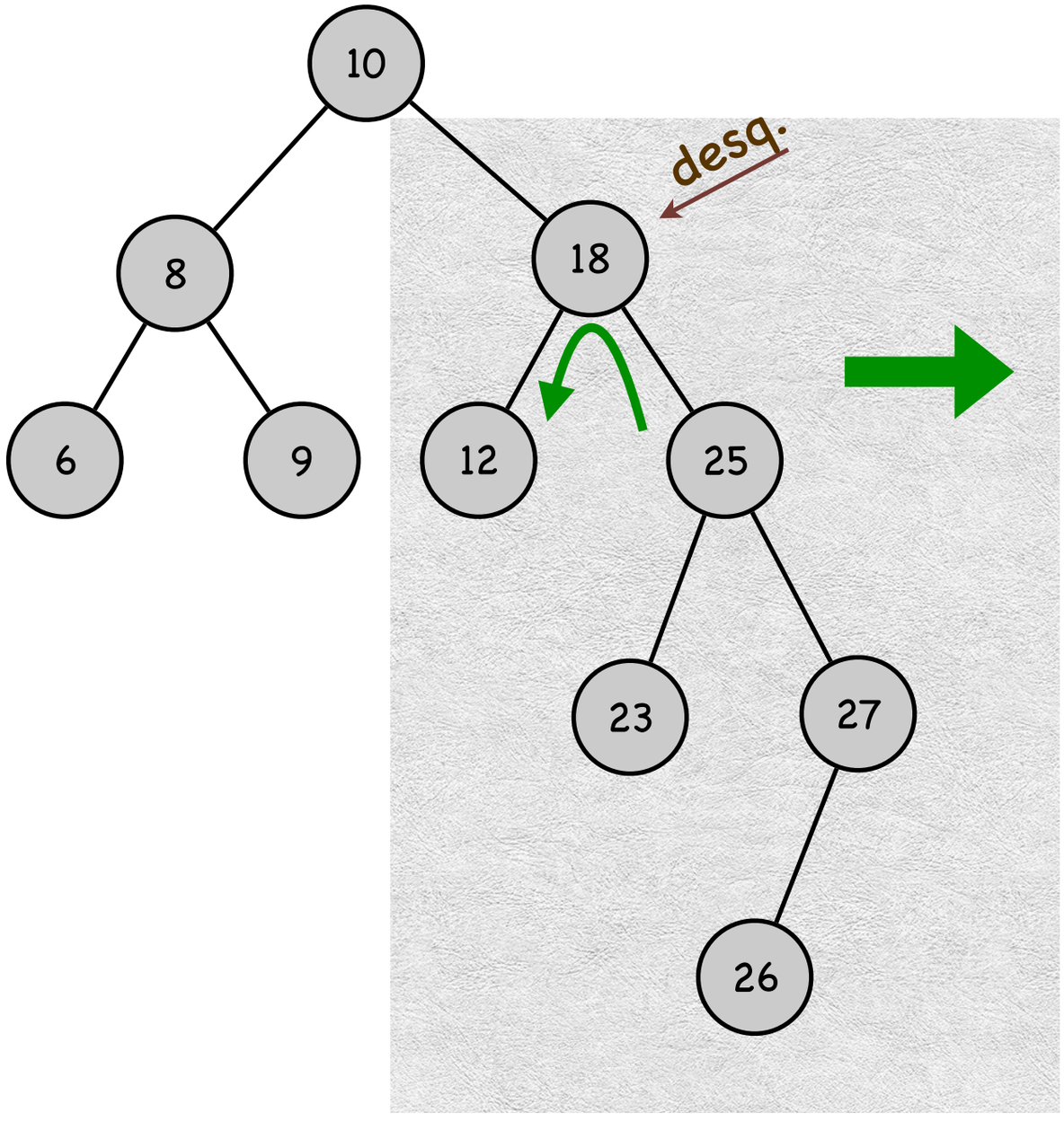
- Inserir numa AVL, inicialmente vazia os nós:
  - 27;18;10;12;8;9;25;6;23;26;30;29
  - Após a inserção deve indicar se houve desequilíbrio, qual o caso e se é possível resolvê-lo. Se sim, faça a respetiva rotação e prossiga. Senão pare!

Exercício resolvido

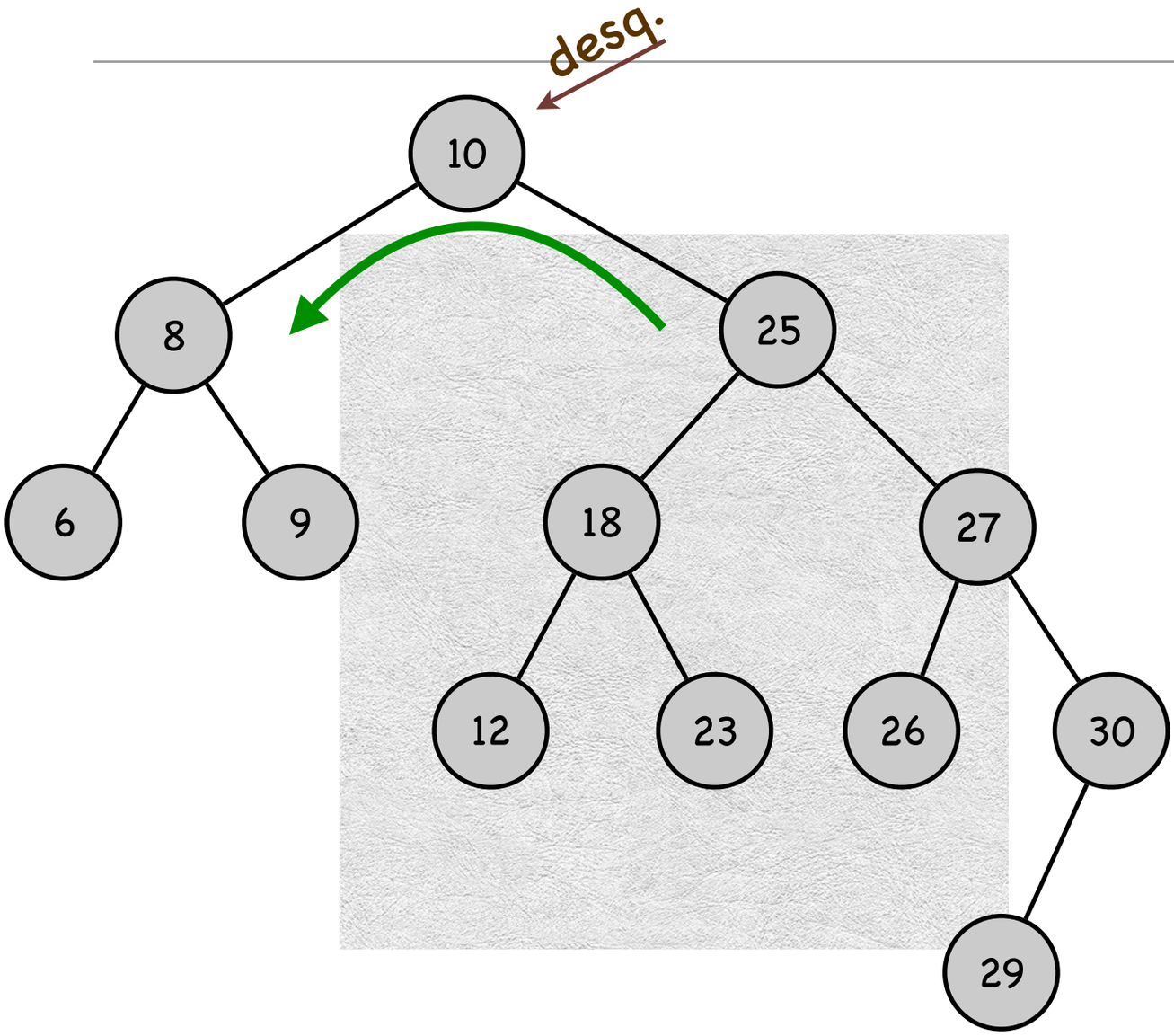


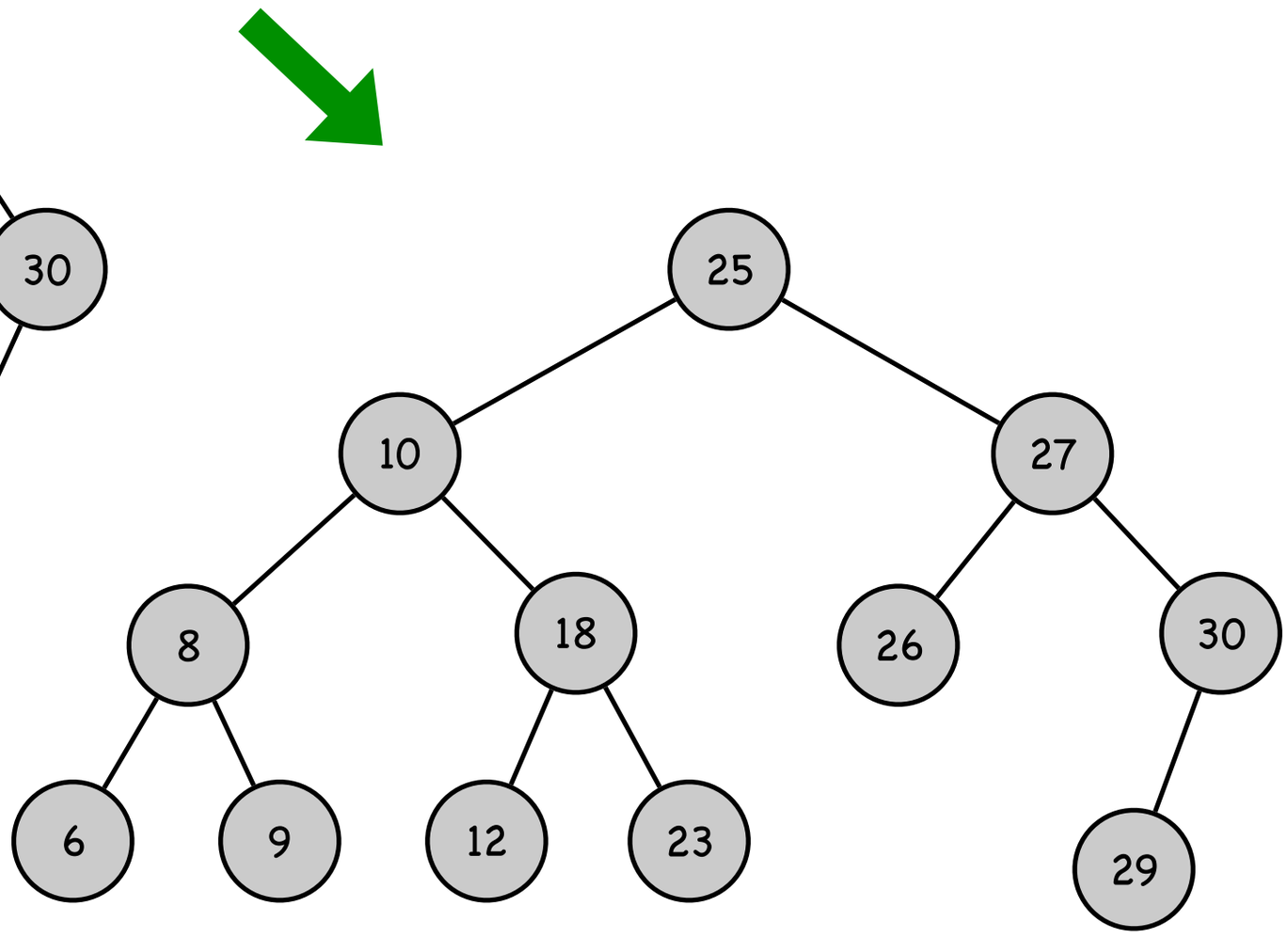
CONTINUAÇÃO





CONTINUAÇÃO





Árvores AVL

- Casos **2** e **3** (filho **direito** da sub-árvore **esquerda** e vice-versa)

- A rotação simples não resolve

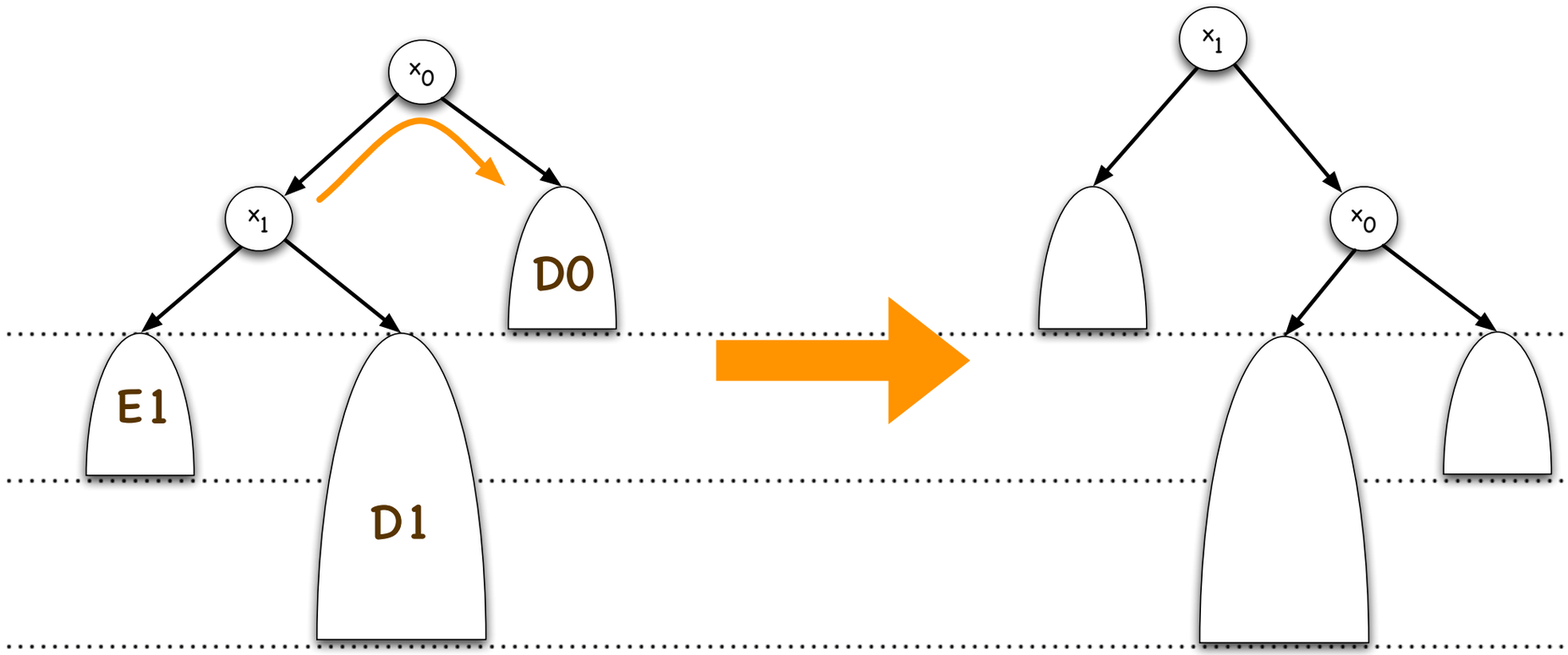
  - (ver pag. seguinte)

- Ou seja

  - Continuamos na mesma situação

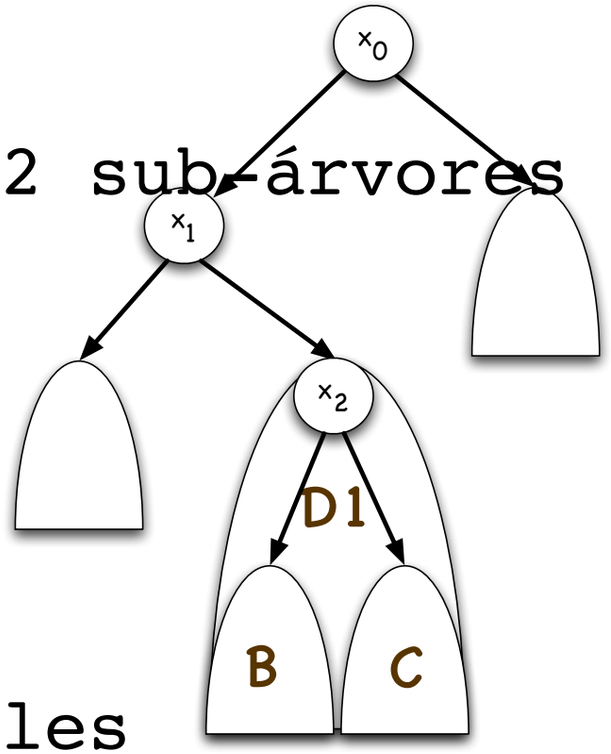
  - É preciso fazer outra coisa...

Casos Exteriores

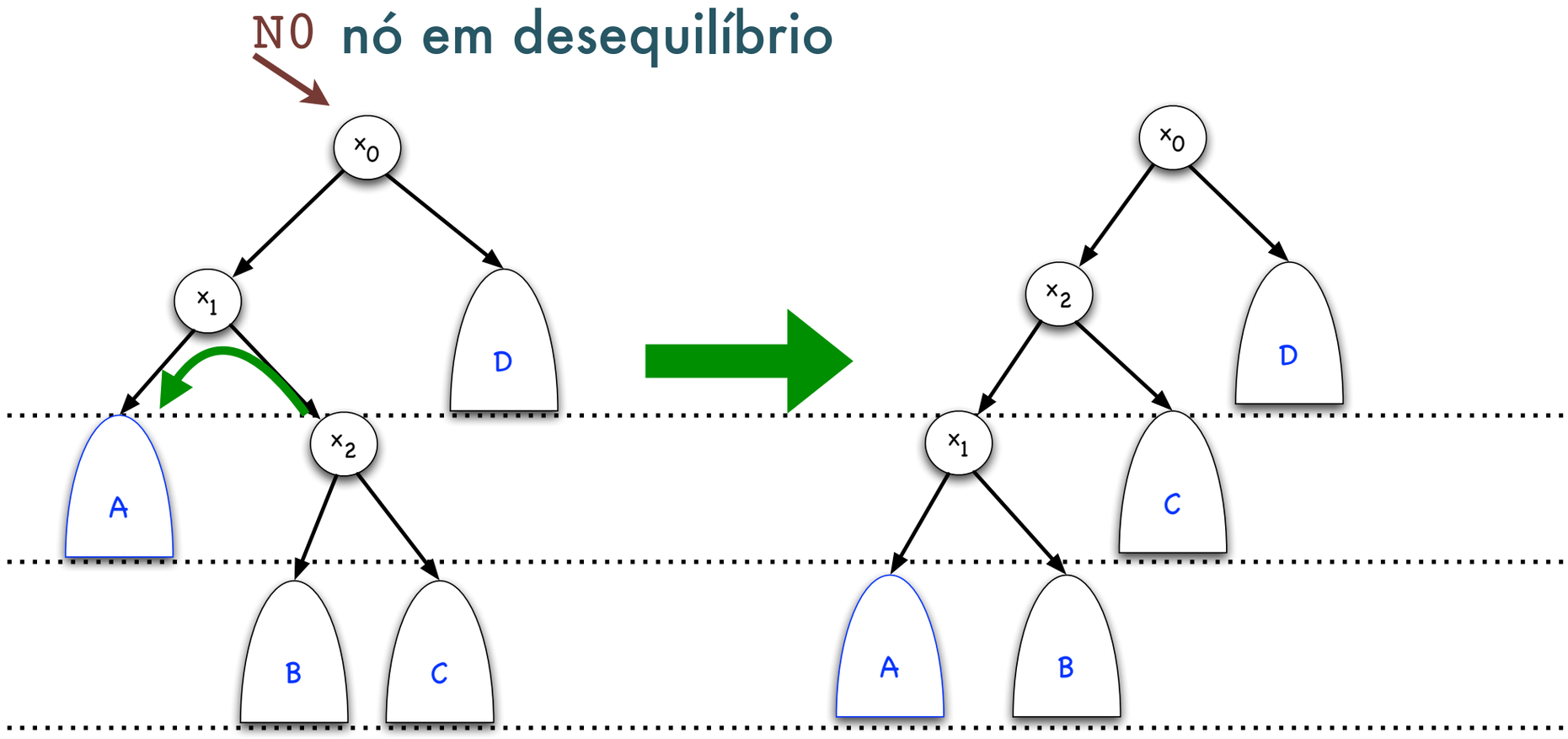


Árvores AVL

- Em vez de considerar 3 sub-árvores…
  - Vamos dividir a sub-árvore D1 em 2 sub-árvores
  - Precisamos de especificar 3 nós
- Iremos fazer uma rotação dupla
  - Mas à custa de duas rotações simples
  - Notar a indeterminação da altura das árvores mais altas (B e C)

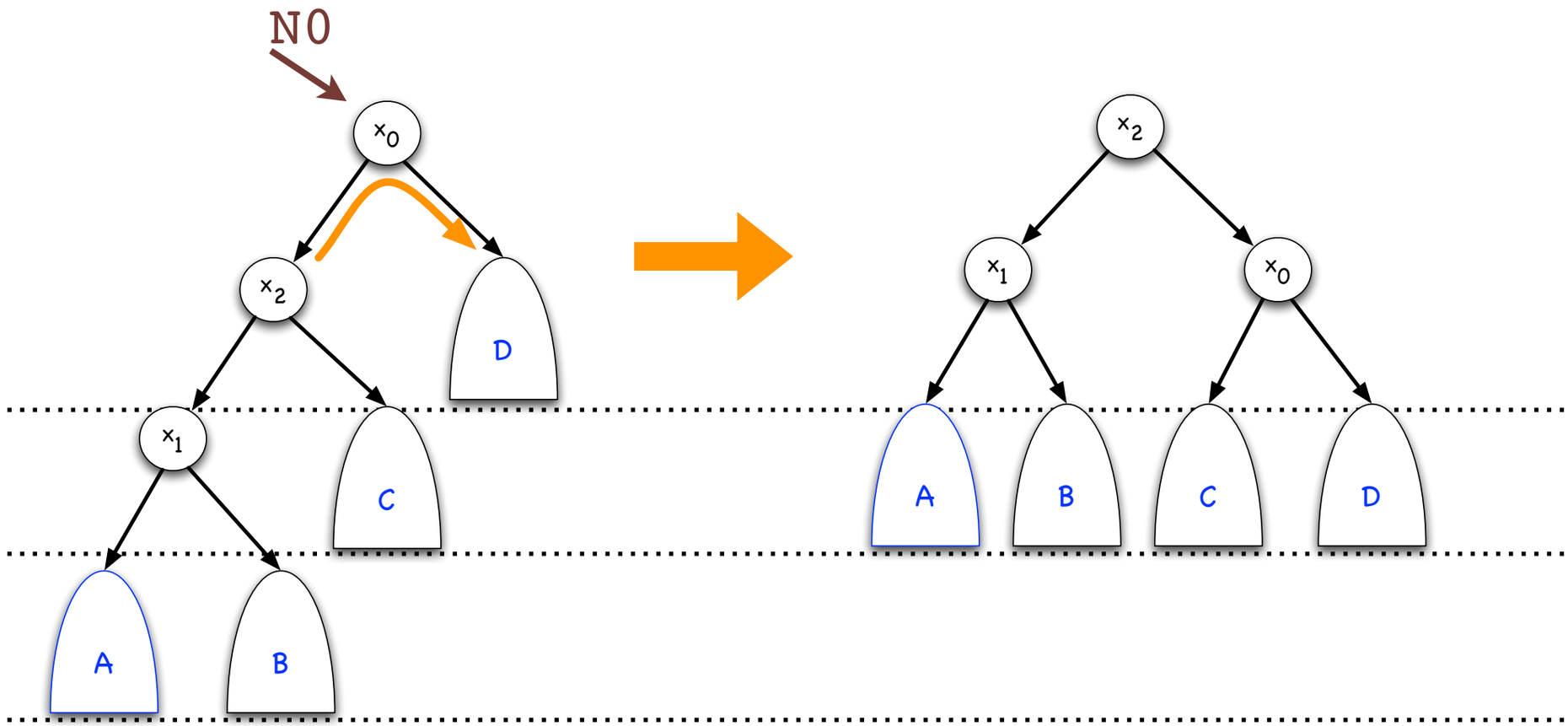


Casos Interiores - caso 2(esquerda direita)




```
N0->Left = RSE( N0->Left );
```

Casos Interiores - caso 2



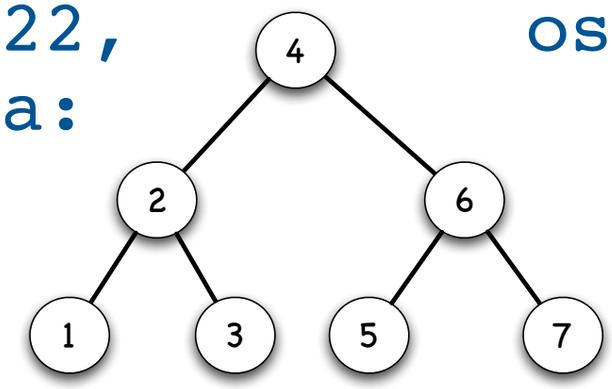

```
NO->Left = RSE( NO->Left );

return RSD( NO )
```

Rotação dupla (ED) - caso 2

```
static Position RDED( Position N0 ){

    printf("Nó desequilibrio: %d\n",N0->Element);
    printf("Rotação Dupla Esquerda Direita - caso 2 \n");
    N0->Left = RSE( N0->Left );

    return RSD( N0 );
}
```

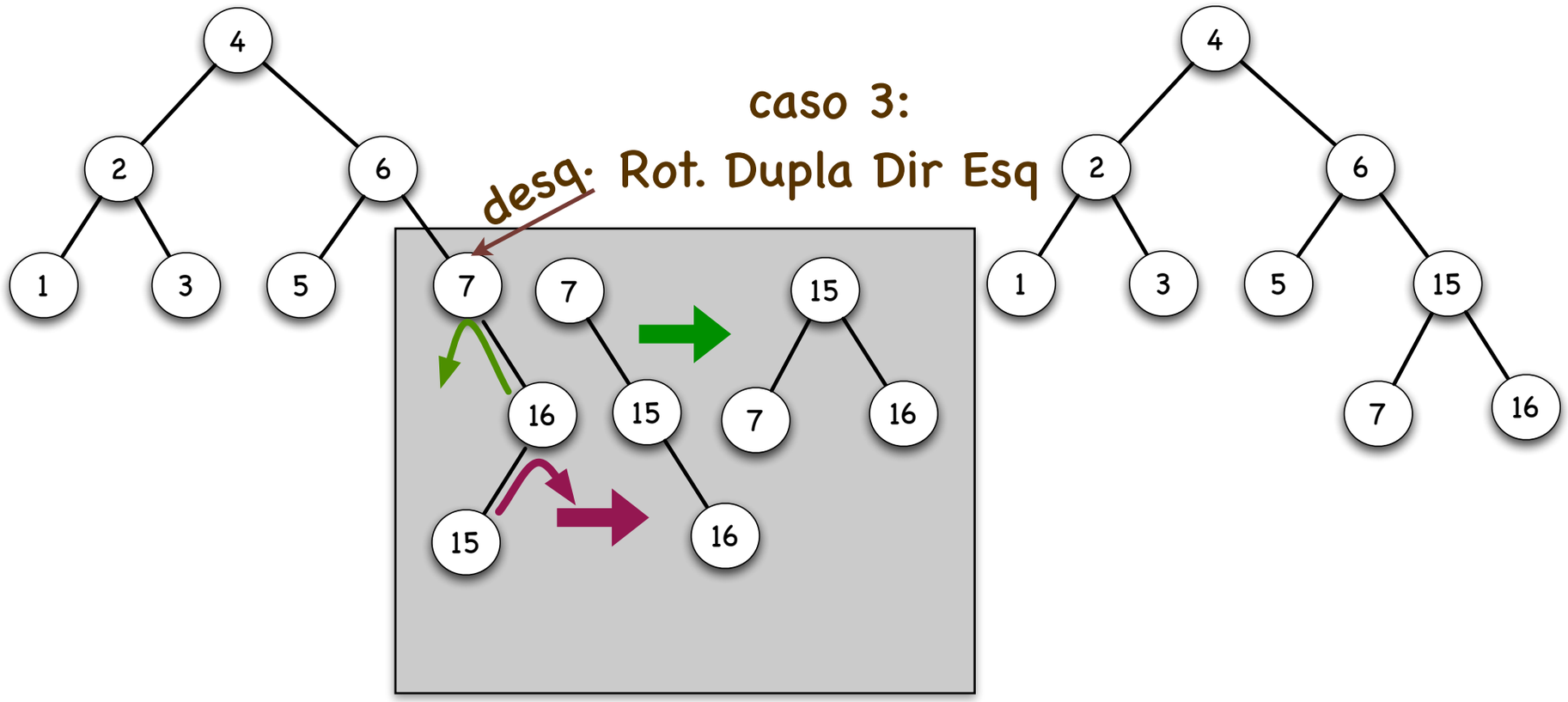
Casos interiores

- Realizar a inserção na árvore final do exercício da página 22, os seguintes valores, pela ordem indicada:

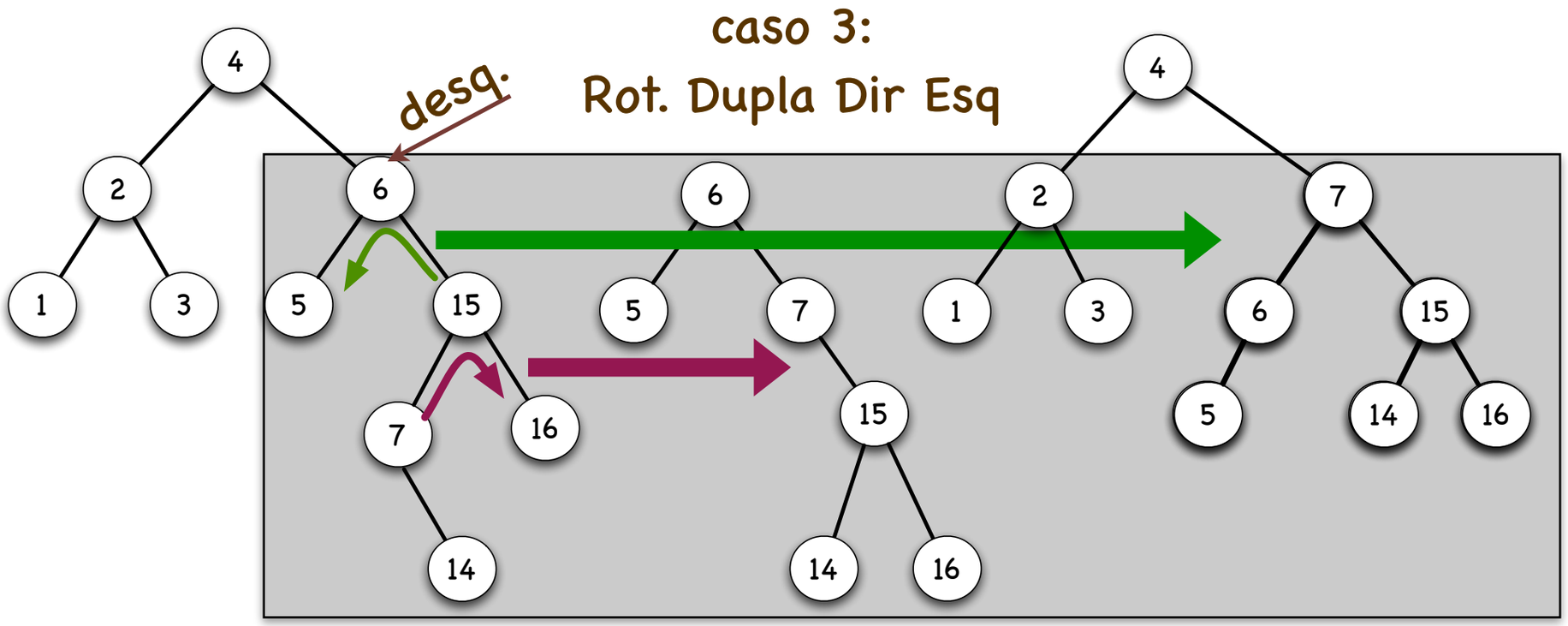


- 16, 15, 14, 13, 12, 11, 10
- 8, 9

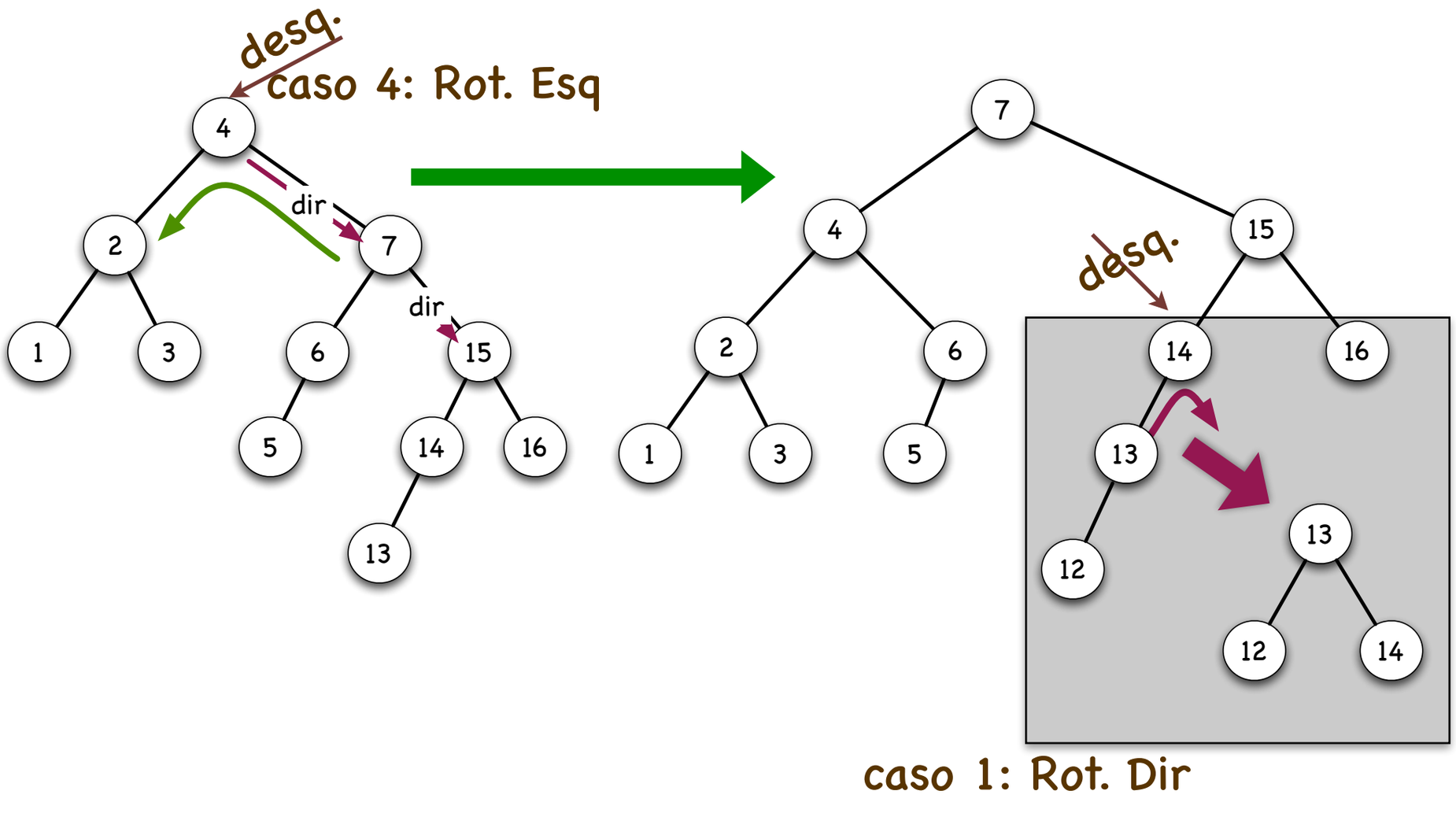
INSERE (16); INSERE(15)



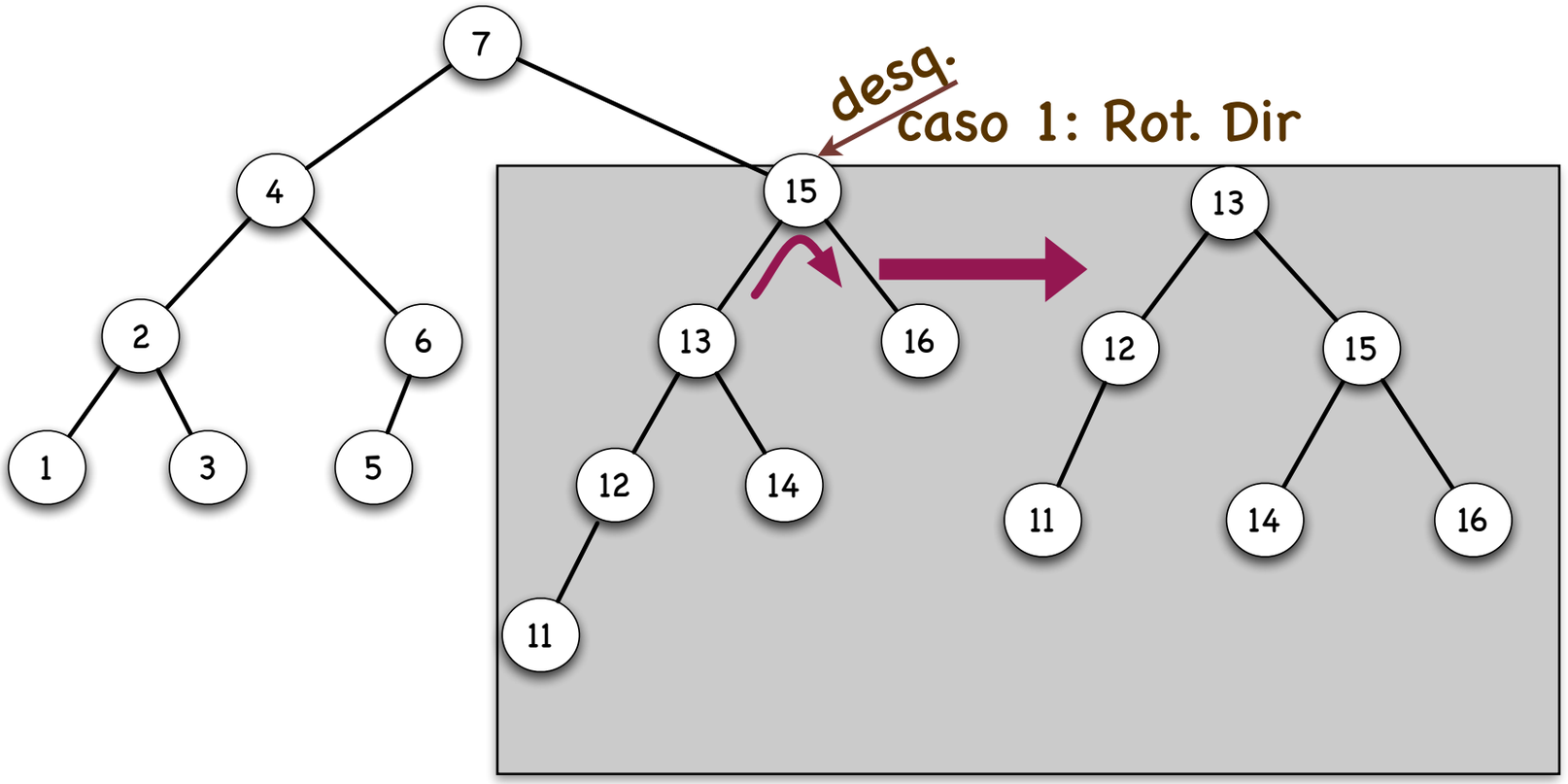
INSERE(14)


caso 3:
Rot. Dupla Dir Esq
desq:
4
2
6
1
3
5
15
7
16
14
6
5
7
15
14
16
4
2
7
1
3
6
15
5
14
16

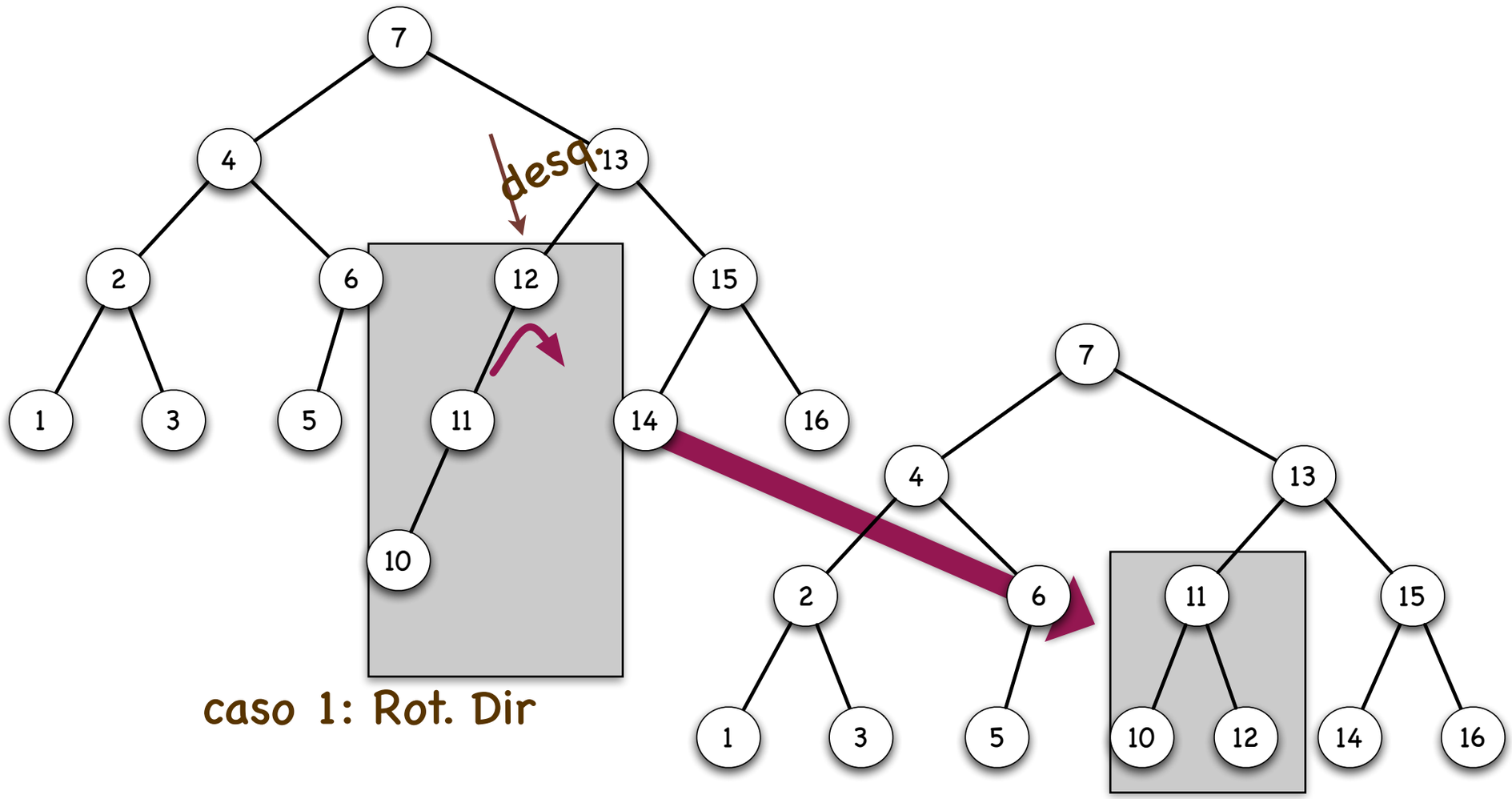
INSERE(13); INSERE(12)



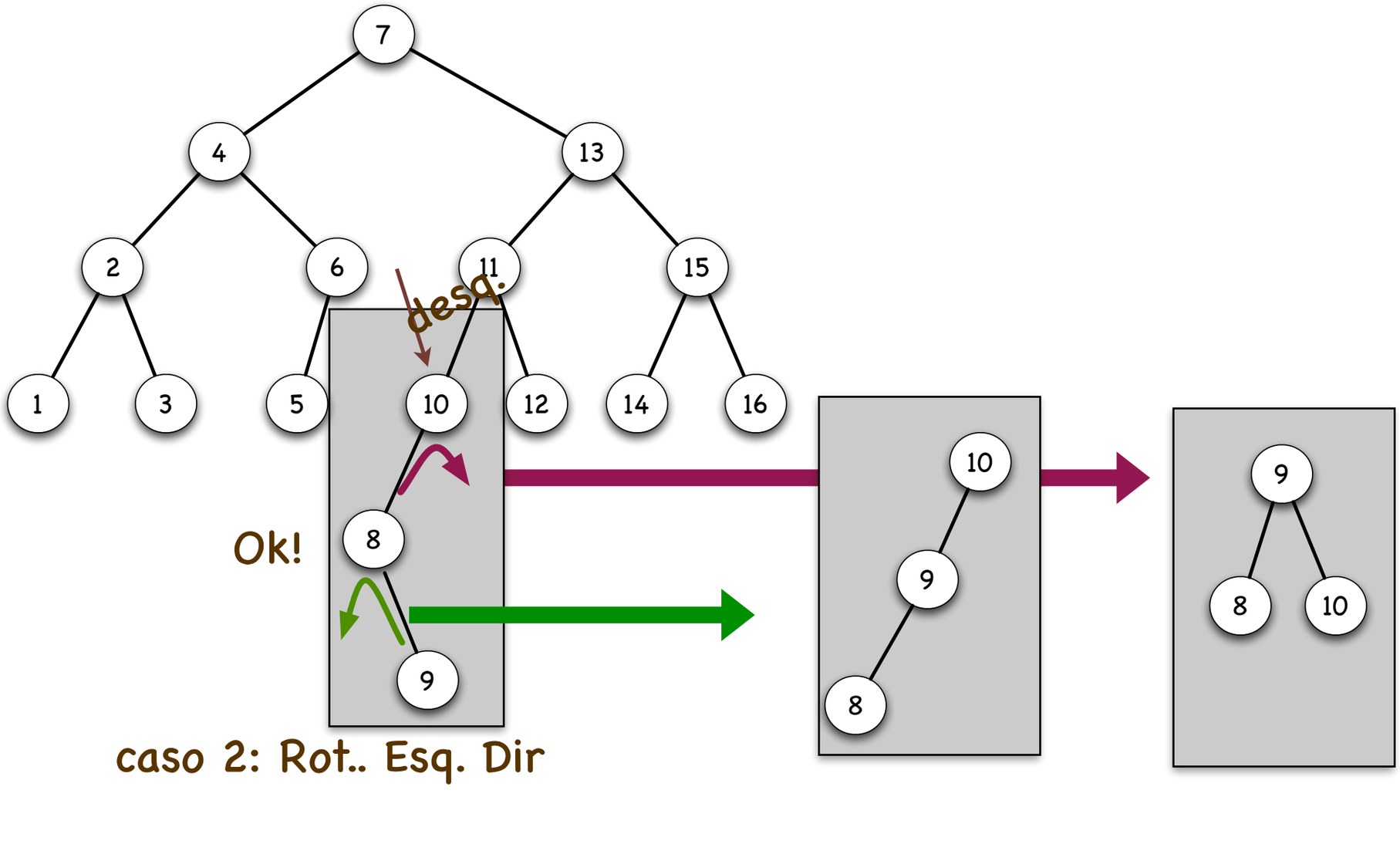
INSERE(11)



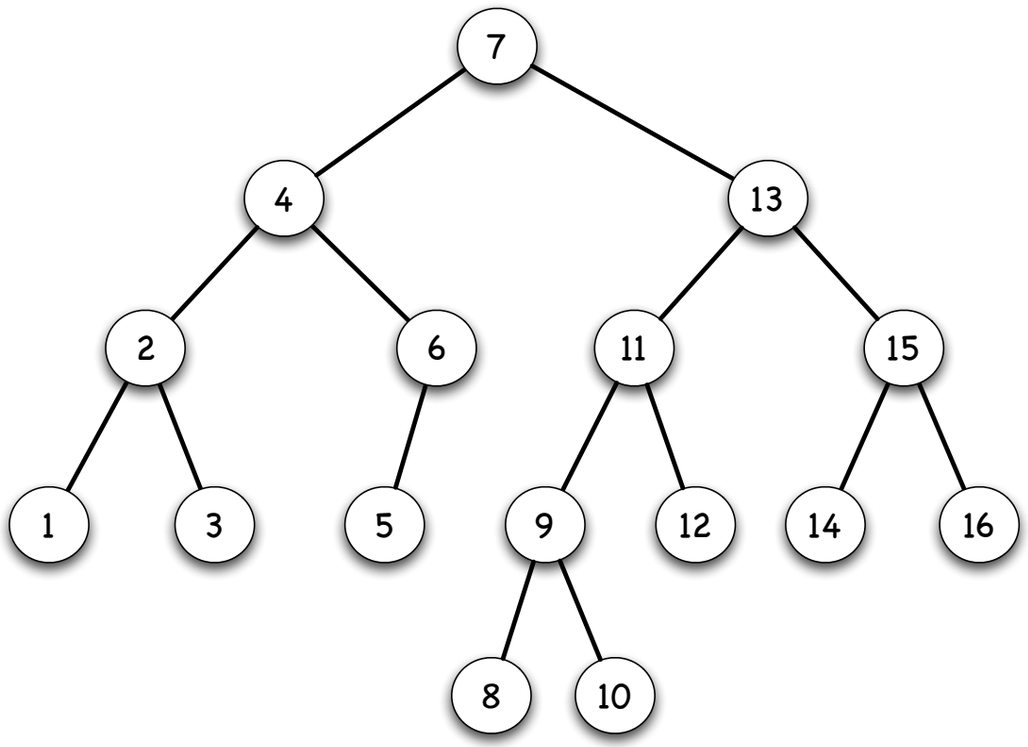
INSERE(10)


7
4
13
2
6
12
15
1
3
5
11
14
16
10
desq.
caso 1: Rot. Dir
7
4
13
2
6
11
15
1
3
5
10
12
14
16

INSERE(8); INSERE(9)



Árvore Final



Exemplo

- Inserir numa AVL vazia os seguintes nós pela ordem indicada:
  - 30,20,7,15,12,8,10,25,28,29
    - insere(30)

      Ok!
    - insere(20)

      Ok!
    - insere(7)

      30 desq, caso 1:Rot. Dir
    - insere(15)

      Ok!
    - insere(12)

      7 desq, caso 3:Dupla Dir/Esq

Exemplo

- Inserir numa AVL vazia os seguintes nós pela ordem indicada:
  - 30,20,7,15,12,8,10,25,28,29
    - insere(8)

      20 desq, caso 1:Rot. Dir
    - insere(10)

      7 desq, caso 4:Rot. Esq
    - insere(25)

      Ok!
    - insere(28)

      30 desq, caso2:Dupla Esq/Dir
    - insere(29)

      20 desq, caso4:Rot. Esq