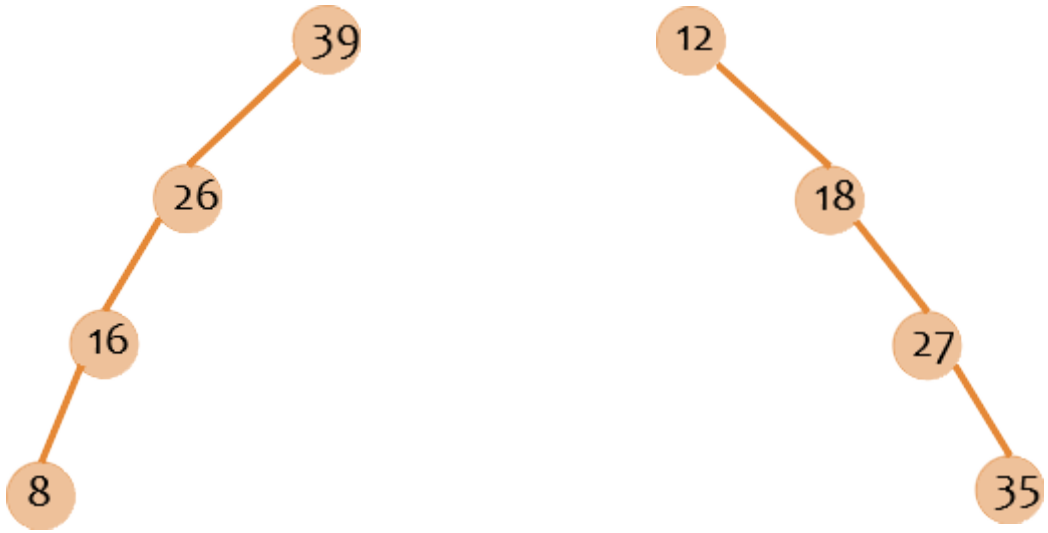
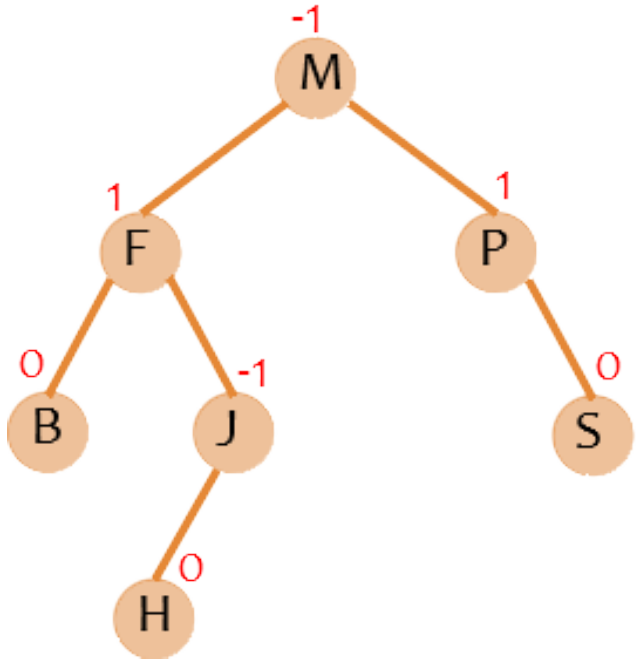
Tema 4
Árboles III: Árboles equilibrados AVL

- **Eficiencia de la búsqueda en un árbol binario**
  - Varía entre **O(n)** y **O(log(n))**, dependiendo de la estructura que presente el árbol
  - La estructura depende del orden de inserción de los datos.
    - Orden creciente: sólo ramas derechas
    - Orden decreciente: sólo ramas izquierdas



- **Árbol equilibrado**
  - Árbol binario en el que las alturas de los dos subárboles para cada nodo nunca difieren en más de una unidad.
  - También se les llama **árboles AVL** en honor a los matemáticos rusos *Adelson-Velskii* y *Landis* (primeros en proponer y desarrollar este TAD).
  - A cada nodo se le añade un nuevo campo que indica la diferencia de alturas entre el subárbol derecho e izquierdo: **FACTOR DE EQUILIBRIO** o **BALANCE**



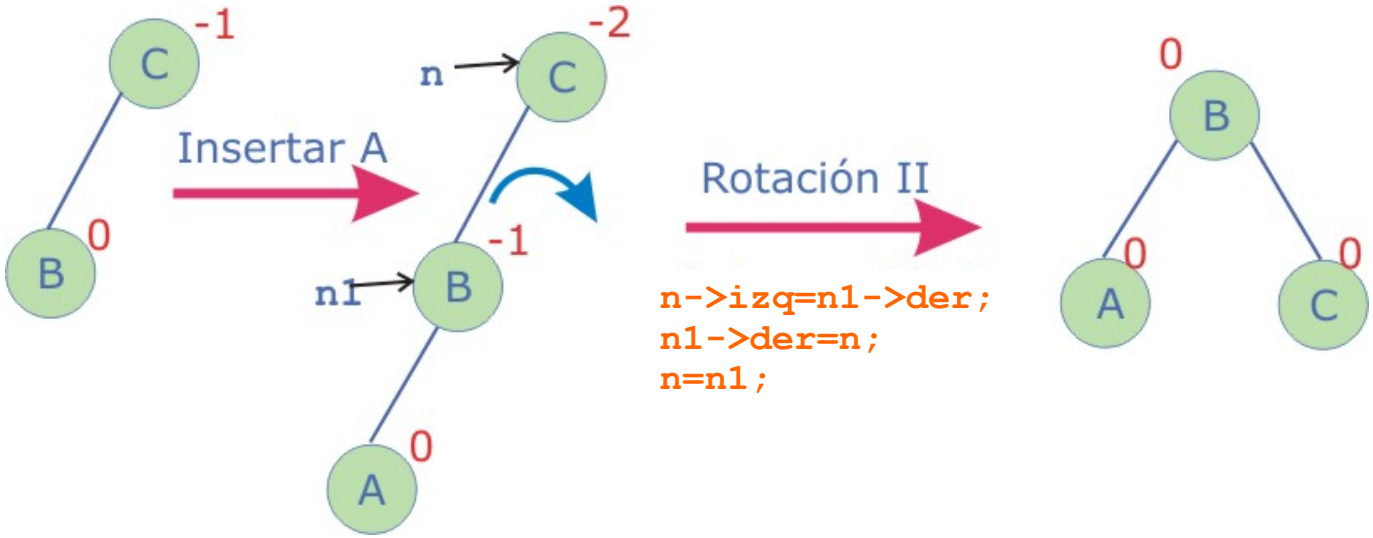

```
struct tipocelda{
     tipoelem info;
     int fe;
     struct tipocelda *izq, *der;
}

typedef tipocelda * avl;
```

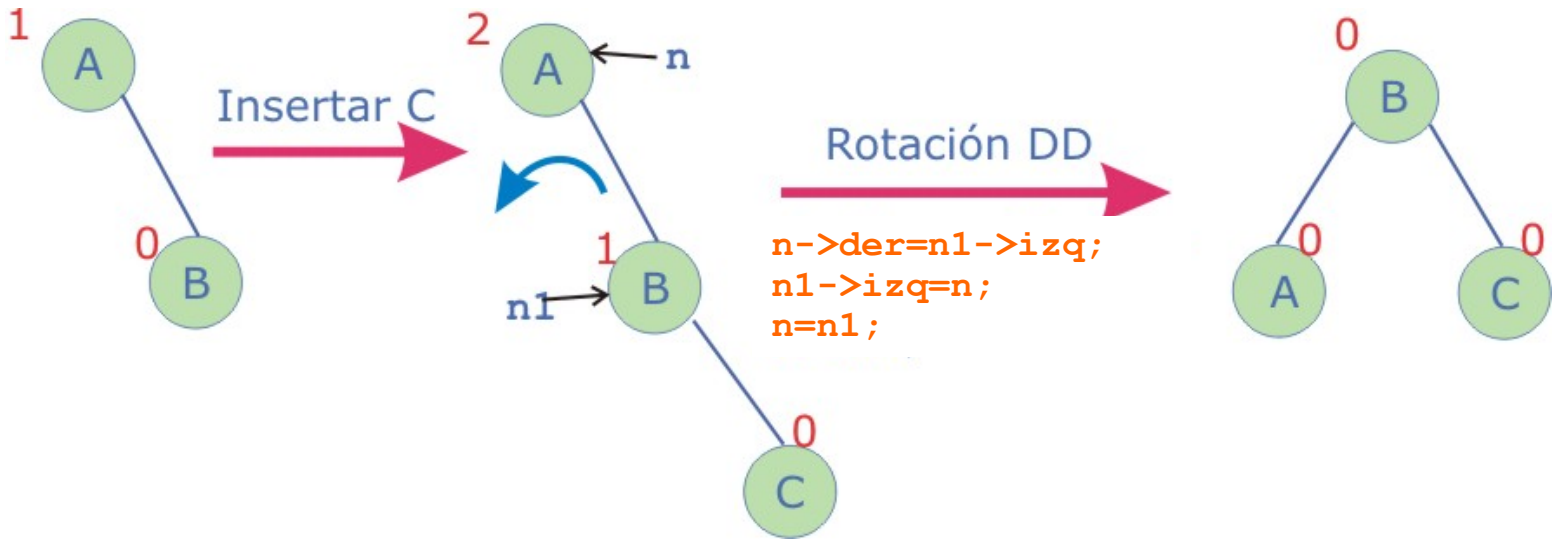
- **Inserción de un nuevo nodo (como nodo hoja)**
  1. Si las ramas izquierda (Ri) y derecha (Rd) del árbol tienen la misma altura ($h_{Ri}=h_{Rd}$), que se inserte en rama izquierda o en rama derecha no va a ser causa de romper el equilibrio.
  2. Si las ramas izquierda y derecha del árbol tienen altura diferente ($|h_{Ri} - h_{Rd}|=1$):
     1. Suponiendo que $h_{Ri} < h_{Rd}$ puede ocurrir:
        1. Si se inserta el nodo en rama izquierda, entonces $h_{Ri}=h_{Rd}$. Se ha mejorado el equilibrio.
        2. Si se inserta el nodo en rama derecha, entonces puede romperse el criterio de equilibrio del árbol y será necesario reestructurarlo.
     2. Suponiendo que $h_{Ri} > h_{Rd}$ puede ocurrir:
        1. Si se inserta el nuevo nodo en rama izquierda, entonces puede quedar roto el equilibrio y habrá que reestructurar el árbol.
        2. Si se inserta el nuevo nodo en rama derecha, entonces $h_{Ri}=h_{Rd}$. Se ha mejorado el equilibrio.

- **Reestructuración del árbol: rotaciones**
  - Insertar nodo en árbol binario de búsqueda:
    - Bajamos por el árbol hasta encontrar su posición, como en un ABB sin equilibrar
    - El nodo se inserta como hoja del árbol, por lo que su factor de equilibrio `fe` será 0.
  - Regresamos por el camino de búsqueda marcado, calculando el `fe` de los distintos nodos que forman el camino. Si en este cálculo un nodo viola el criterio de balanceo, debe reestructurarse el árbol cuya raíz es dicho nodo.
  - El proceso de cálculo del `fe` finaliza cuando alcanzo la raíz del árbol o bien cuando llego a un nodo que no cumple la propiedad de equilibrio y haga necesaria la reestructuración.

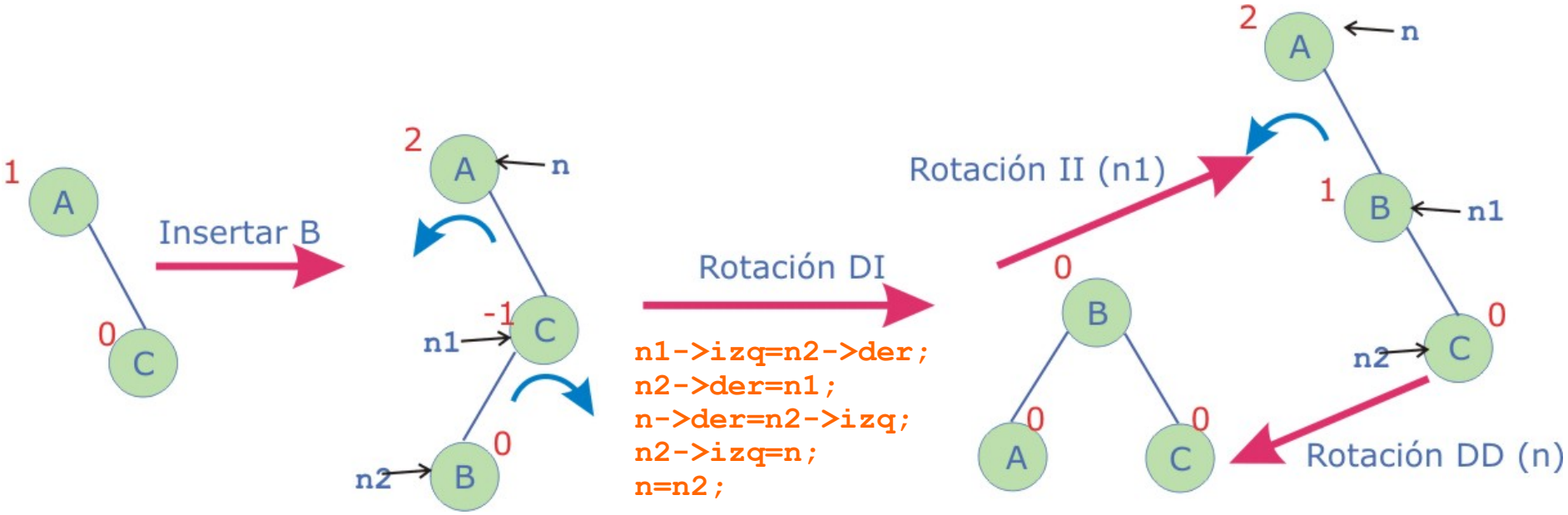
1. Inserción en AVL: Rotaciones

- **Reestructuración del árbol: rotaciones simples (II, DD)**

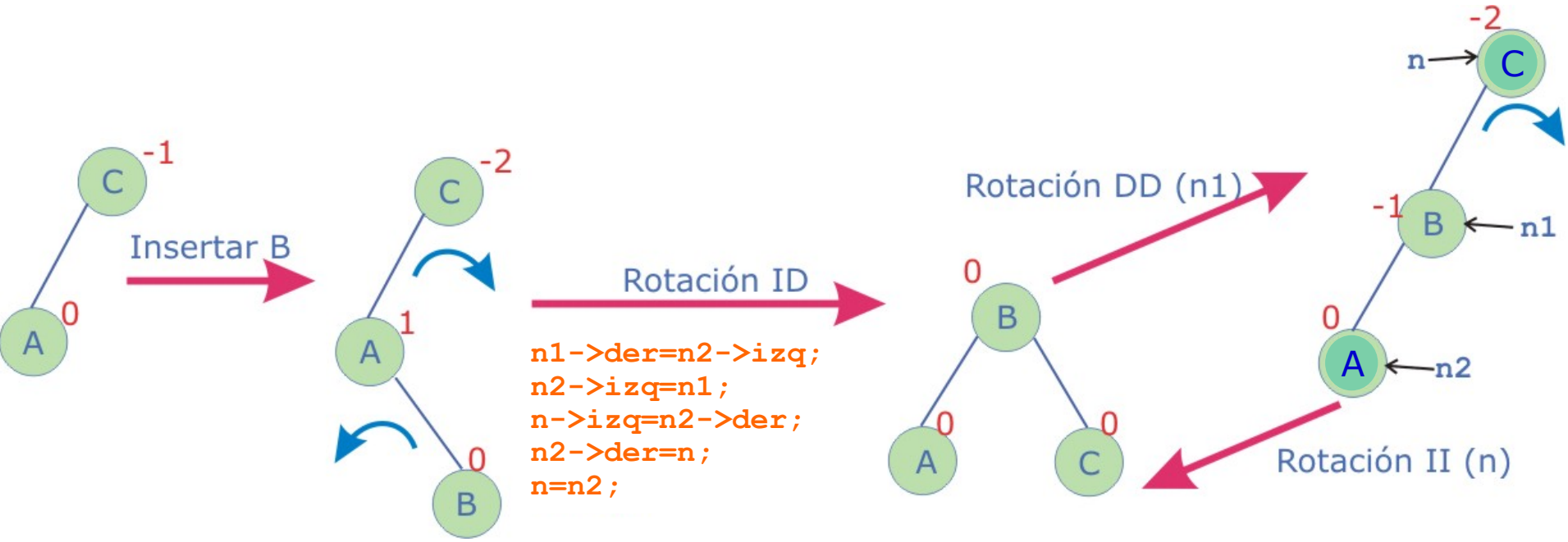




- **Reestructuración del árbol: rotaciones dobles: DI**



- **Reestructuración del árbol: rotaciones dobles: ID**

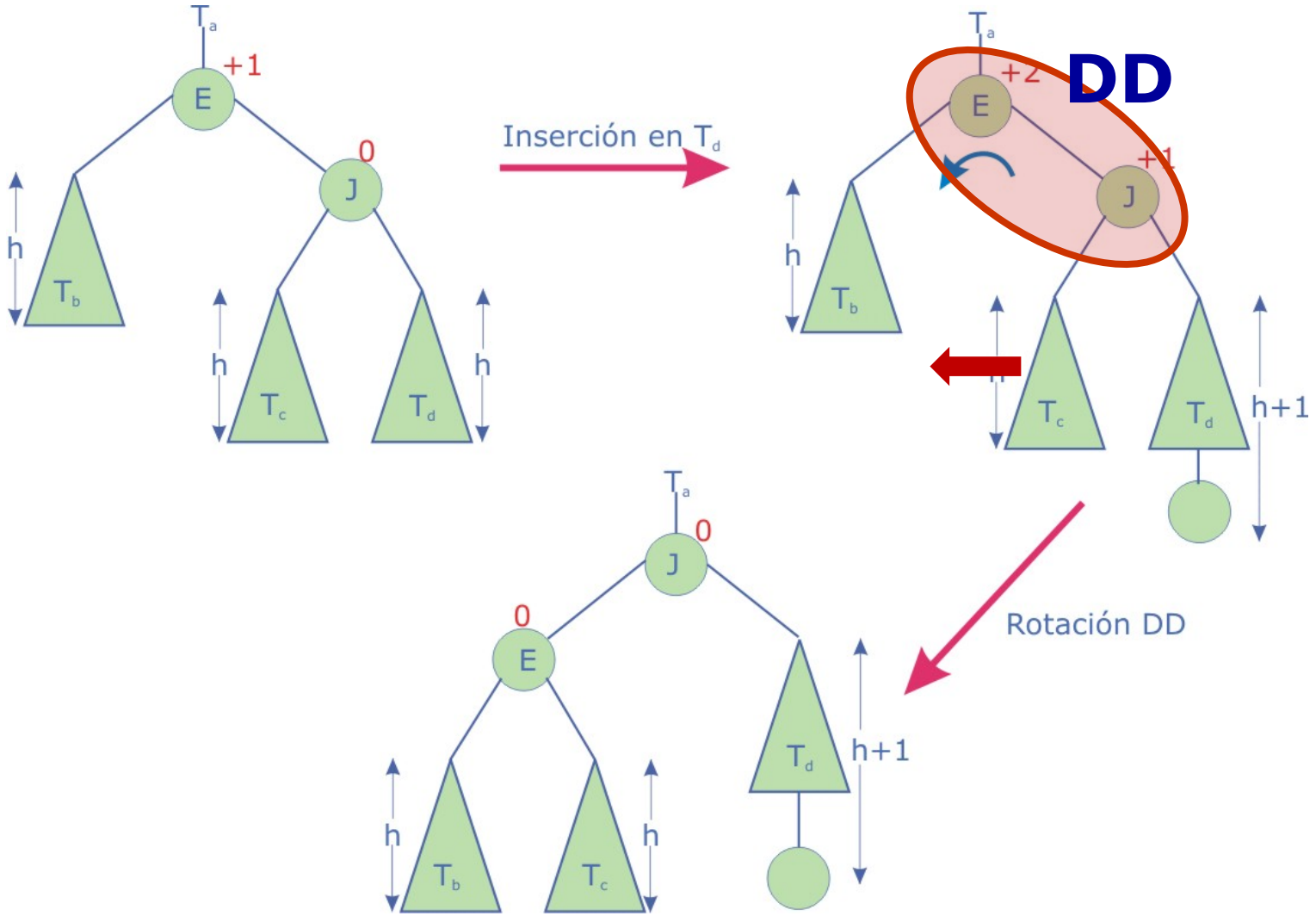


1. Inserción en AVL: Rotaciones

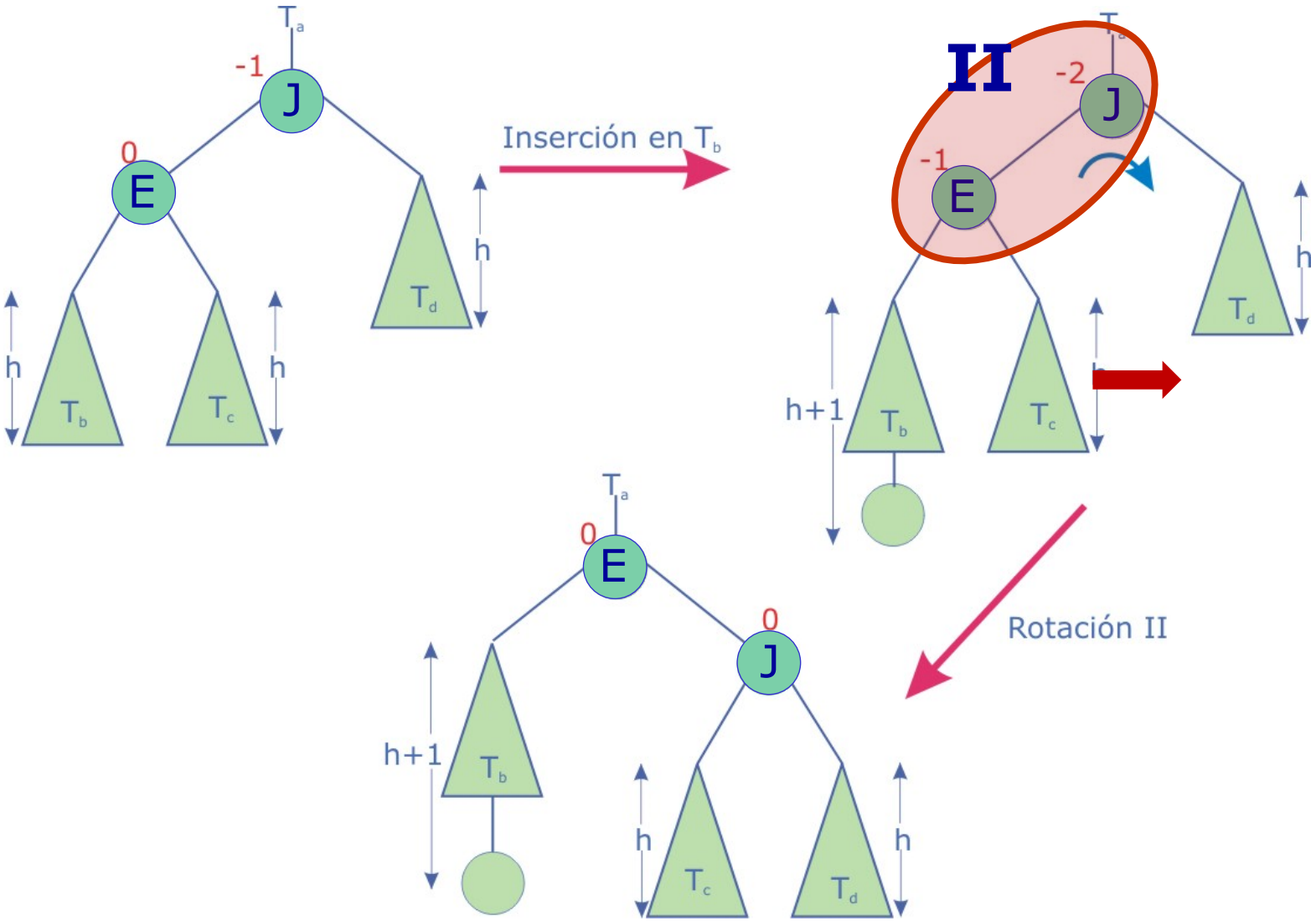
- **REGLAS GENERALES PARA EL EQUILIBRADO DE ÁRBOLES EN LA INSERCIÓN**

n→fe	n1→fe	Rotación
+2	>=0	D D
	<0	D I
-2	<=0	I I
	>0	I D

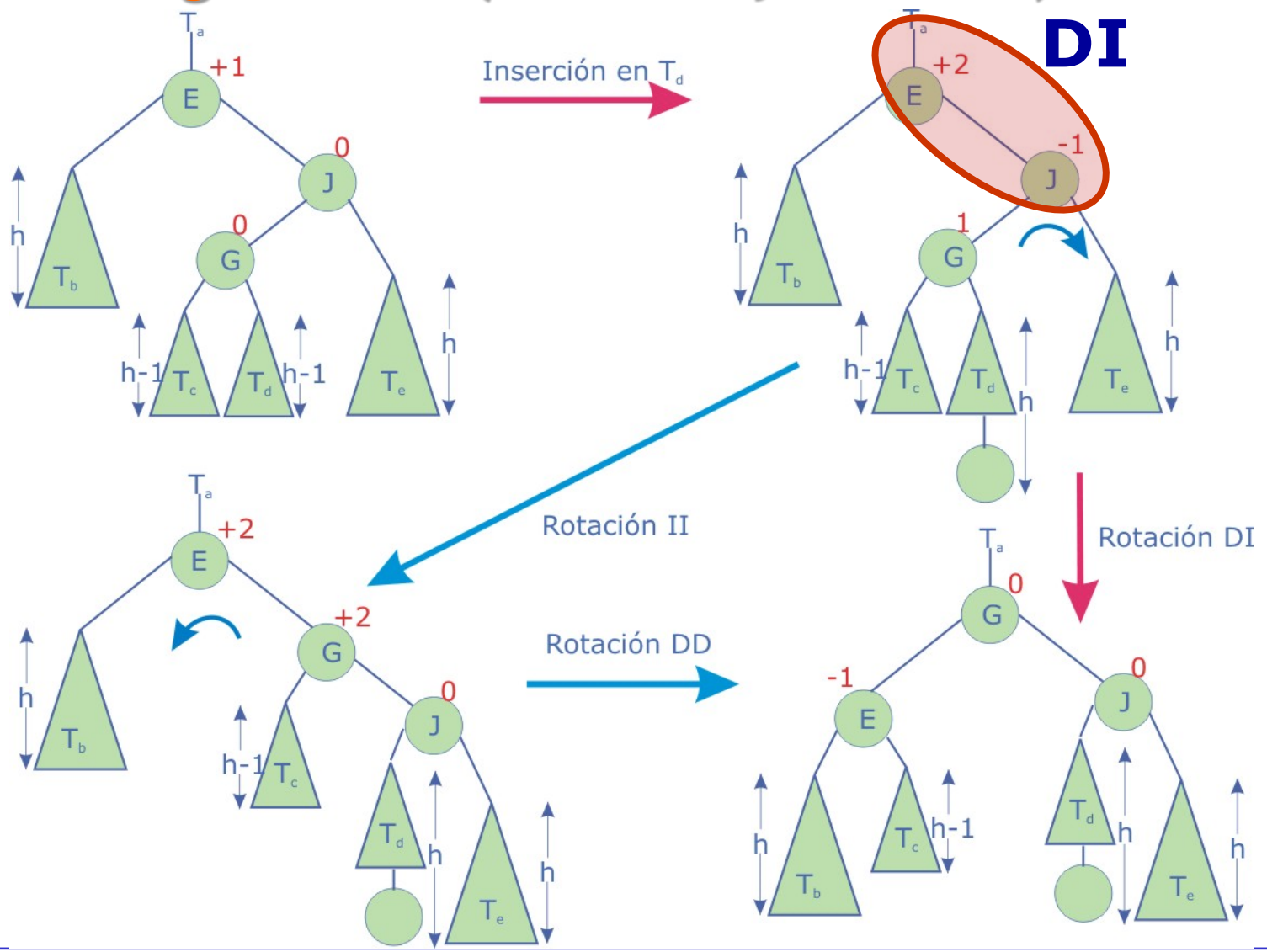
- **Rotación general DD (n→fe=+2 y n1→fe>=0)**



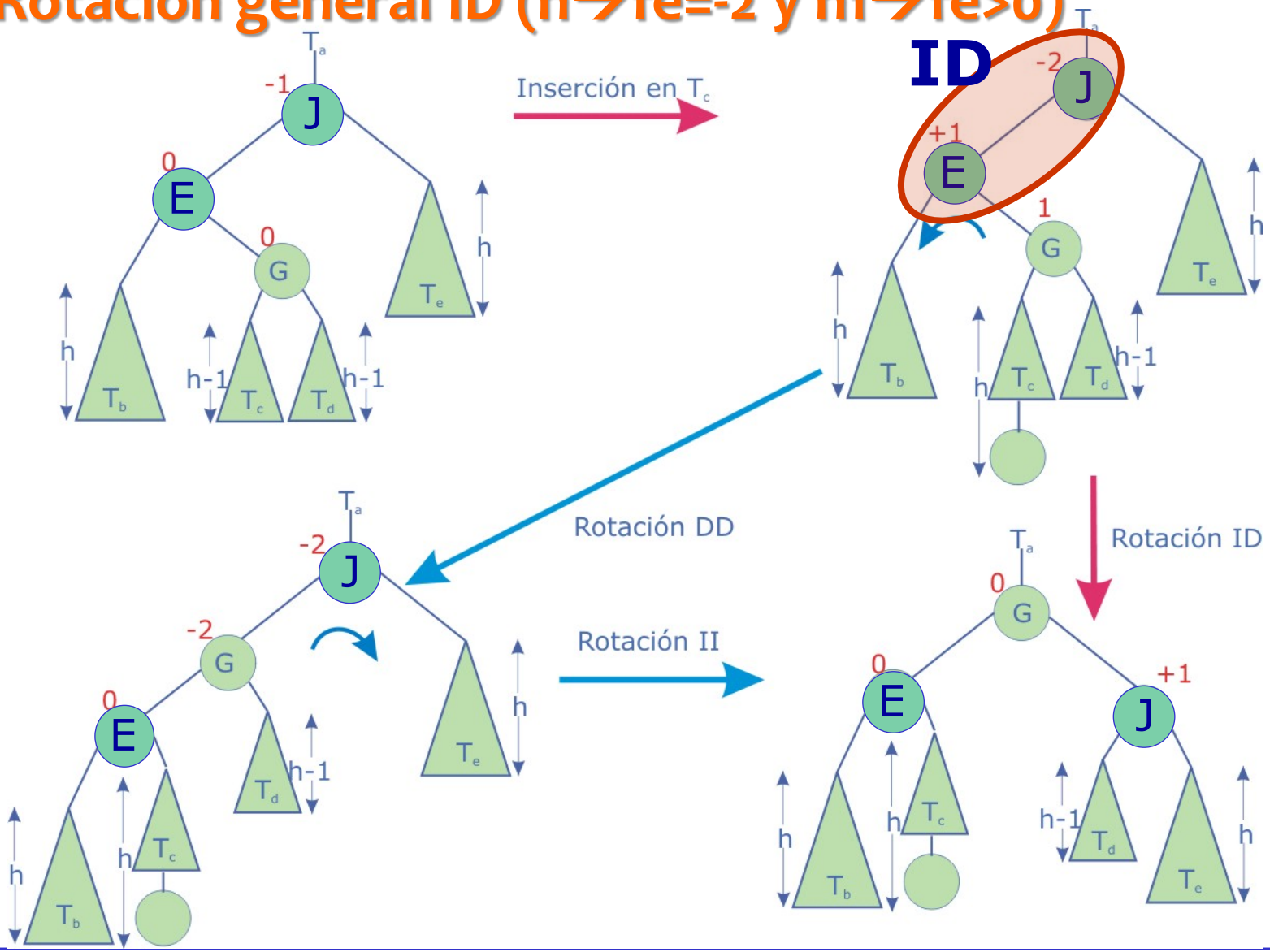
- **Rotación general II (n→fe=-2 y n1→fe<=0)**



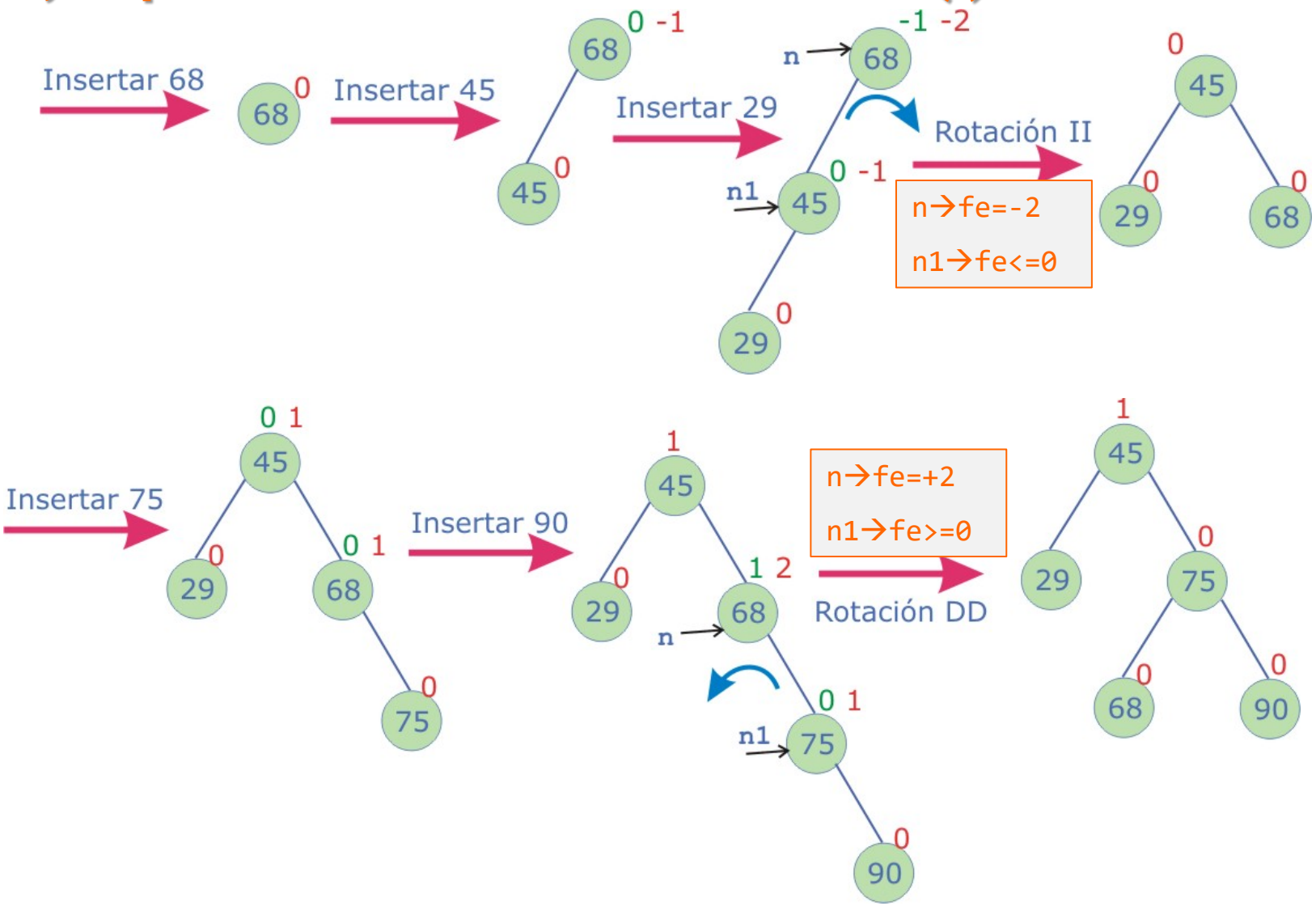
- **Rotación general DI (n→fe=+2 y n1→fe<0)**



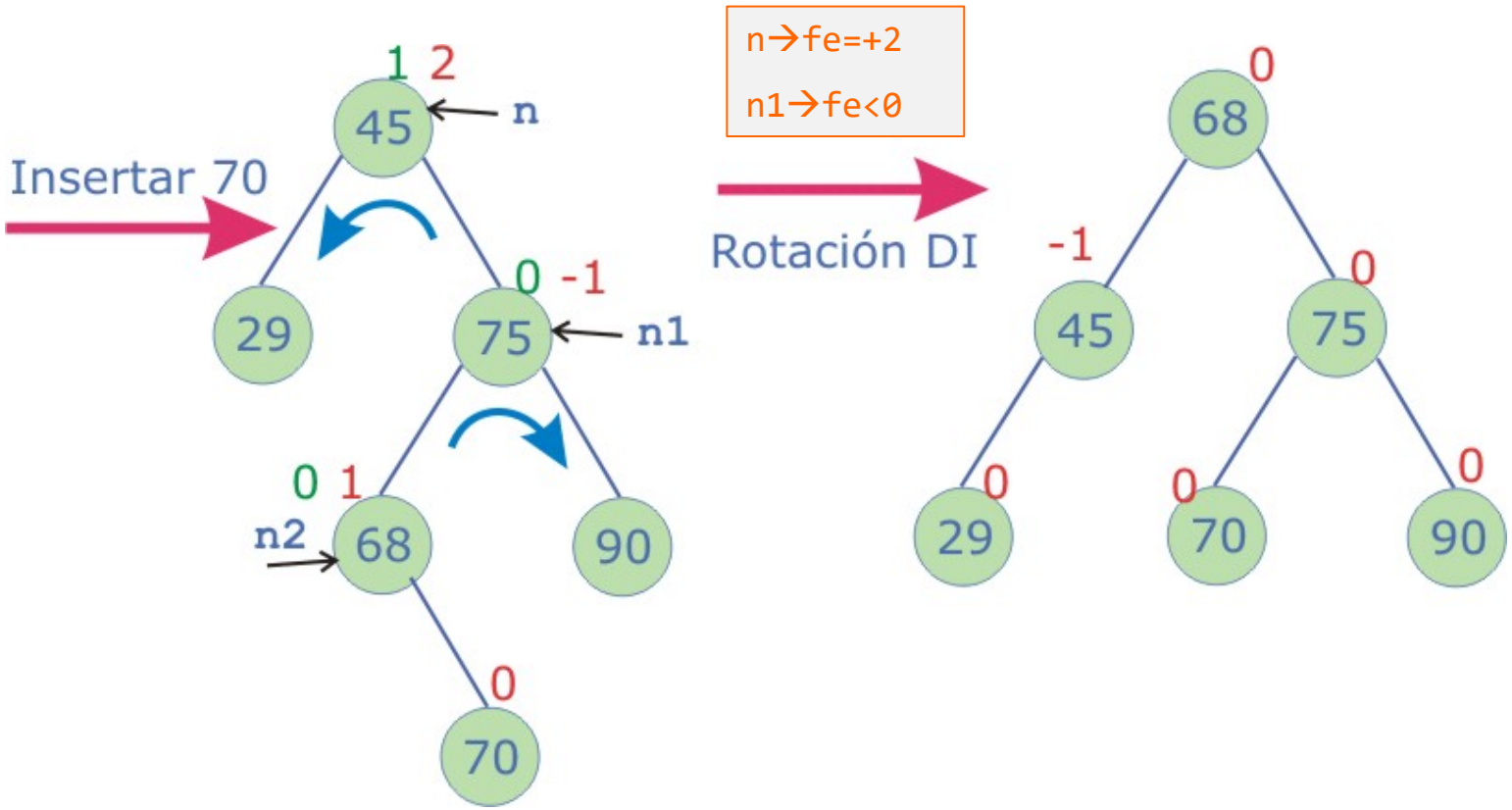
- **Rotación general ID (n→fe=-2 y n1→fe>0)**



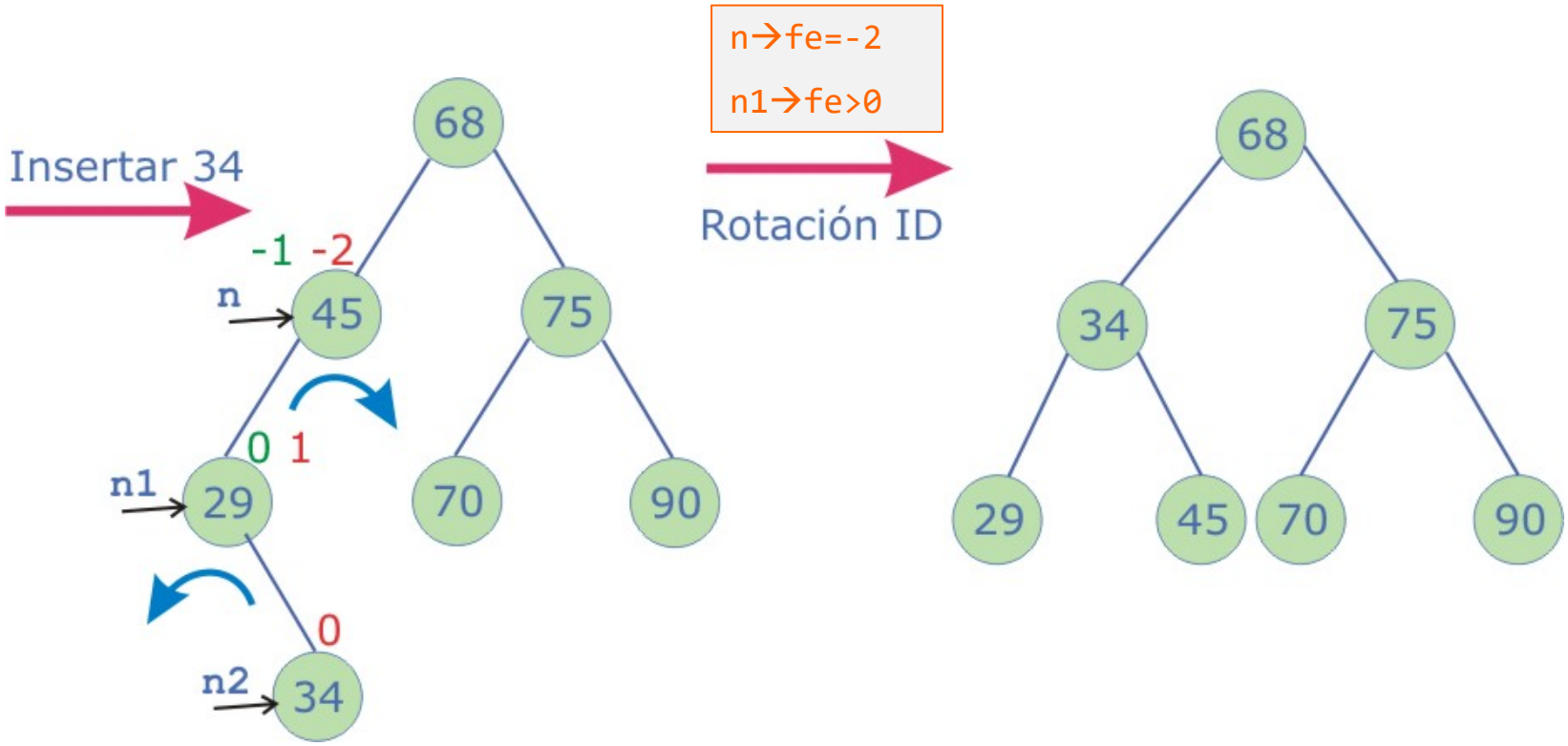
- **Ejemplo de construcción de un AVL (i)**



- **Ejemplo de construcción de un AVL (ii)**



- **Ejemplo de construcción de un AVL (iii)**



2. Eliminación en árboles AVL

- **Mismo algoritmo que en ABB, pero incluyendo operaciones de restauración del equilibrio (rotaciones DD, II, DI, ID)**

- **Casos ABB:**
  1. Si el nodo a suprimir es un nodo hoja, simplemente se elimina.
  2. Si el nodo a suprimir tiene un único descendiente, se sustituye por él.
  3. Si el nodo a suprimir tiene dos subárboles, se busca el nodo más a la izquierda del subárbol derecho, o el más a la derecha del subárbol izquierdo, y se sustituye.
- Se regresa por el camino de búsqueda calculando los nuevos **`fe`** de los nodos visitados. Si en algún nodo se rompe el equilibrio, se realizan rotaciones para equilibrarlo. Hemos de seguir calculando **`fe`** y reequilibrando hasta llegar a la raíz del árbol.

2. Eliminación en árboles AVL

- Dado un nodo, si la diferencia entre las alturas de sus subárboles es igual a 2, hay que determinar cuál de los dos subárboles tiene mayor altura (eso nos va a indicar qué rotación corresponde aplicar: izquierda o derecha), y dentro de ese subárbol determinar nuevamente cuál de sus hijos tiene más altura (eso nos va a indicar si la segunda rotación será izquierda o derecha). Visto en pseudocódigo sería:

```
SI (fe= -2) ENTONCES
   SI (fe(hijo izquierdo)<=0) ENTONCES
      HACER Rotación Izquierda-Izquierda
   SINO
      HACER Rotación Izquierda-Derecha
   FIN_SI
FIN_SI
```

fe=-2 =>Miro subárbol izquierdo

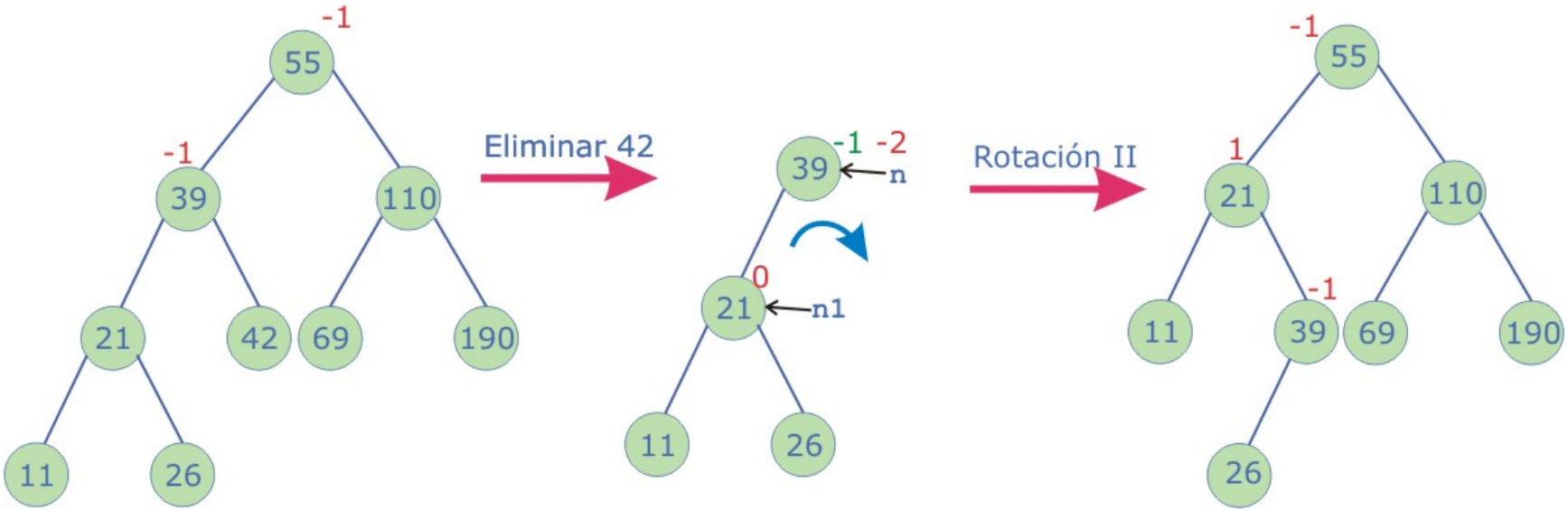
```
SI (fe = 2) ENTONCES
   SI (fe(hijo derecho)>=0 ) ENTONCES
      HACER Rotación Derecha-Derecha
   SINO
      HACER Rotación Derecha-Izquierda
   FIN_SI
FIN_SI
```

fe=2 =>Miro subárbol derecho

- **REGLAS GENERALES PARA EL EQUILIBRADO DE ÁRBOLES EN LA ELIMINACIÓN**

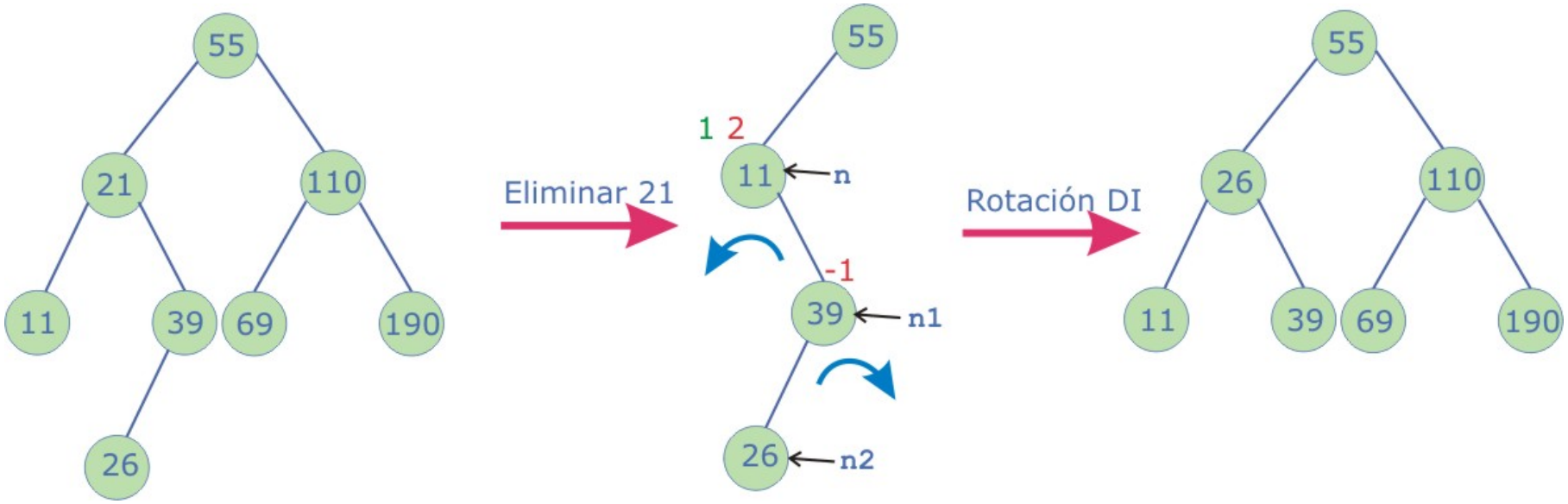
n→fe	Hijo a examinar	hijo→fe	Rotación
+2	Hijo derecho	fe(HD)>=0	D D
		fe(HD)<0	D I
-2	Hijo izquierdo	fe(HI)<=0	I I
		fe(HI)>0	I D

- **Ejemplo 1 de eliminación (i)**



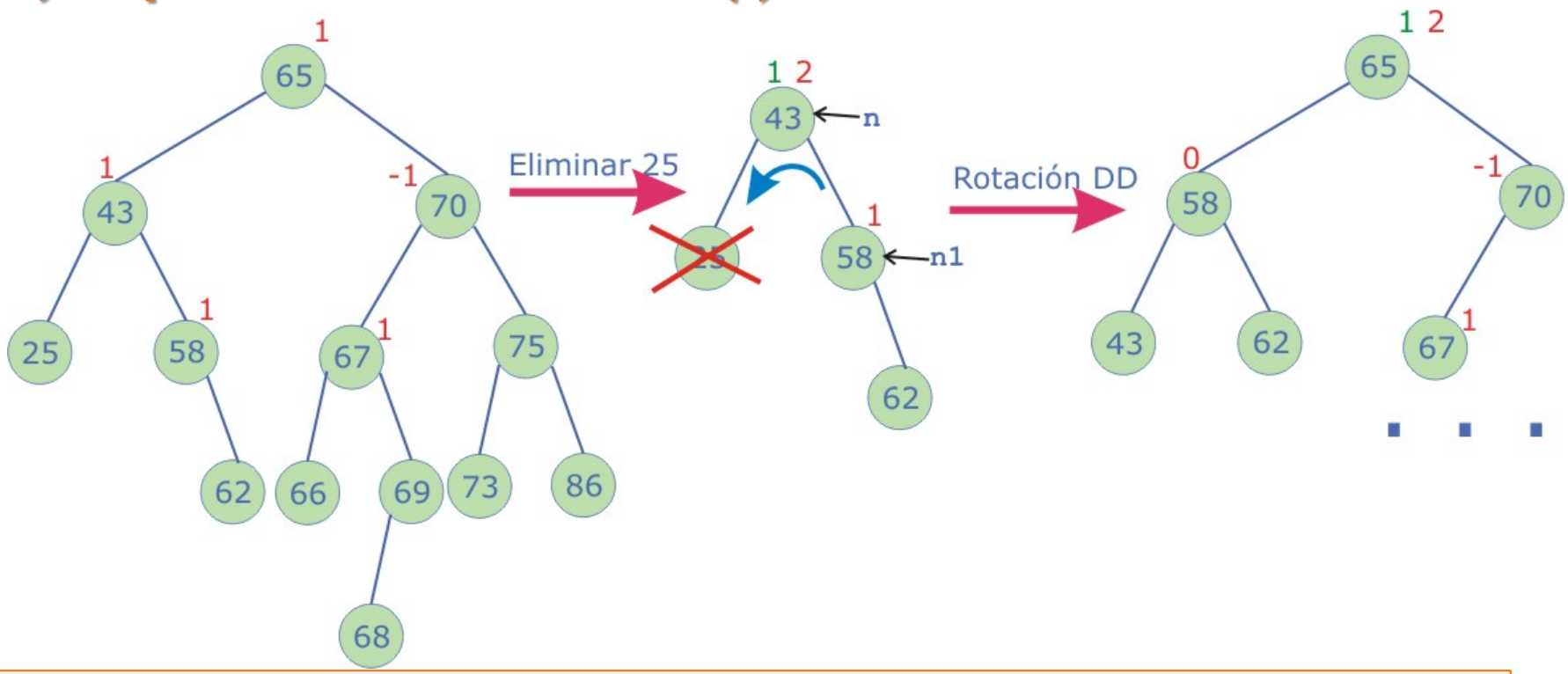
- 42 es una hoja, simplemente se elimina
- Vuelvo por el camino hasta la raíz viendo si hay desequilibrios:
  - fe(39)=-2 =>Miro subárbol izquierdo
  - fe(HI)=fe(21)<=0 => Rotación II
- SIGO HASTA LA RAÍZ, PARA COMPROBAR SI HAY MÁS DESEQUILIBRIOS

- **Ejemplo 1 de eliminación (ii)**



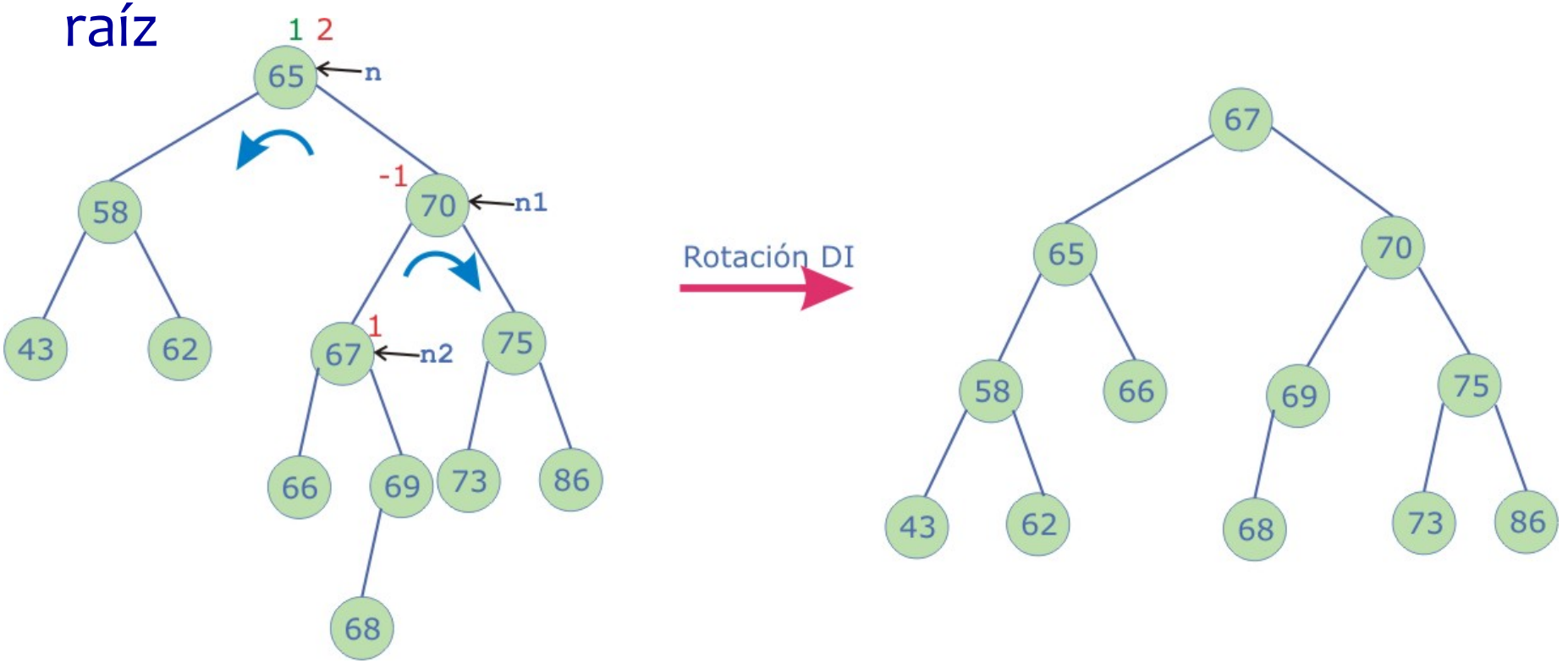
- 21 tiene dos hijos: lo sustituimos por el mayor del subárbol izquierdo, que es 11
- Vuelvo por el camino hasta la raíz viendo si hay desequilibrios:
  - fe(11)=+2 =>Miro subárbol derecho
  - fe(HD)=fe(39)<0 => Rotación DI
- SIGO HASTA LA RAÍZ, PARA COMPROBAR SI HAY MÁS DESEQUILIBRIOS

- **Ejemplo 2 de eliminación (i)**



- 25 es una hoja, simplemente se elimina
- Vuelvo por el camino hasta la raíz viendo si hay desequilibrios:
  - fe(43)=+2 =>Miro subárbol derecho
  - fe(HD)=fe(58)>=0 => Rotación DD
- SIGO HASTA LA RAÍZ, PARA COMPROBAR SI HAY MÁS DESEQUILIBRIOS

- **Ejemplo 2 de eliminación (ii):** sigo calculando fe hasta la raíz



- SIGO HASTA LA RAÍZ, PARA COMPROBAR SI HAY MÁS DESEQUILIBRIOS
- Vuelvo por el camino hasta la raíz viendo si hay desequilibrios:
  - fe(65)=+2 =>Miro subárbol derecho
  - fe(HD)=fe(70)<0 => Rotación DI