

Árboles AVL

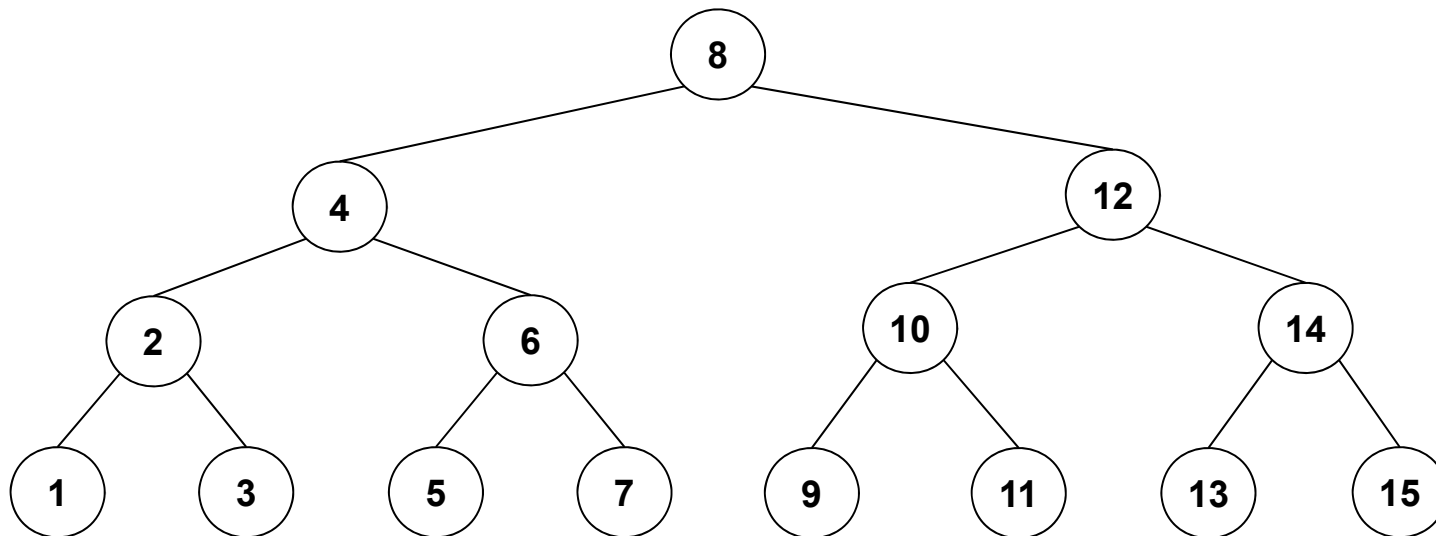
Un caso extremo

Insertar en un Árbol Binario la siguiente serie de valores:
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

O bien:

15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1

¿ $O(n \cdot \log_2 n)$ ó $O(n)$?



Adelson-Velskii and Landis' Tree

Los Árboles AVL son árboles binario de búsqueda auto-balanceados-
En un árbol balanceado la altura de dos subárboles hermanos en cualquier nodo del árbol son iguales o difieren por no más de un nivel de altura.

Si en algún momento la diferencia de alturas entre subárboles hermanos difieren por más de uno, se realiza un rebalanceo para mantener el esto balanceado del árbol.

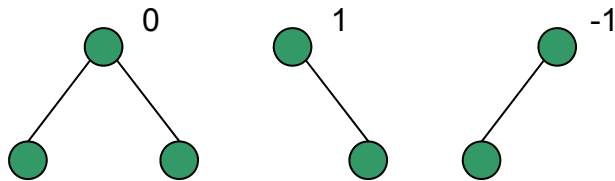
La diferencia de alturas entre subárboles hermanos dan pie al término *factor de equilibrio*

El rebalanceo se realiza por medio de la *Rotación* de la cual existen

- *Rotación Simple a la Izquierda (RSI) – RII: Rotación Izquierda Izquierda*
- *Rotación Simple a la Derecha (RSD) – RDD: Rotación Derecha Derecha*
- *Rotación Doble a la Izquierda (RDI) – RDI: Rotación Derecha Izquierda*
- *Rotación Doble a la Derecha (RDD) – RID: Rotación Izquierda Derecha*

Árboles AVL

Factor de equilibrio

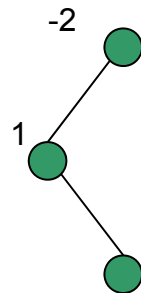
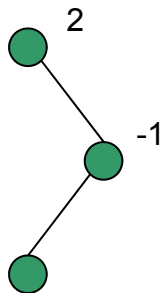
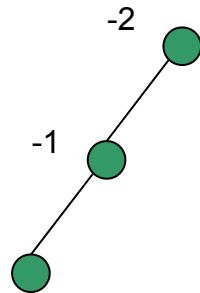
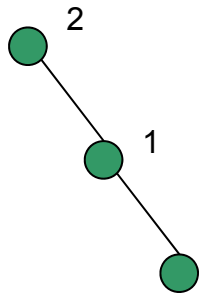


$$FE = ASD - ASI$$

FE: Factor de Equilibrio

ASD: Altura del Subárbol Derecho

ASI: Altura del Subárbol Izquierdo



Factor	
2	Desequilibrio hacia la derecha
1	Equilibrado
0	
-1	
-2	Desequilibrio hacia la izquierda

Árboles AVL

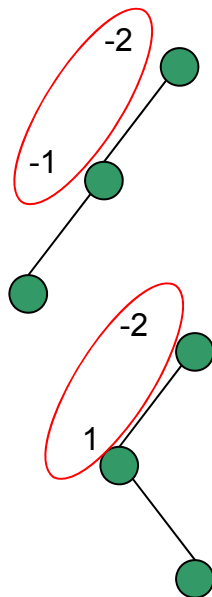
¿Qué rotación aplicar?

Si un árbol está desbalanceado hacia la izquierda (-2), aplicar rotación hacia derecha

Si un árbol está desbalanceado hacia la derecha (2), aplicar rotación hacia izquierda

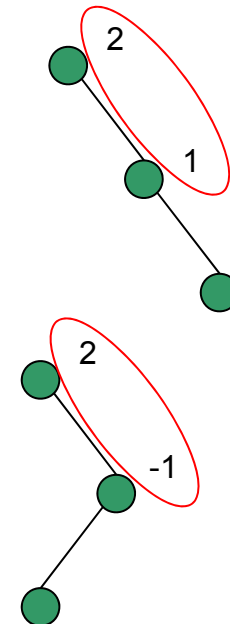
En un árbol que está desbalanceado hacia la izquierda revisar el factor de equilibrio del subárbol izquierdo

En un árbol que está desbalanceado hacia la derecha revisar el factor de equilibrio del subárbol derecho



Si el signo de los factores coincide, se aplica una rotación simple

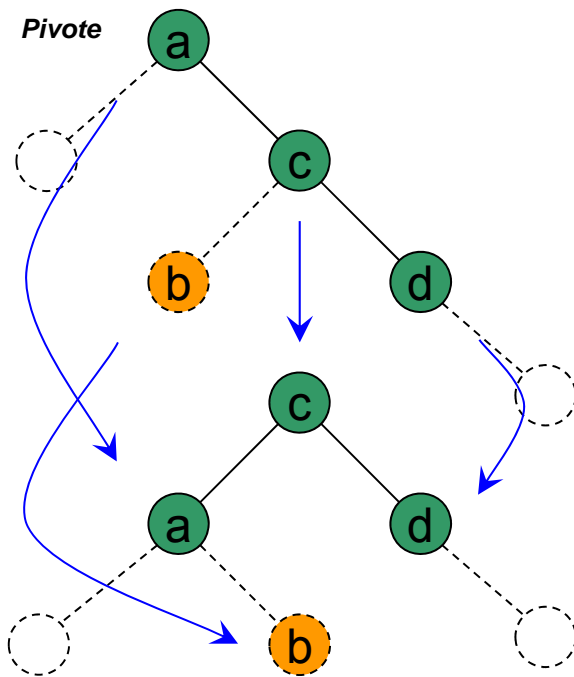
Si el signo de los factores no coincide, se aplica una rotación doble



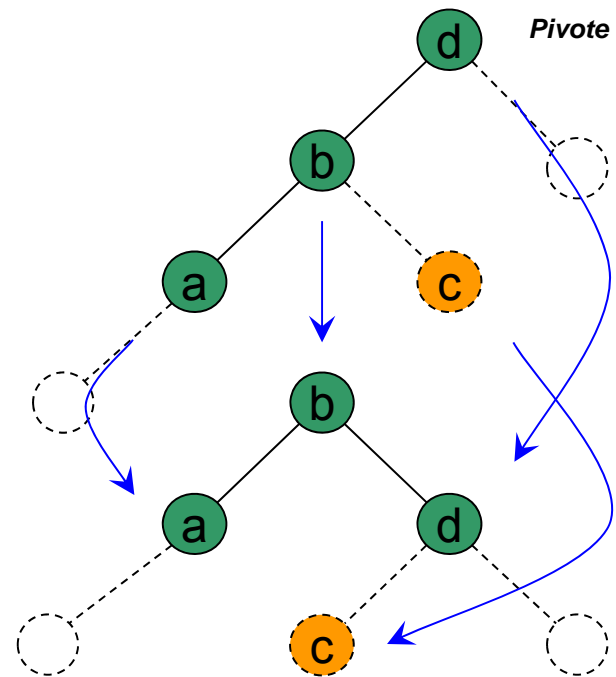
Árboles AVL

Rotaciones Simples

A la izquierda
(RSI)



A la derecha
(RSD)

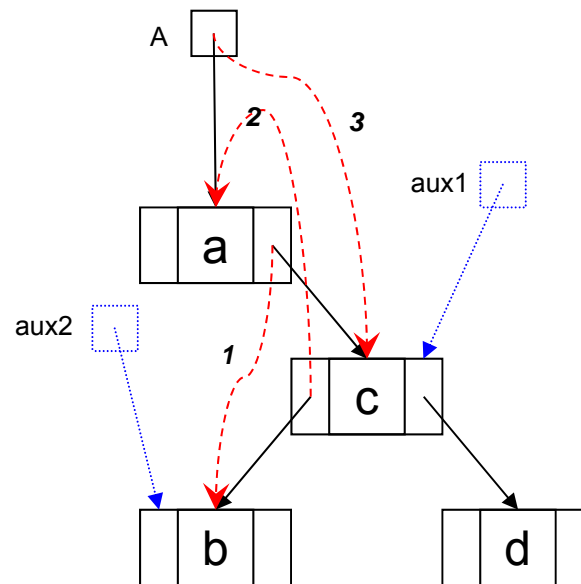
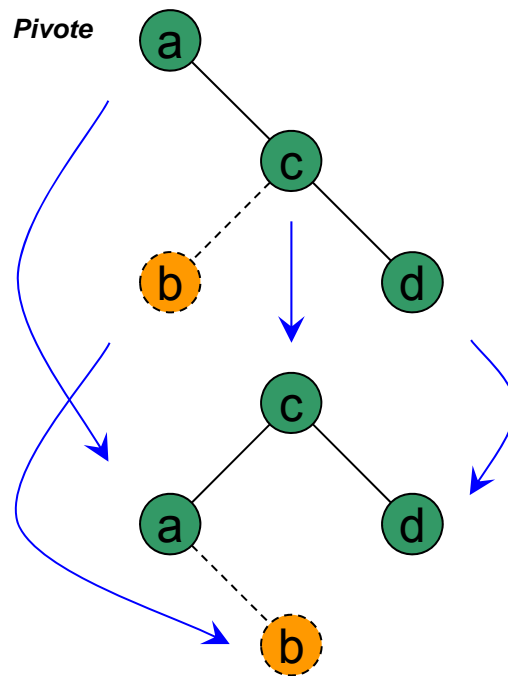


  Árboles opcionales, pueden o no existir

Árboles AVL

Rotaciones Simples

A la izquierda
(RSI)

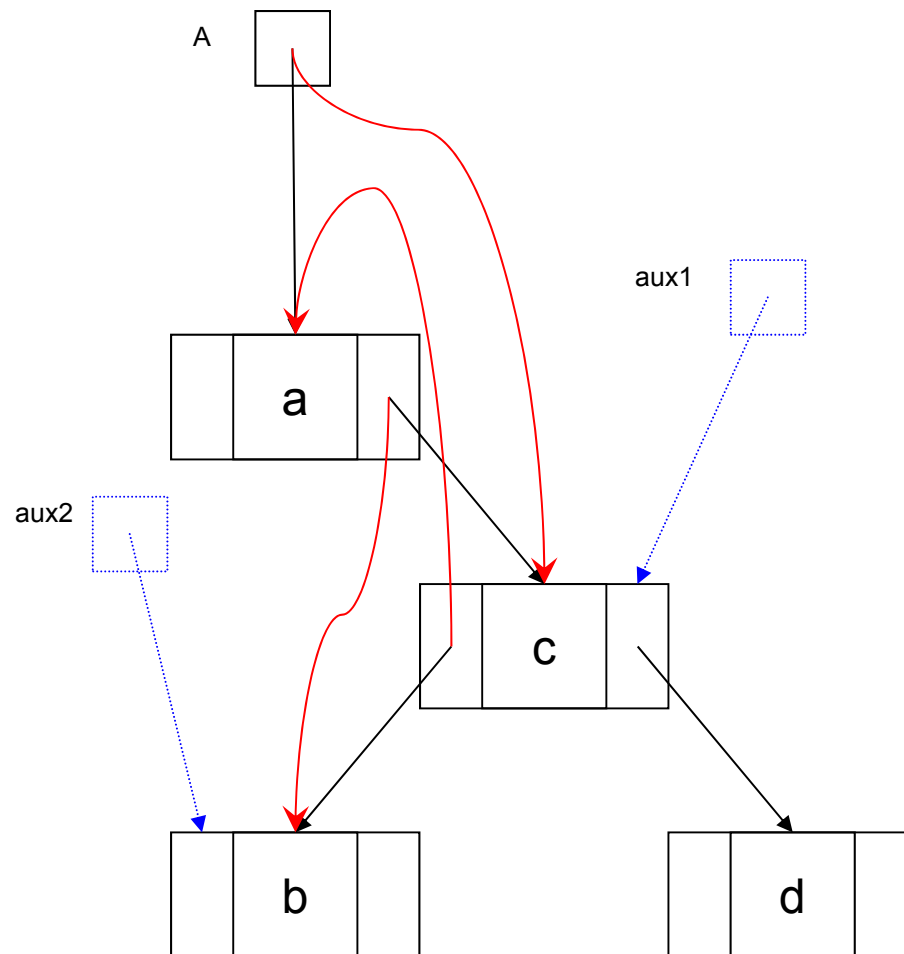


```
aux1 = (*A)->der;
aux2 = aux1->izq;
```

```
(*A)->der = aux2;
aux1->izq = (*A);
(*A) = aux1;
```

Árboles AVL

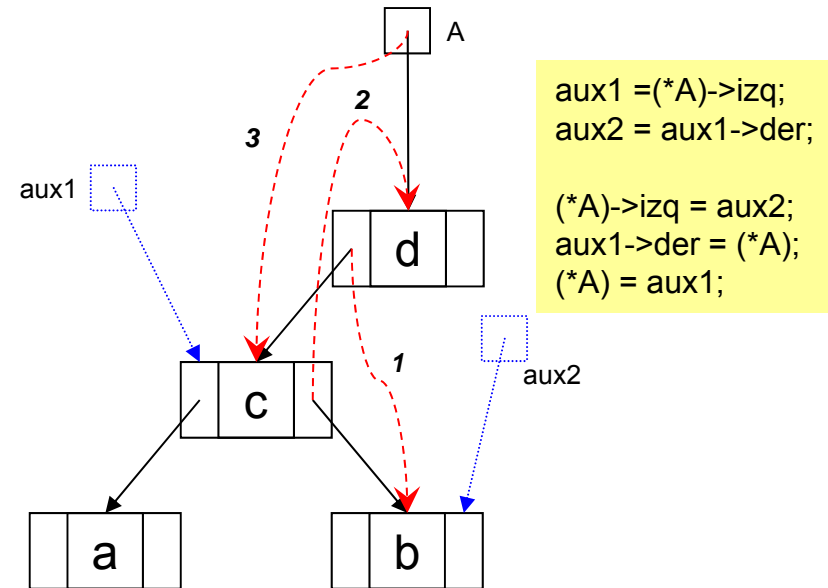
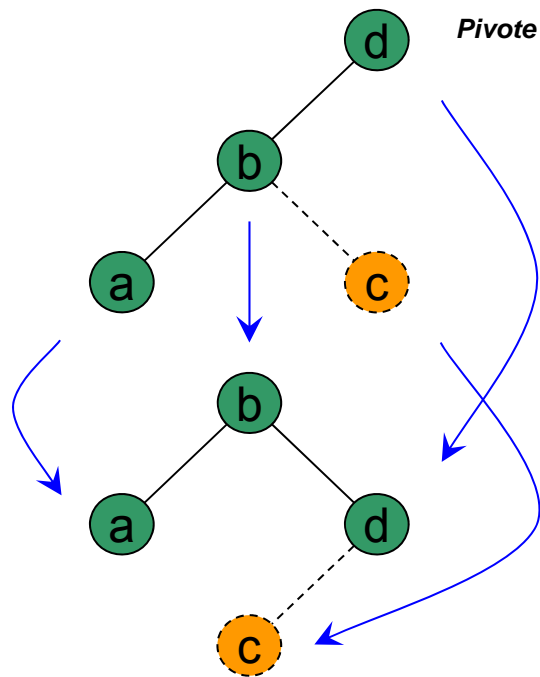
Rotaciones Simples a la Izquierda



Árboles AVL

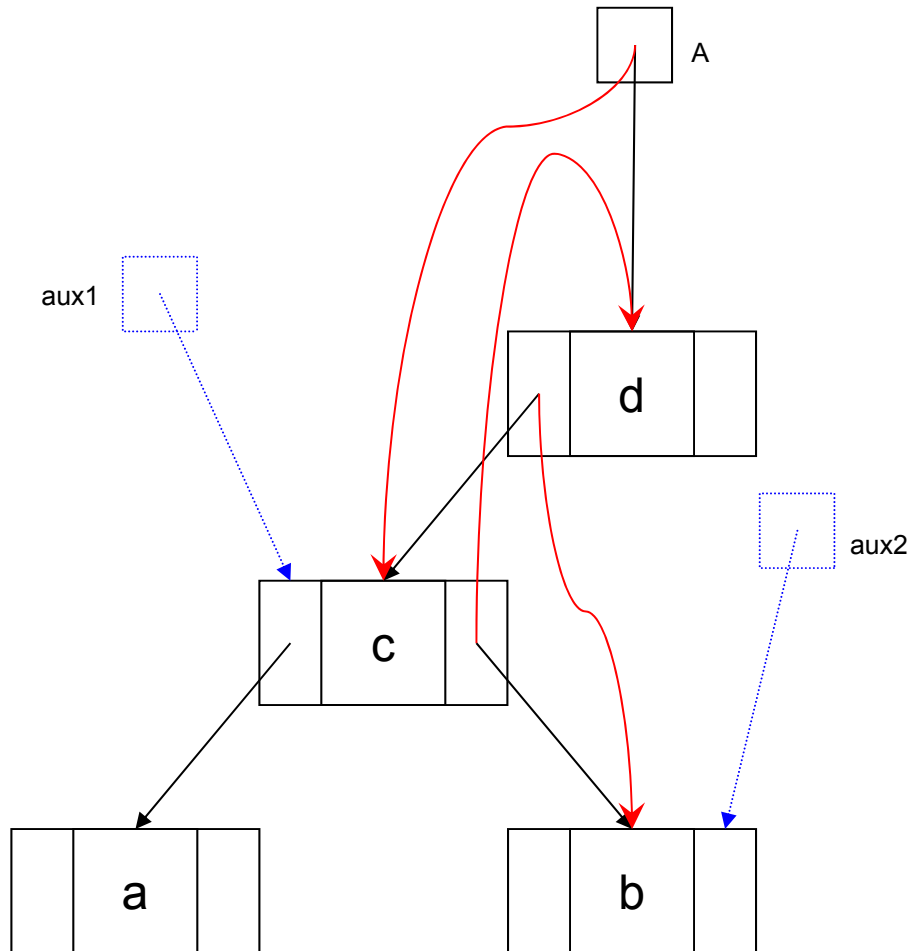
Rotaciones Simples

A la derecha
(RSD)



Árboles AVL

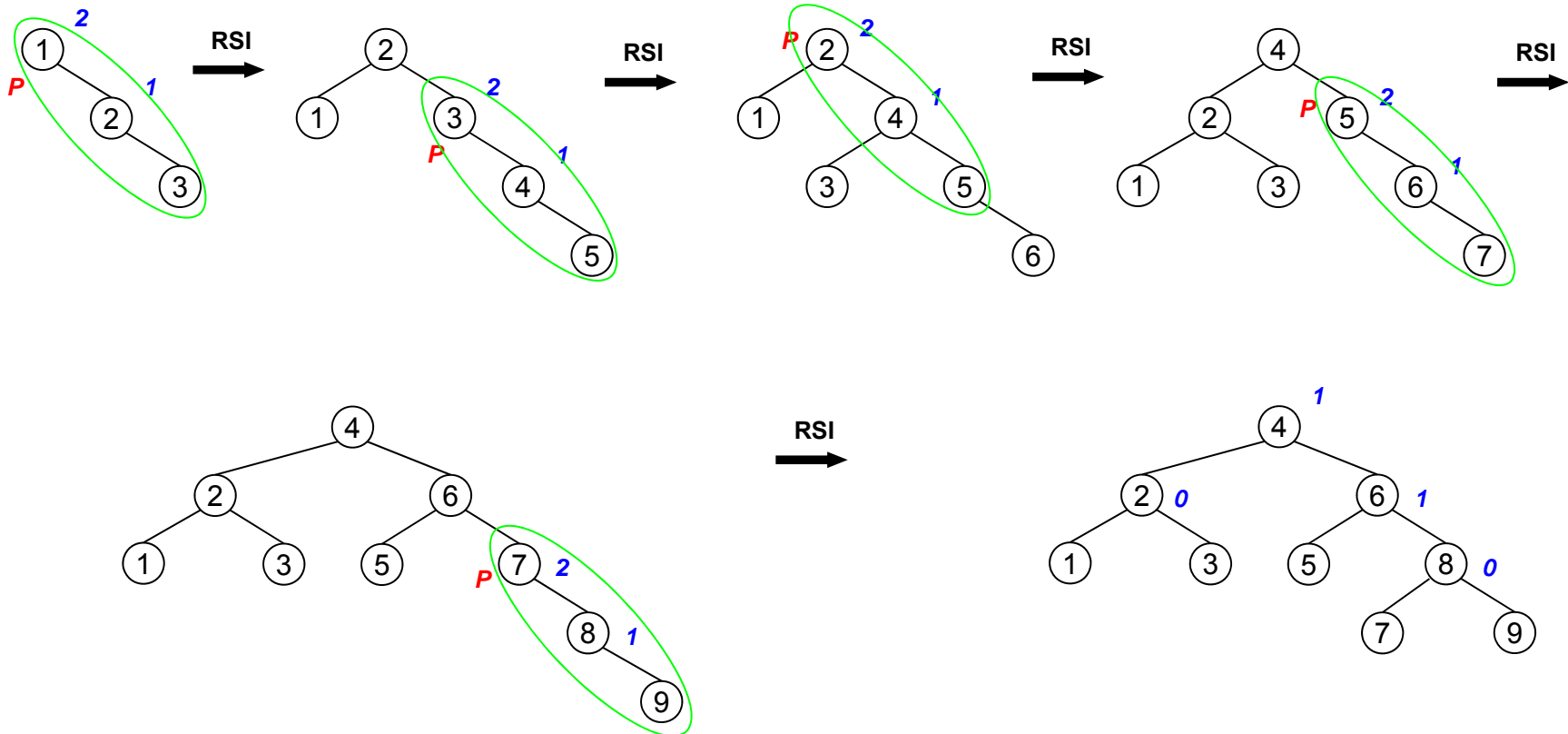
Rotaciones Simples a la Derecha



Árboles AVL

Ejemplo con rotaciones simples a la izquierda

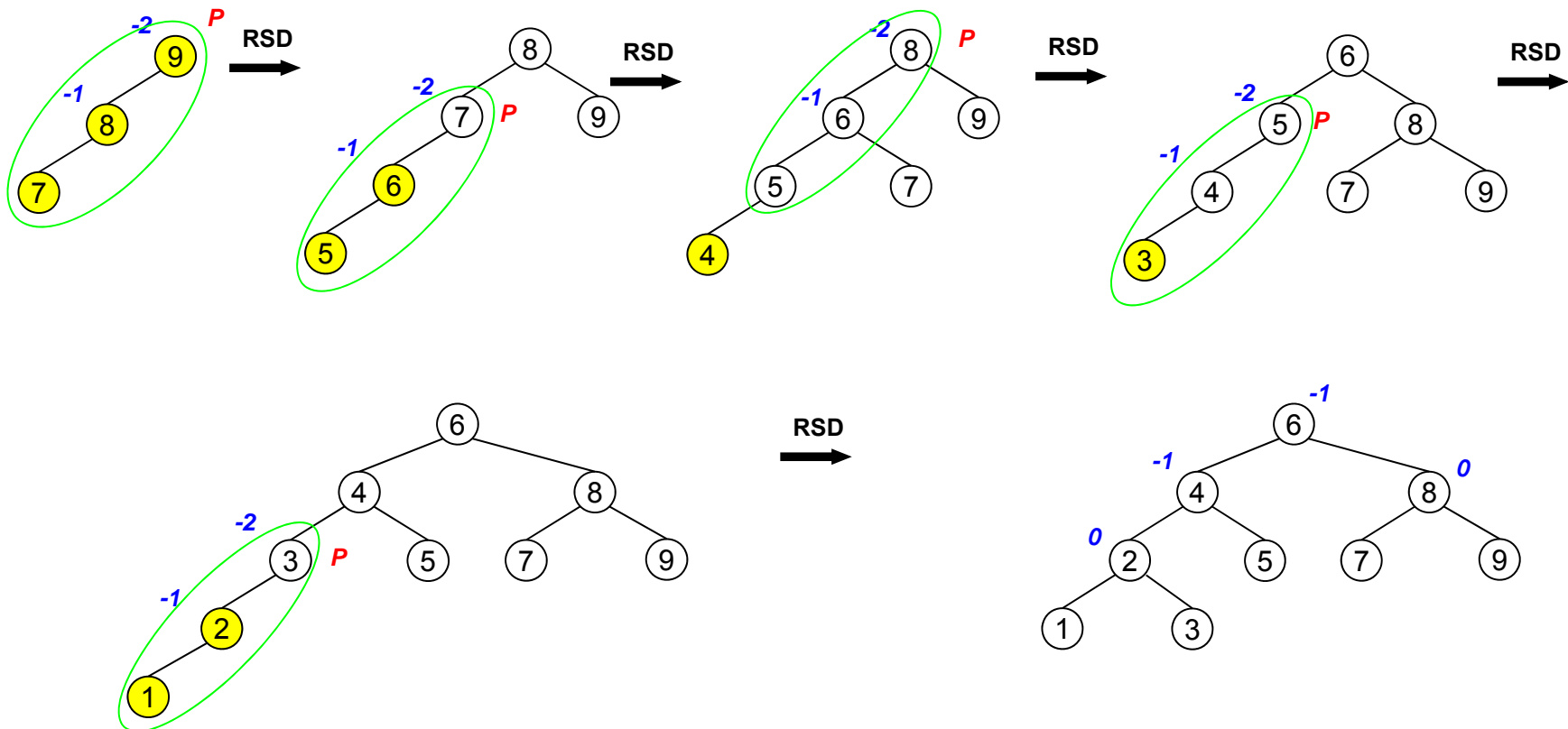
Inserte en un árbol AVL la siguiente secuencia:
1, 2, 3, 4, 5, 6, 7, 8, 9



Árboles AVL

Ejemplo con rotaciones simples a la derecha

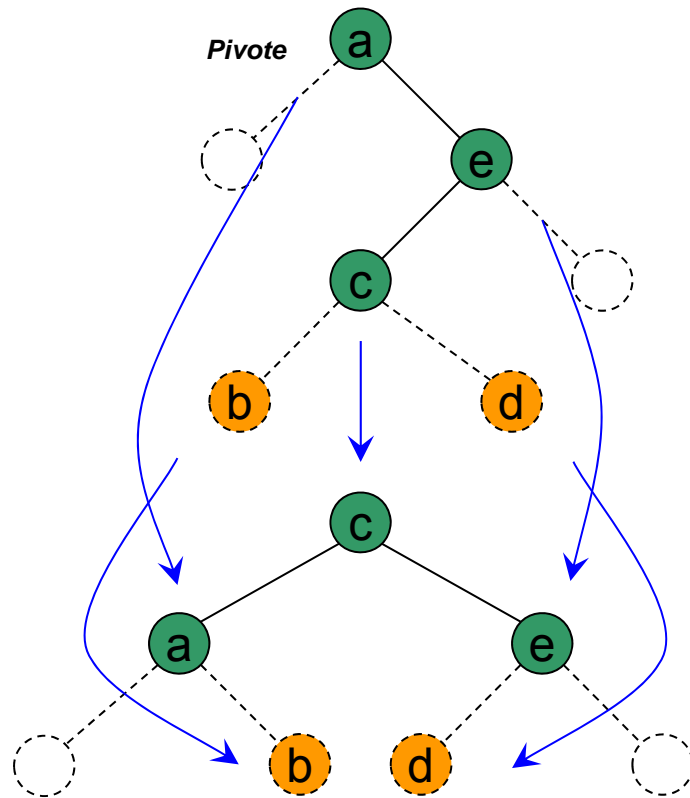
Inserte en un árbol AVL la siguiente secuencia:
9, 8, 7, 6, 5, 4, 3, 2, 1



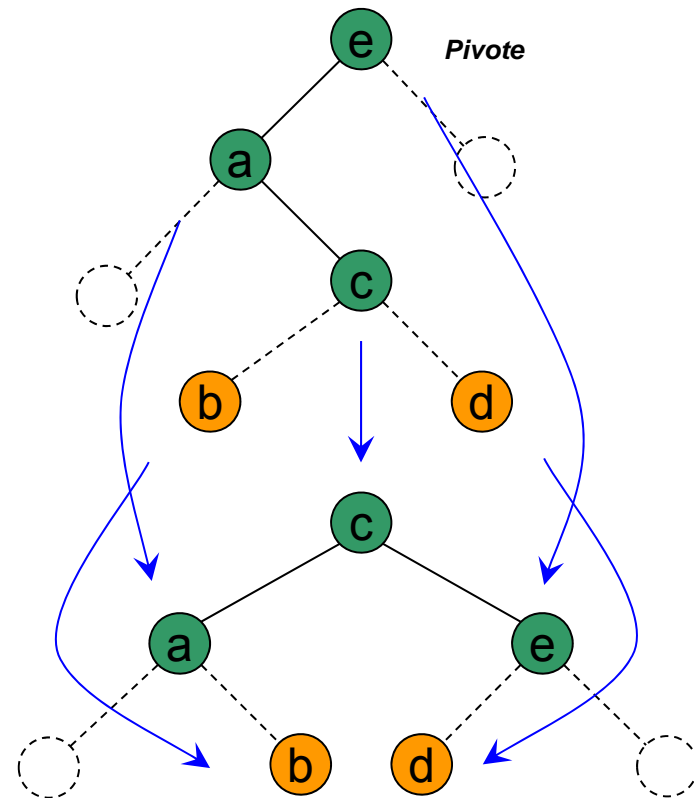
Árboles AVL

Rotaciones Dobles

A la izquierda
(RDI)



A la derecha
(RDD)



  Árboles opcionales, pueden o no existir

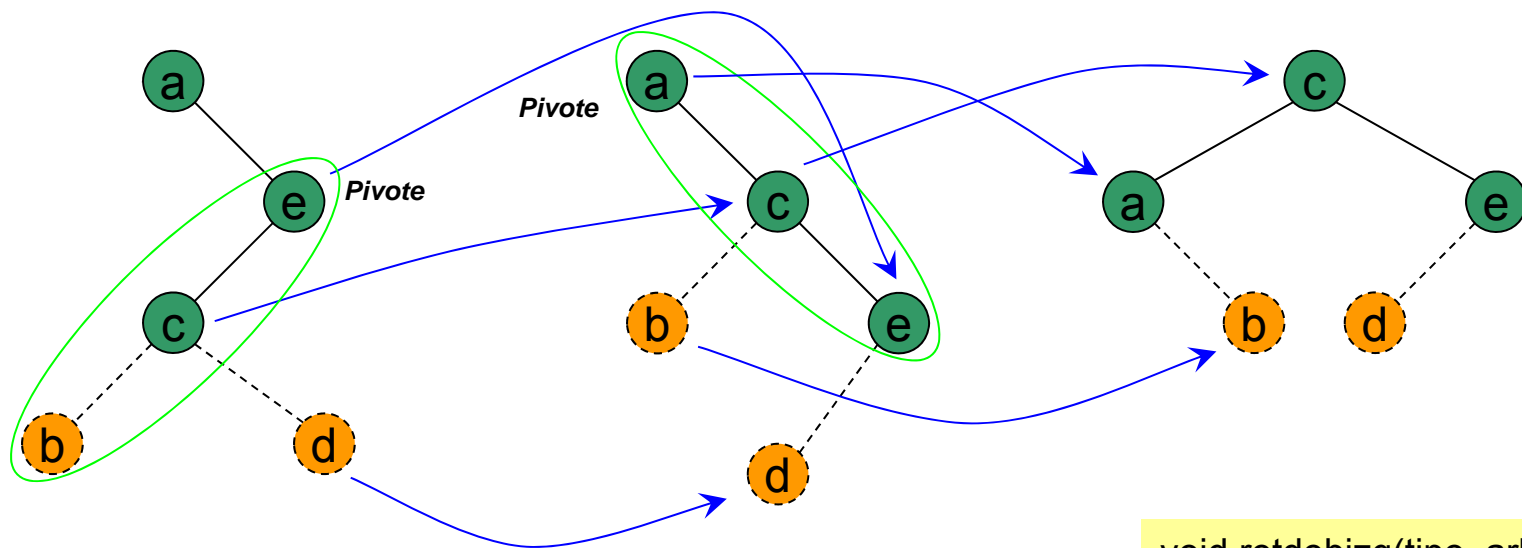
Árboles AVL

Rotaciones Dobles

A la izquierda como una secuencia de dos rotaciones simples
(RDI)

1) RSD en el subárbol derecho

2) RSI en la raíz



```
void rotdobizq(tipo_arbol *A) {  
    rotsimder(&(*A)->der);  
    rotsimizq(A);  
}
```

● Árbol opcional, puede o no existir

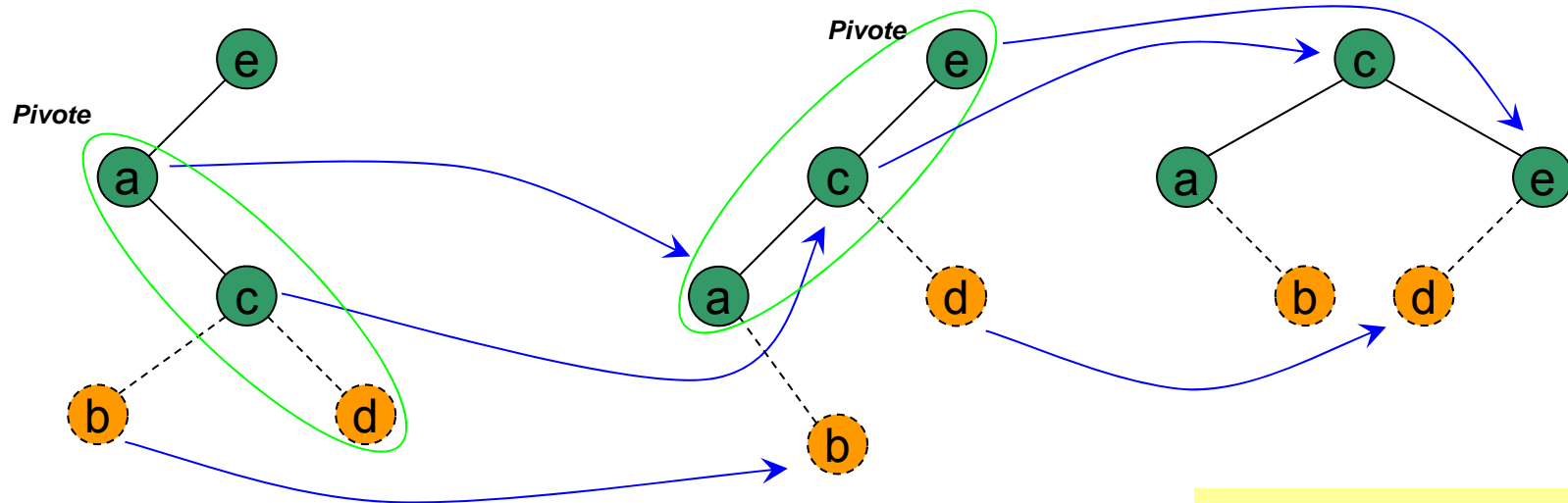
Árboles AVL

Rotaciones Dobles

A la derecha como una secuencia de dos rotaciones simples
(RDD)

1) RSI en el subárbol izquierdo

2) RSD en la raíz



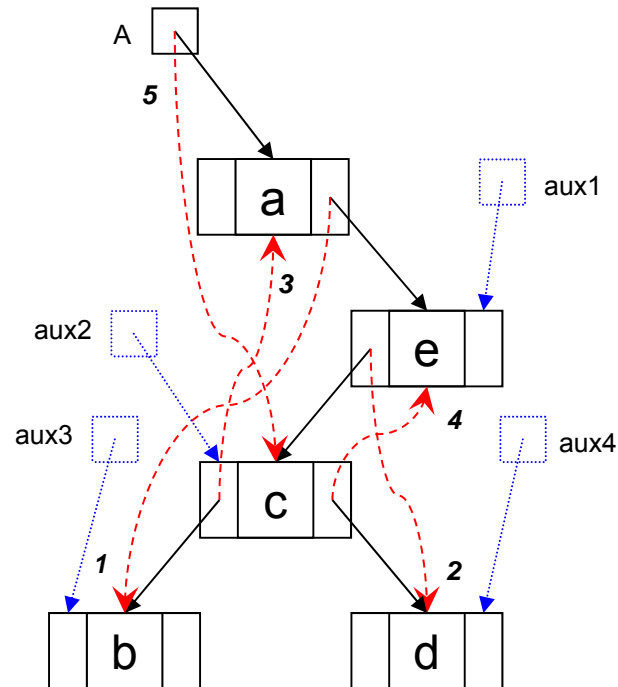
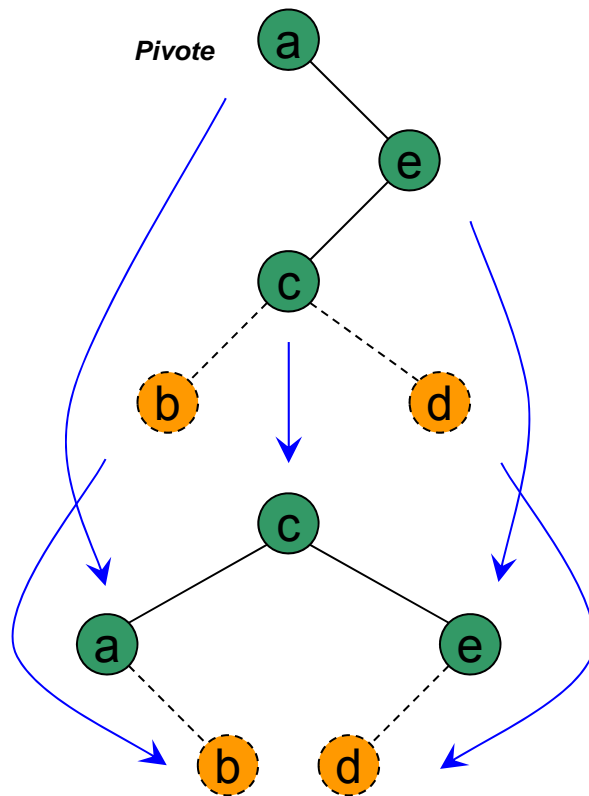
```
void rot Dobder(tipo_arbol *A) {  
    rotsim izq(&(*A)->izq);  
    rotsim der(A);  
}
```

● Árbol opcional, puede o no existir

Árboles AVL

Rotaciones Dobles

A la izquierda
(RDI)



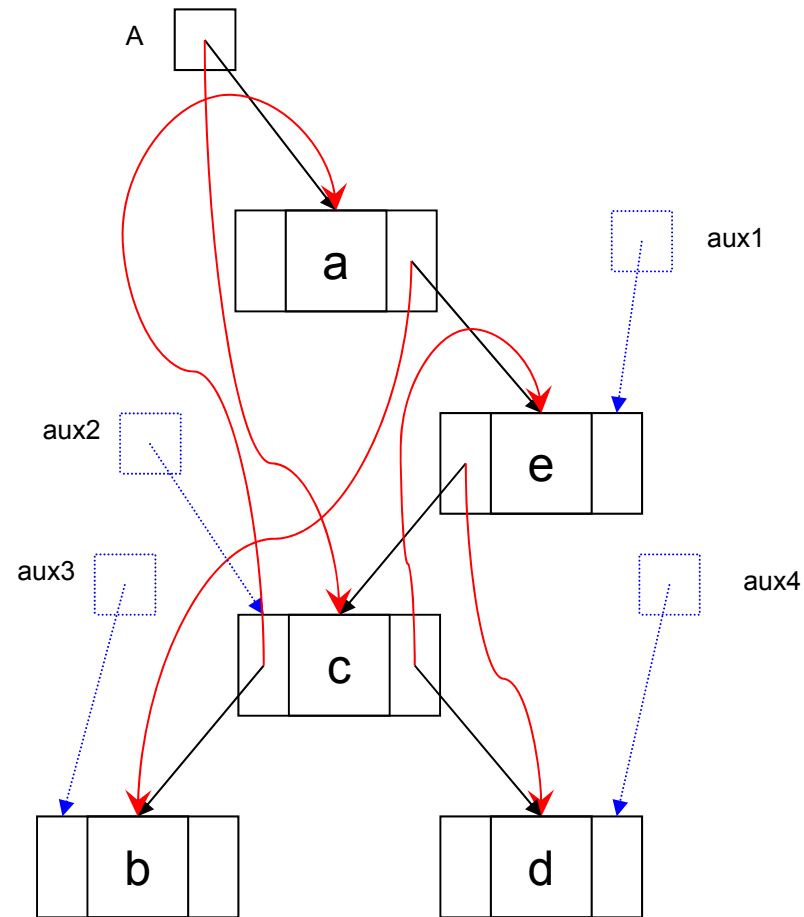
```

aux1 = (*A)->der;
aux2 = (*A)->der->izq;
aux3 = aux2 -> izq;
if(aux3 == NULL)
    aux4 = NULL
else
    aux4 = aux2->der;

(*A)->der = aux3;
aux1->izq = aux4;
aux2->izq = (*A);
aux2->der = aux1;
(*A) = aux2;
    
```


Árboles AVL

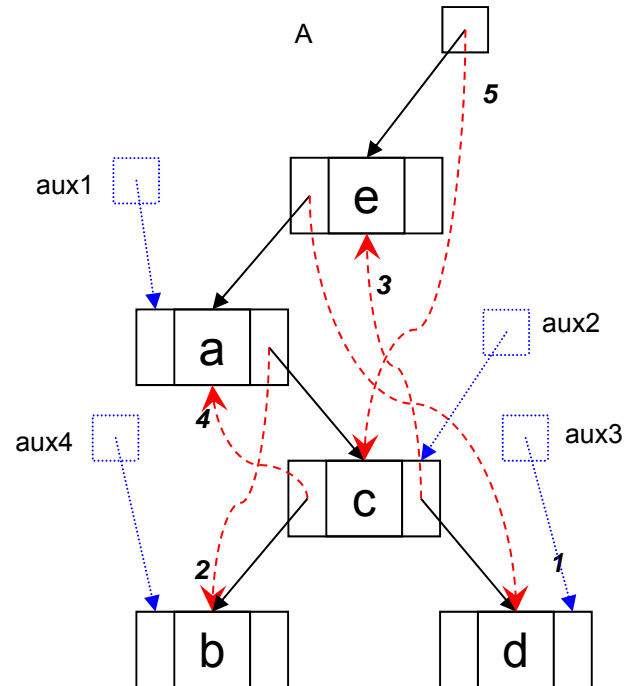
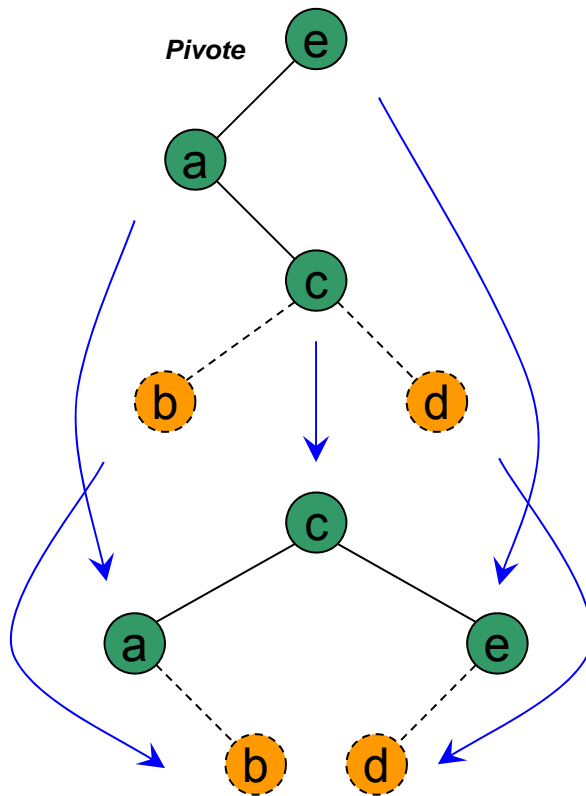
Rotaciones Dobles a la Izquierda



Árboles AVL

Rotaciones Dobles

A la derecha
(RDD)

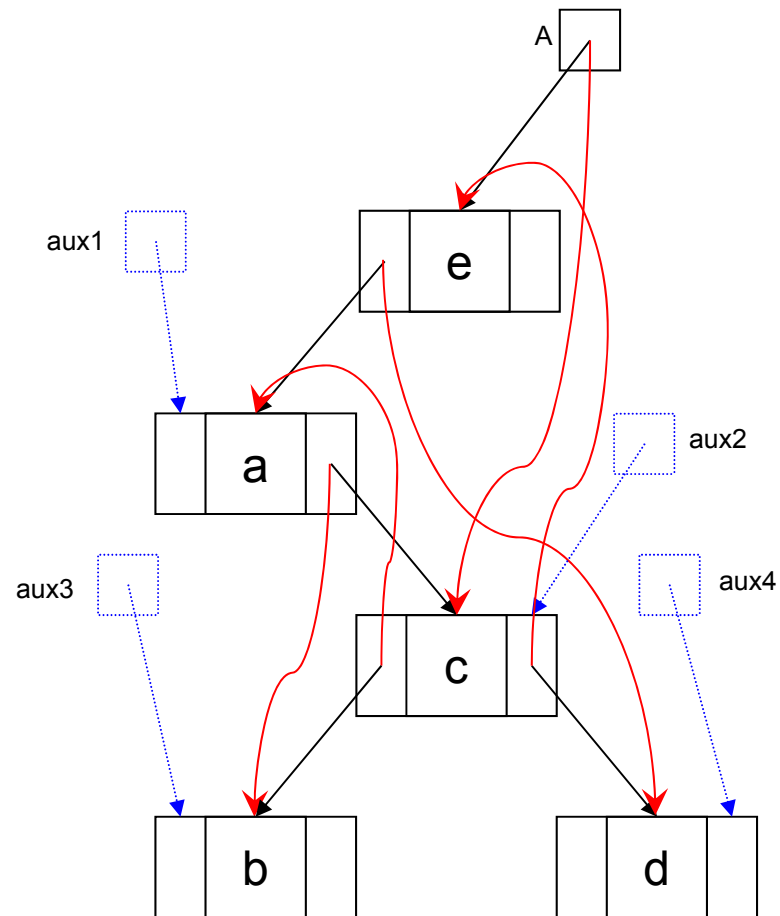


```
aux1 = (*A)->izq;
aux2 = aux1->der;
aux3 = aux2->der;
if(aux3 == NULL)
    aux4 = NULL
else
    aux4 = aux2->izq;
```

```
(*A)->izq = aux3
aux1->der = aux4;
aux2->der = (*A);
aux2->izq = aux1;
(*A) = aux2;
```

Árboles AVL

Rotaciones Dobles a la Derecha

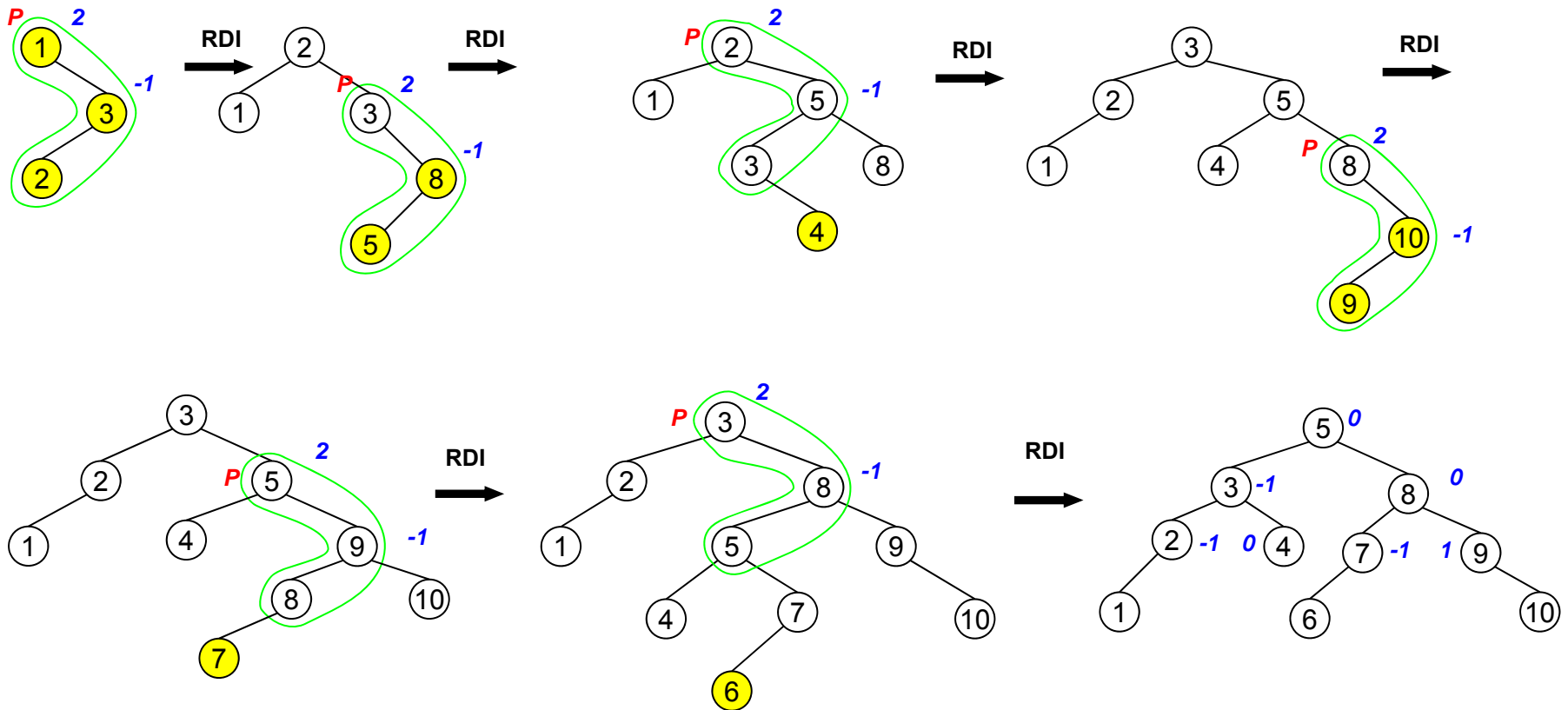


Árboles AVL

Ejemplo con rotaciones dobles a la izquierda

Inserte en un árbol AVL la siguiente secuencia:

1, 3, 2, 8, 5, 4, 10, 9, 7, 6



Árboles AVL

Ejemplo con rotaciones dobles a la derecha

Inserte en un árbol AVL la siguiente secuencia:
10, 8, 9, 3, 6, 7, 1, 2, 4, 5

