Maintaining tree balance

9/13/24

Administrivia

- Before class on Monday: Read Section 2.3 and complete RQ
- Submit clicker ID (due Tuesday night)
- HW 1 (arrays w/ fast initialization, asymptotic ordering, induction) due Tuesday night

Recall: AVL tree

[Adelson-Velsky and Landis, 1962]

- left height = height if first step must go left
- right height = height if first step must go right
- balance of a node: left height – right height

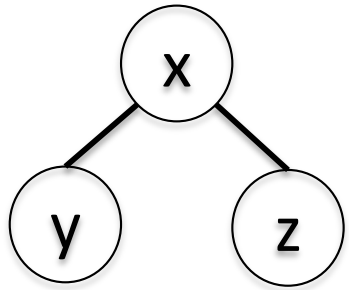
- AVL tree: Every node has balance in {0,+1,-1}

Height of an arbitrary node

- Height of a node is the height of the subtree rooted at that node

Which of the following is a formula for height(x)?

A. height(y)
B. height(y)+1
C. (height(y)+height(z))/2
D. max{height(y), height(z)} + 1
E. Not exactly one of the above

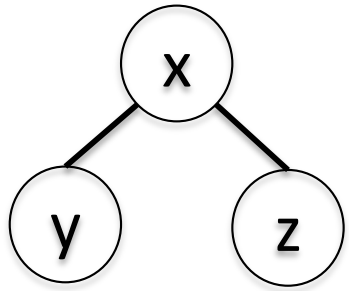


Height of an arbitrary node

- Height of a node is the height of the subtree rooted at that node

Which of the following is a formula for height(x)?

A. height(y)
B. height(y)+1
C. (height(y)+height(z))/2
D. max{height(y), height(z)} + 1
E. Not exactly one of the above



How does the height of a tree change when a node is inserted (using insertion as you learned in 142)?

A. Increases by 1
B. Decreases by 1
C. Remains the same
D. Cannot be determined
E. None of the above

How does the height of a tree change when a node is inserted (using insertion as you learned in 142)?

A. Increases by 1

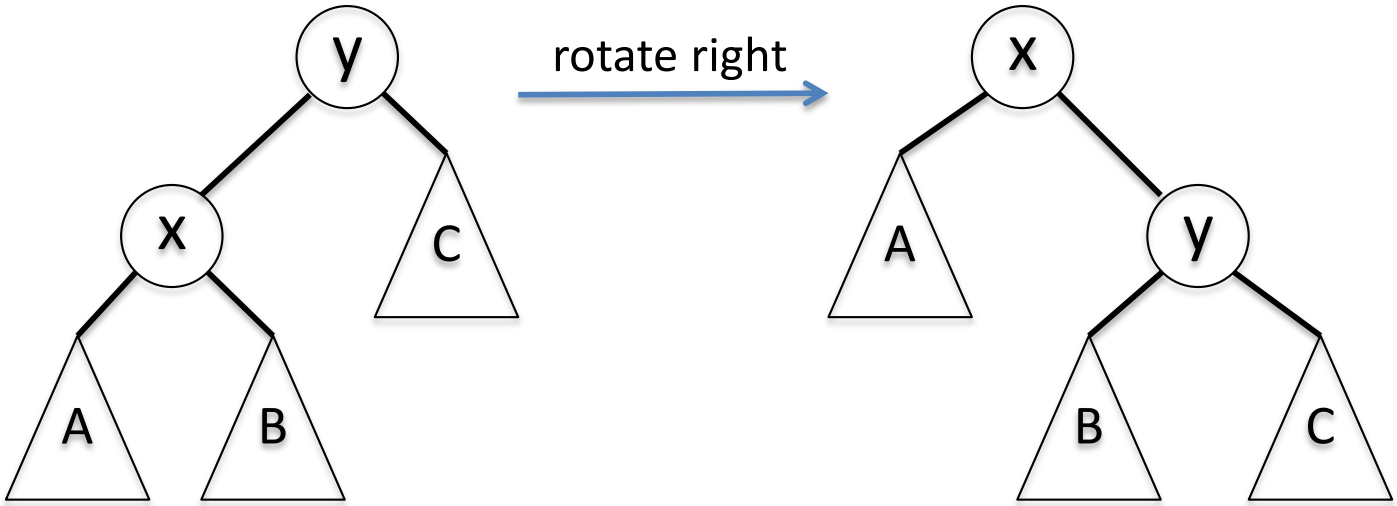
B. Decreases by 1

C. Remains the same

D. Cannot be determined

E. None of the above

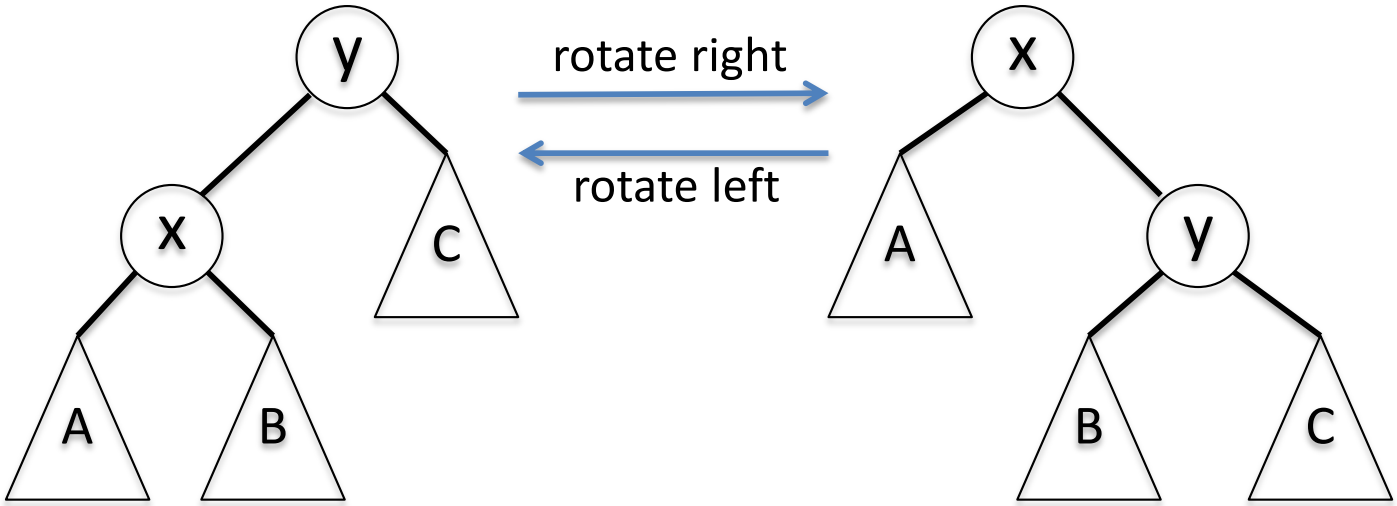
Rotations



Triangles denote arbitrary subtrees

- NOT necessarily the same size

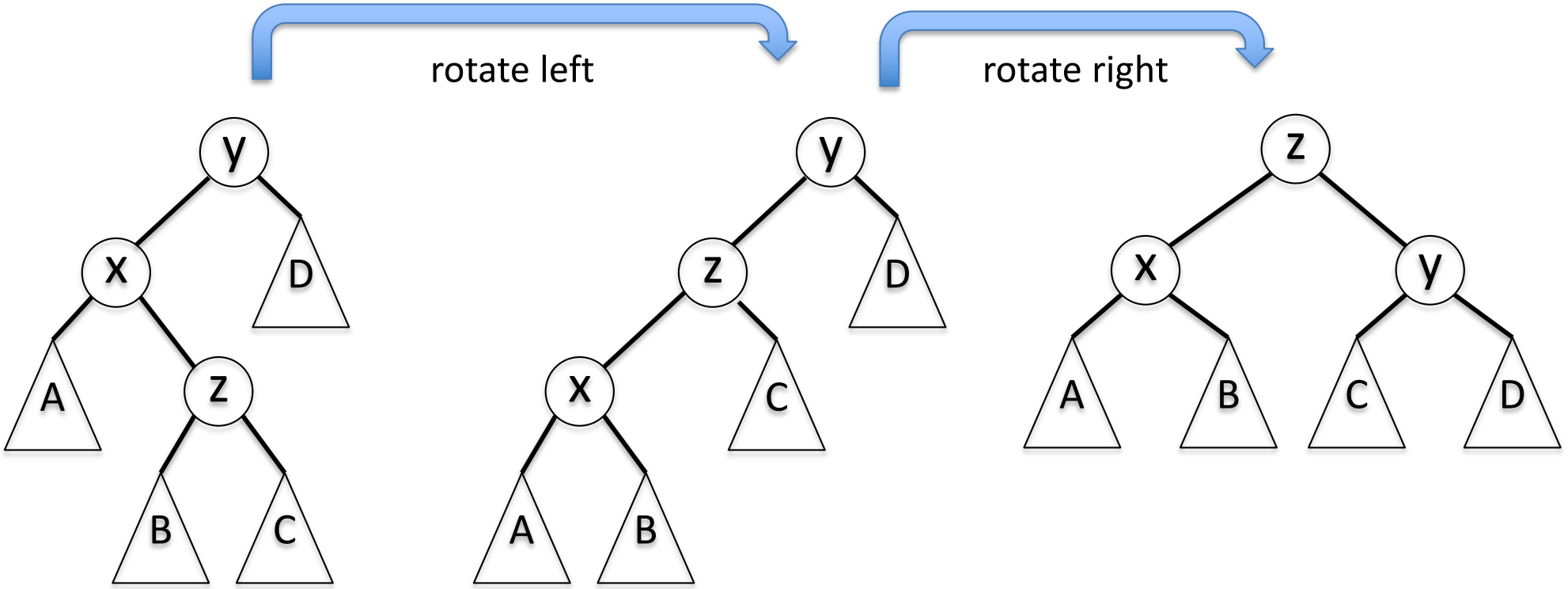
Rotations



Triangles denote arbitrary subtrees

- NOT necessarily the same size

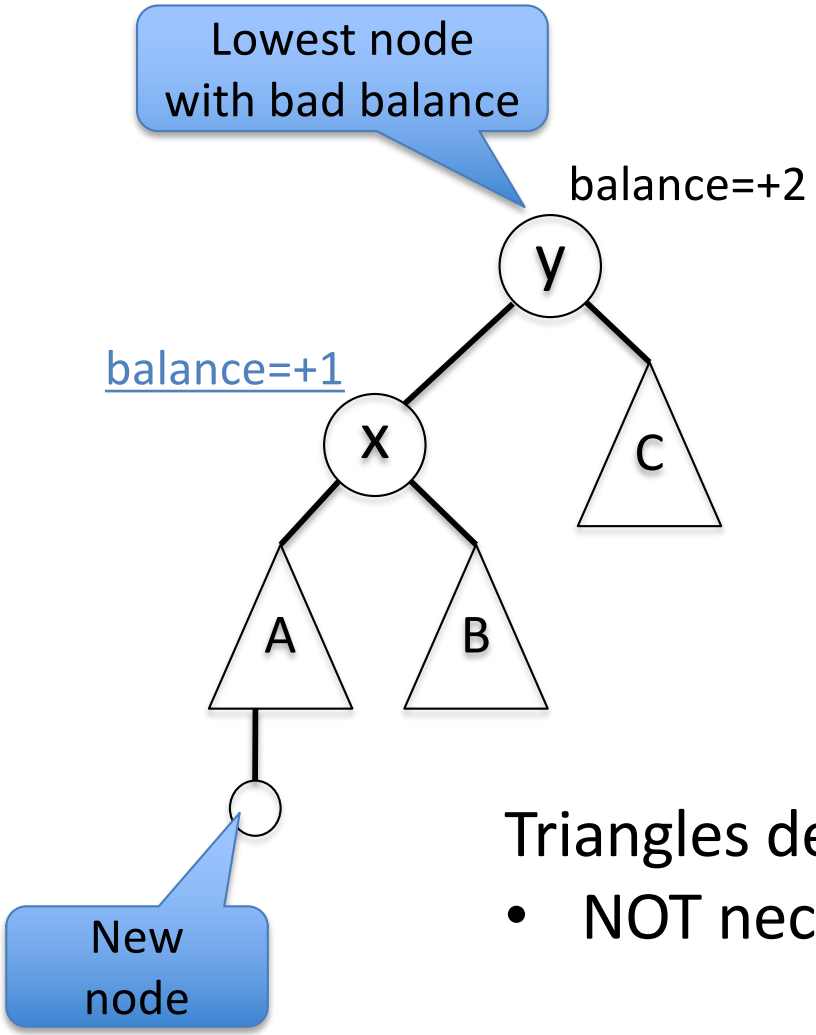
Double rotations



Fixing an AVL tree after insertion

- Find lowest node y that is out of balance
- If balance(y) = 2, check left neighbor x
  - If balance(x)=1, single right rotation
  - Else double rotation (left and then right)
- Else check right neighbor x
  - If balance(x) = -1, single left rotation
  - Else double rotation (right and then left)

Insertion case 1: Adding to “outside”

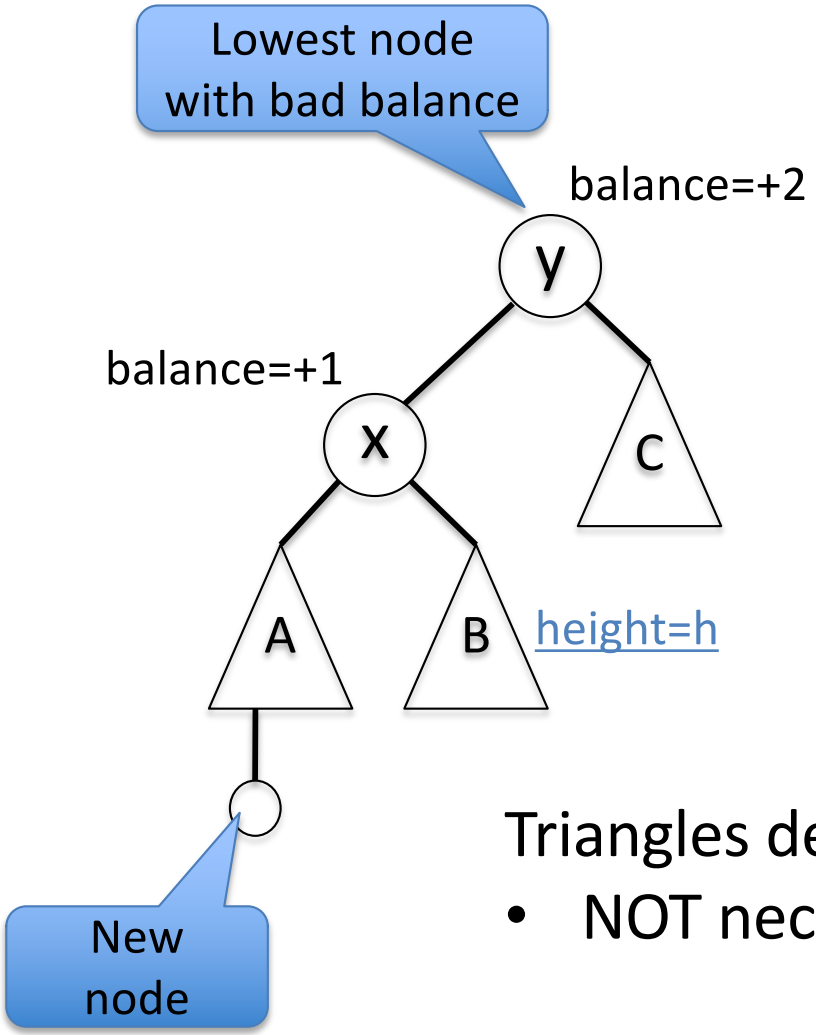


Triangles denote arbitrary subtrees

- NOT necessarily the same size

Subtree A includes the new node

Insertion case 1: Adding to "outside"



Let h be the height of B. What are the heights of A and C?

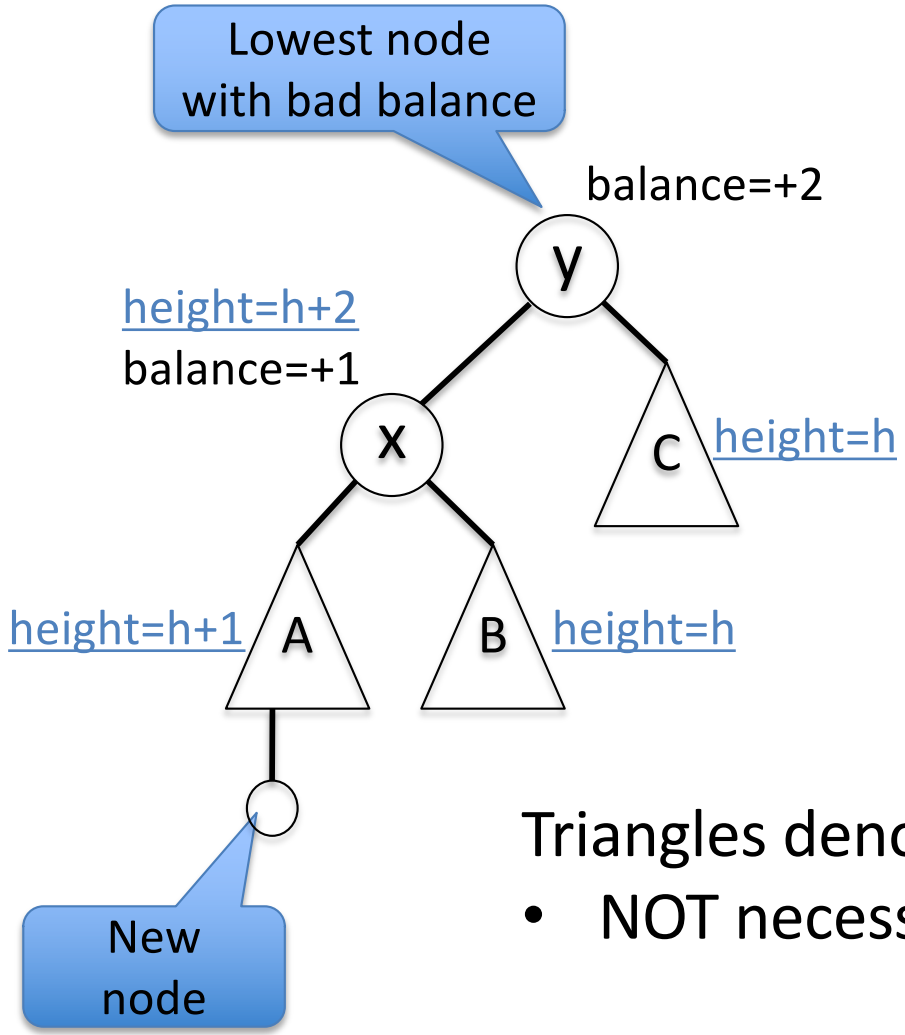
A. h and h+1
B. h+1 and h
C. h+1 and h+2
D. None of the above
E. Cannot be determined

Triangles denote arbitrary subtrees

- NOT necessarily the same size

Subtree A includes the new node

Insertion case 1: Adding to "outside"



Let h be the height of B. What are the heights of A and C?

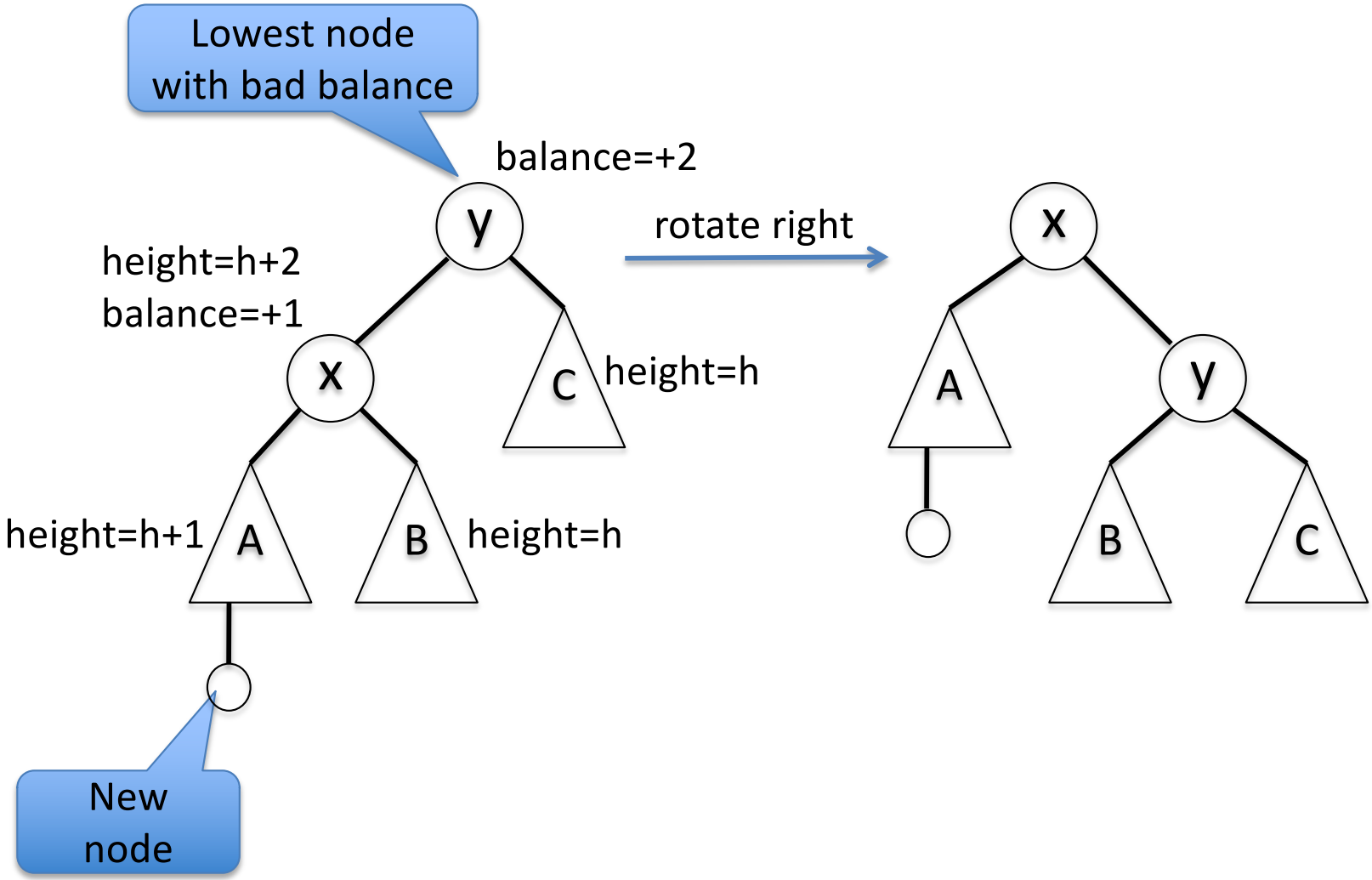
A. h and h+1
B. h+1 and h
C. h+1 and h+2
D. None of the above
E. Cannot be determined

Triangles denote arbitrary subtrees

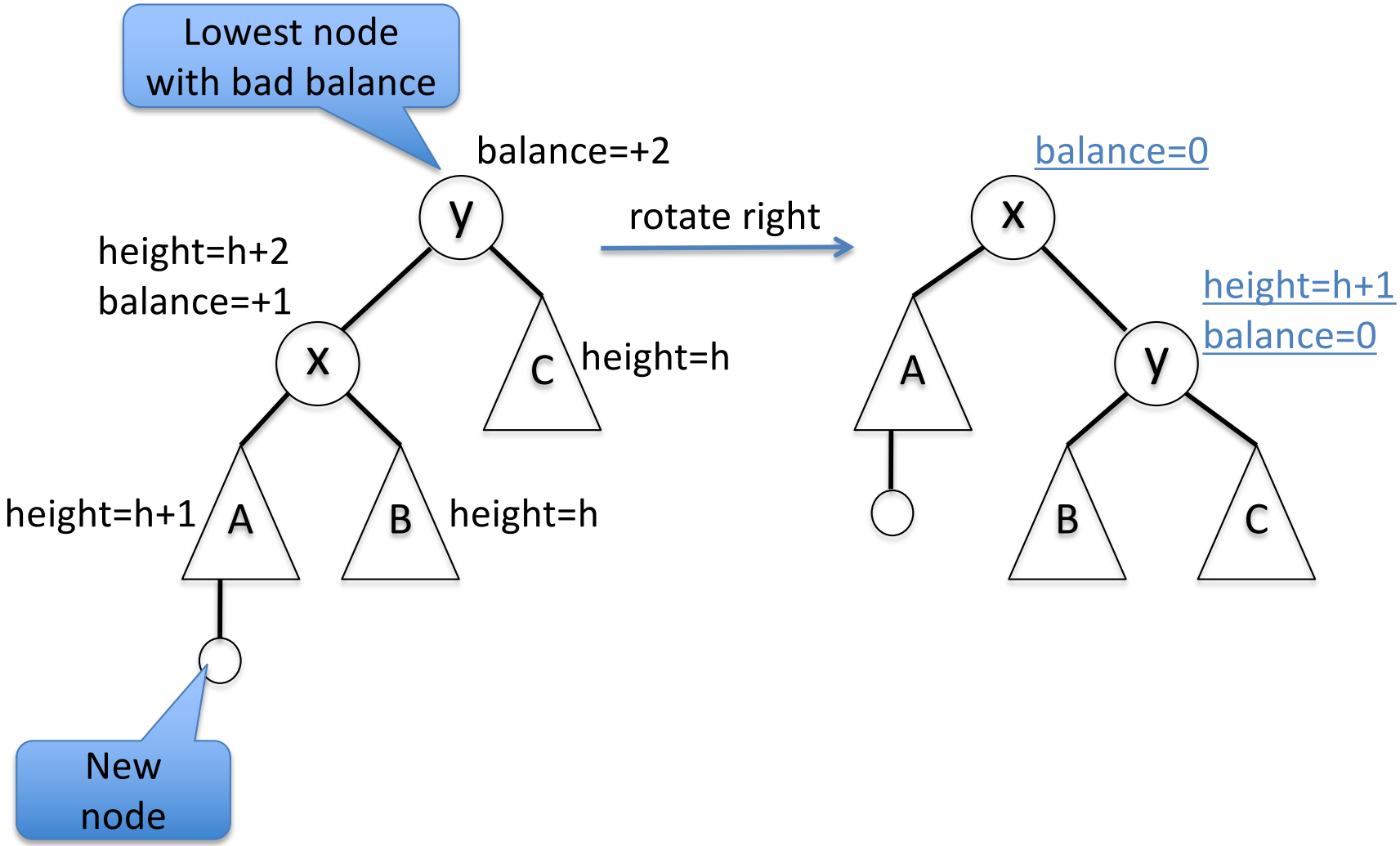
- NOT necessarily the same size

Subtree A includes the new node

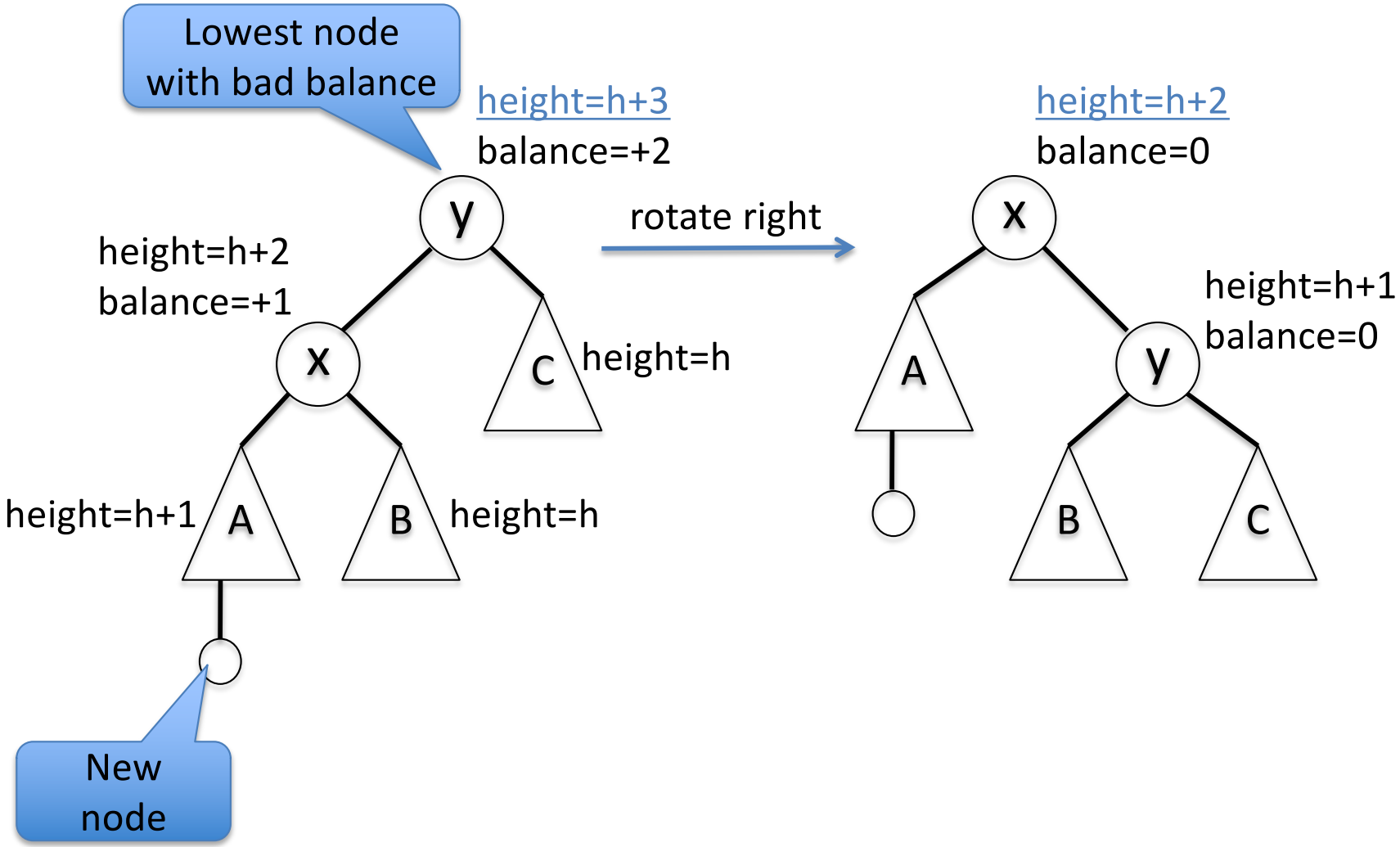
Insertion case 1: Adding to “outside”



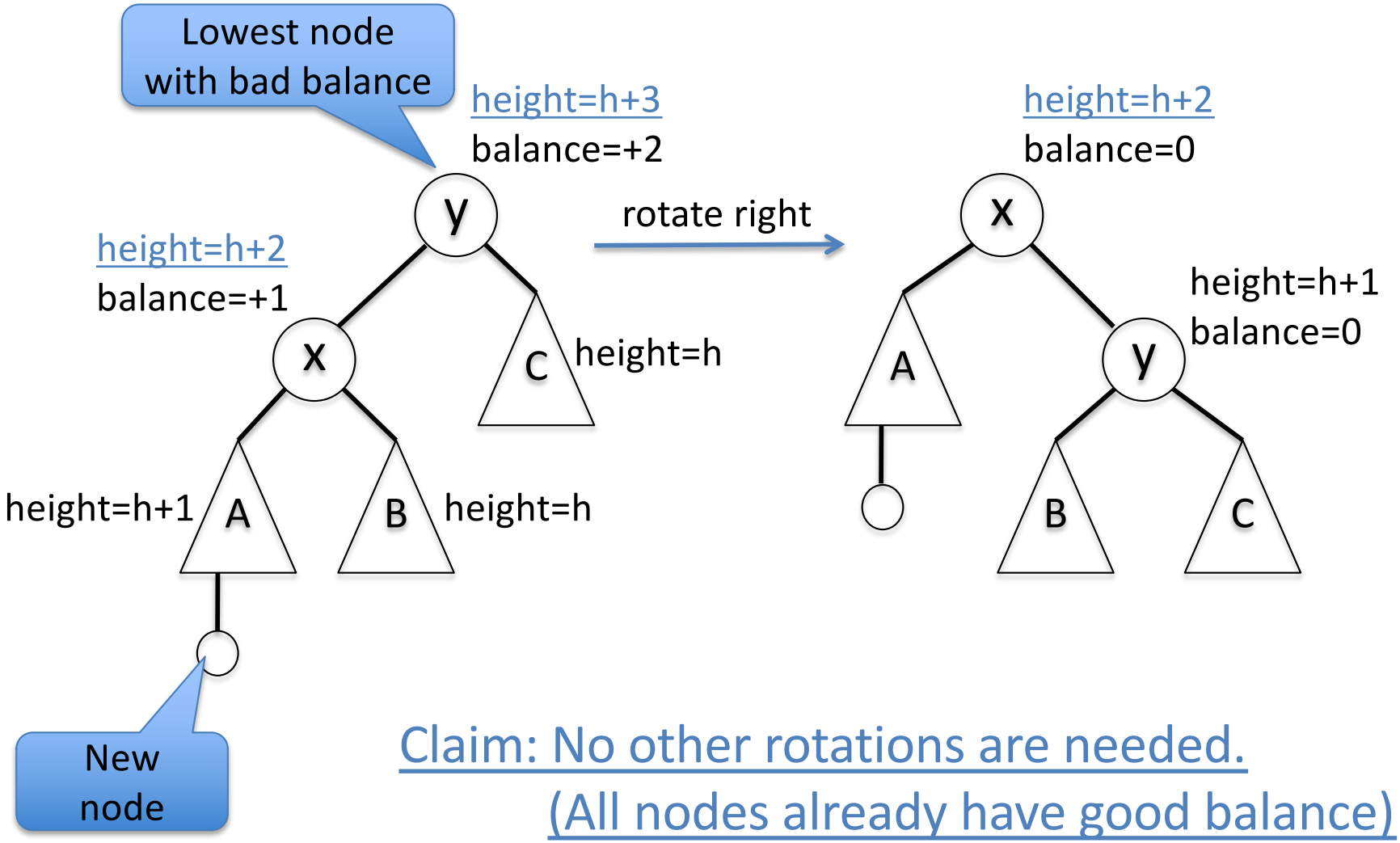
Insertion case 1: Adding to “outside”



Insertion case 1: Adding to “outside”

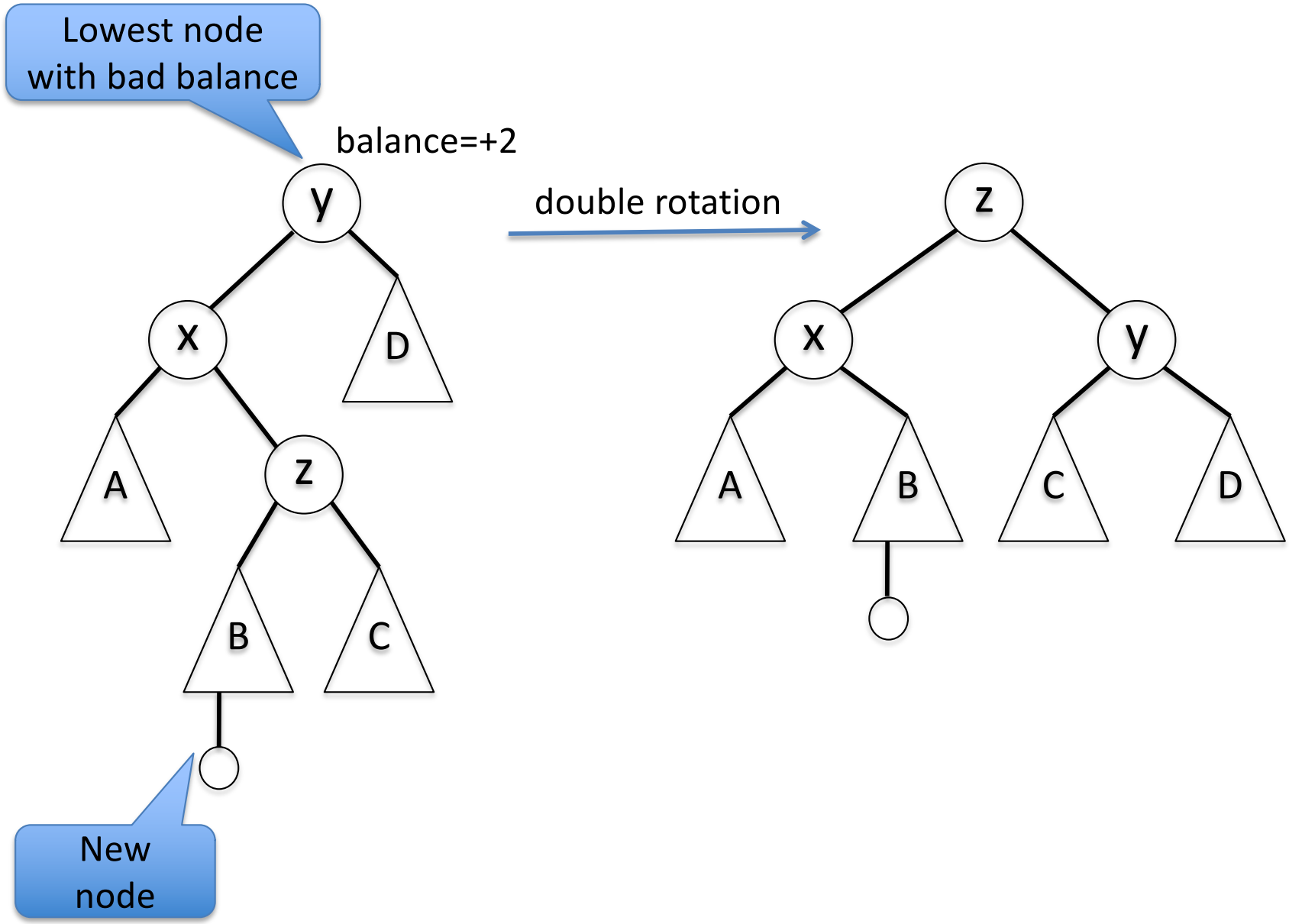


Insertion case 1: Adding to "outside"

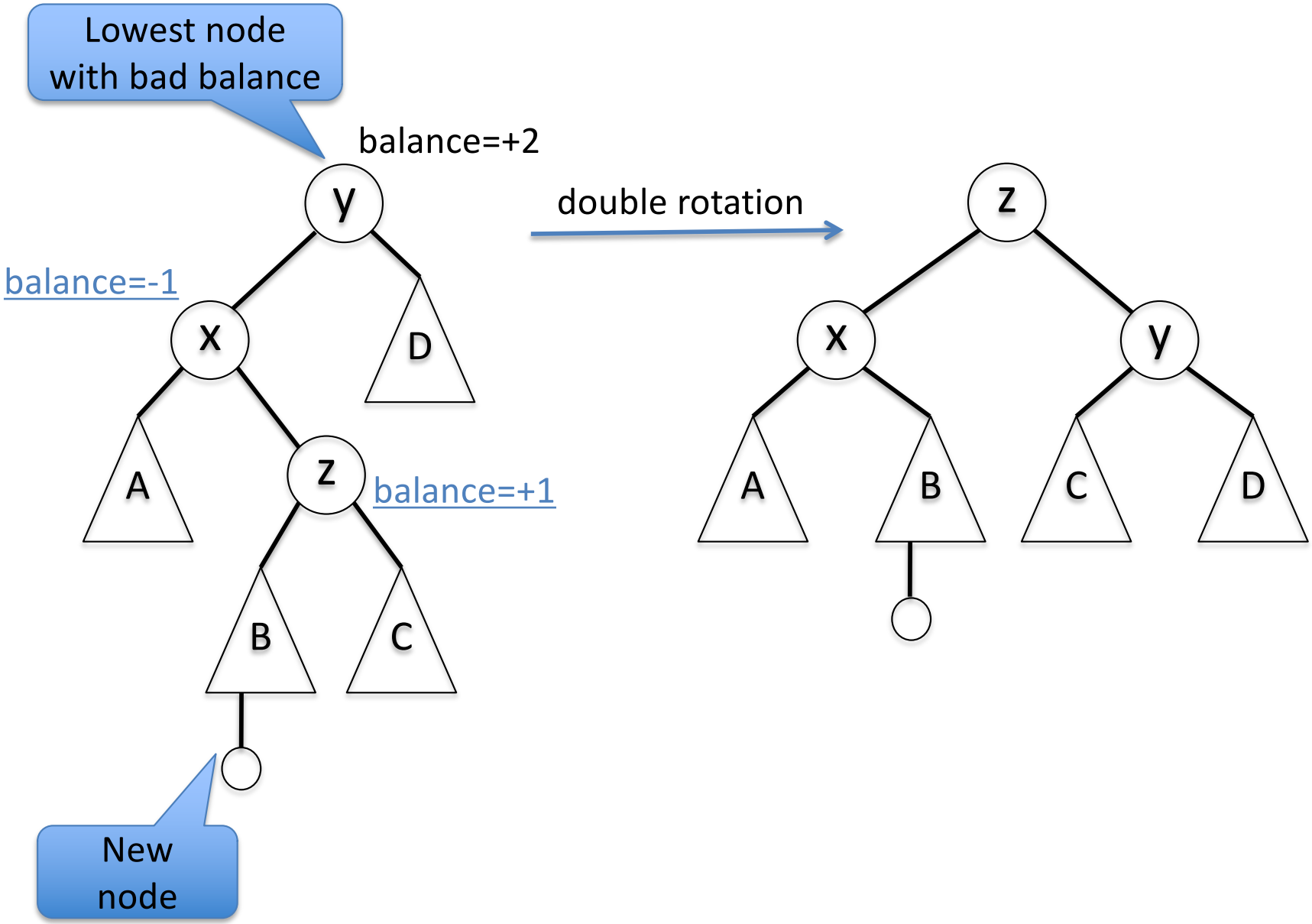


Claim: No other rotations are needed.
(All nodes already have good balance)

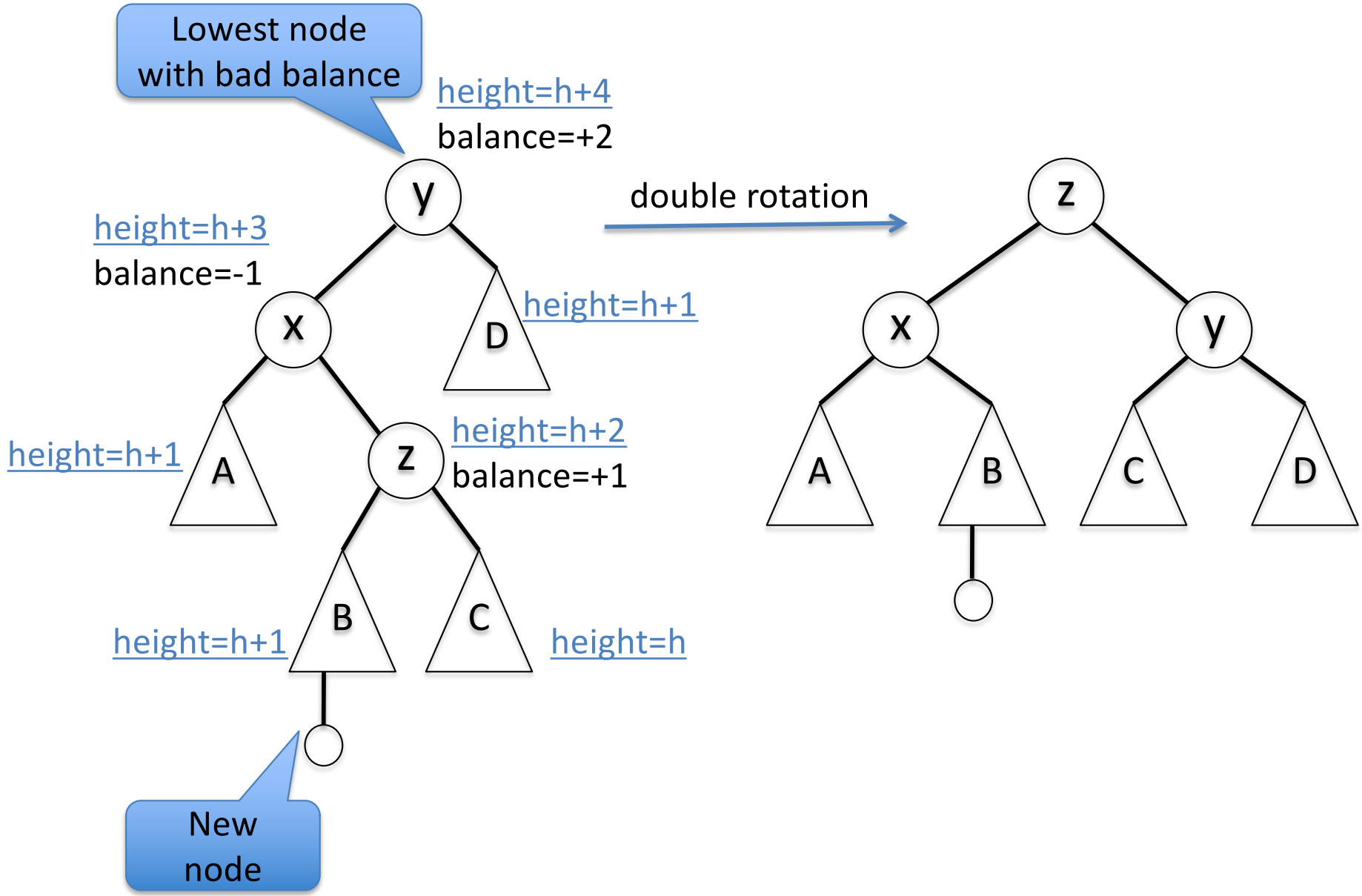
Insertion case 2a: Adding to “inside” (on left)



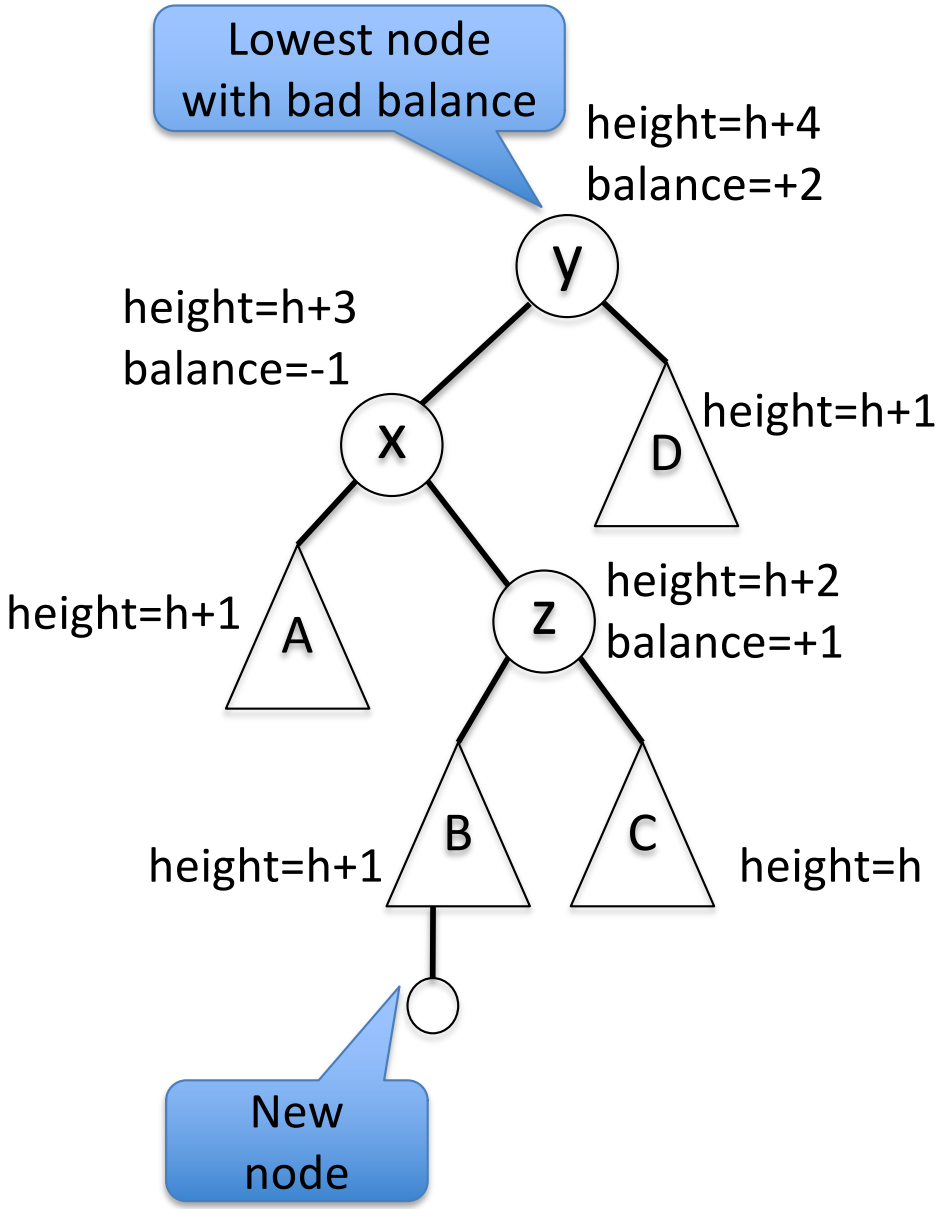
Insertion case 2a: Adding to “inside” (on left)

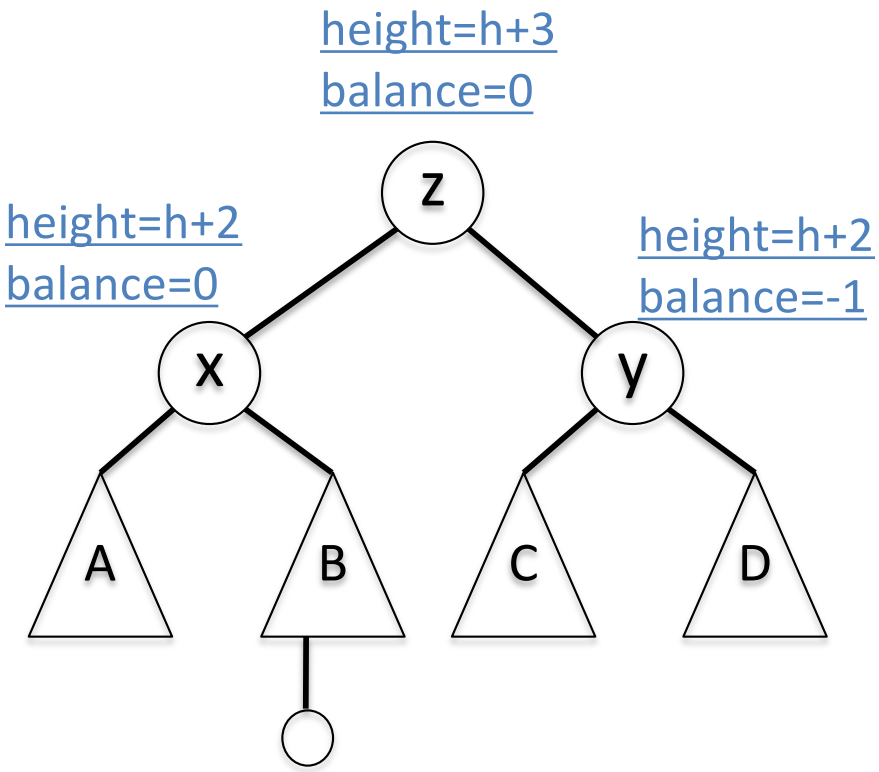


Insertion case 2a: Adding to “inside” (on left)

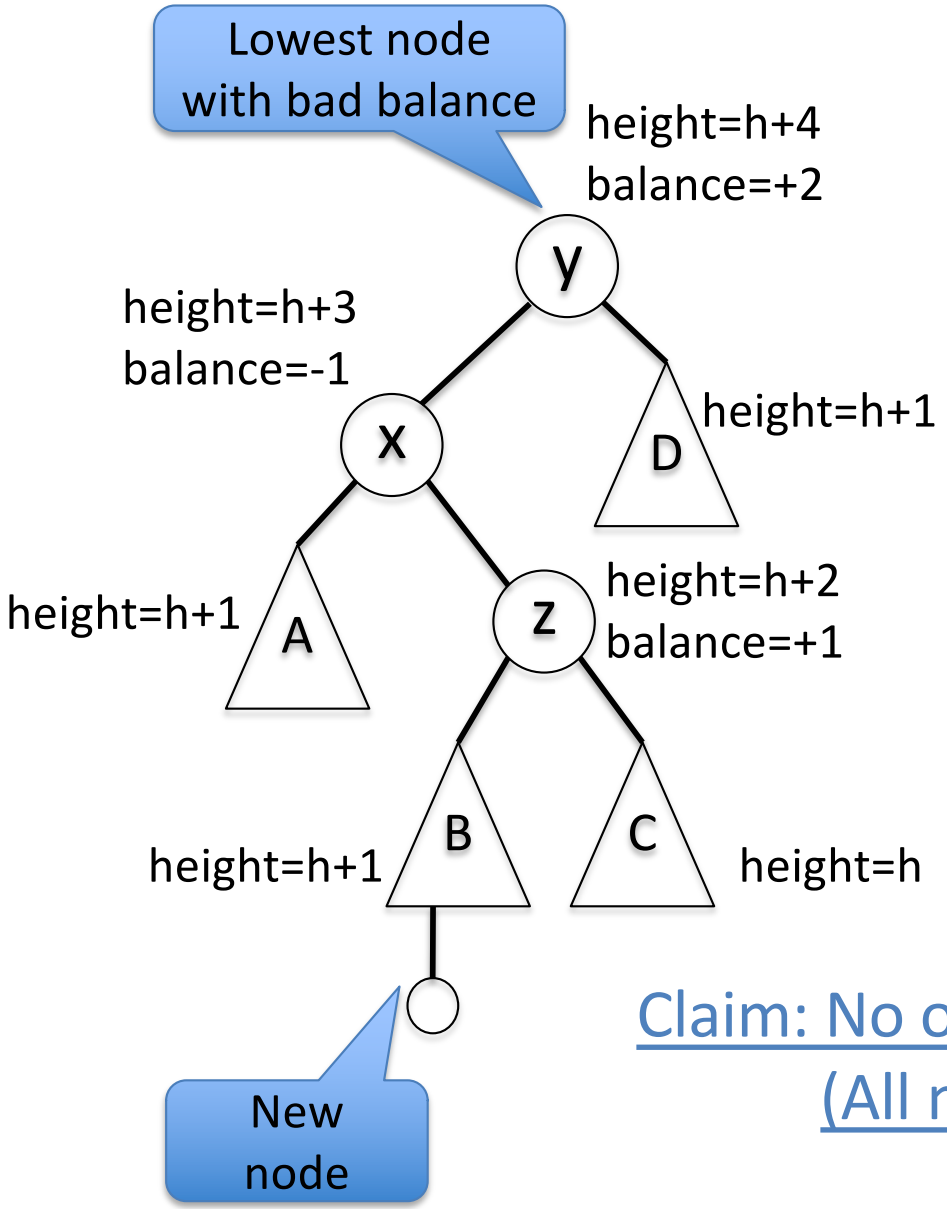


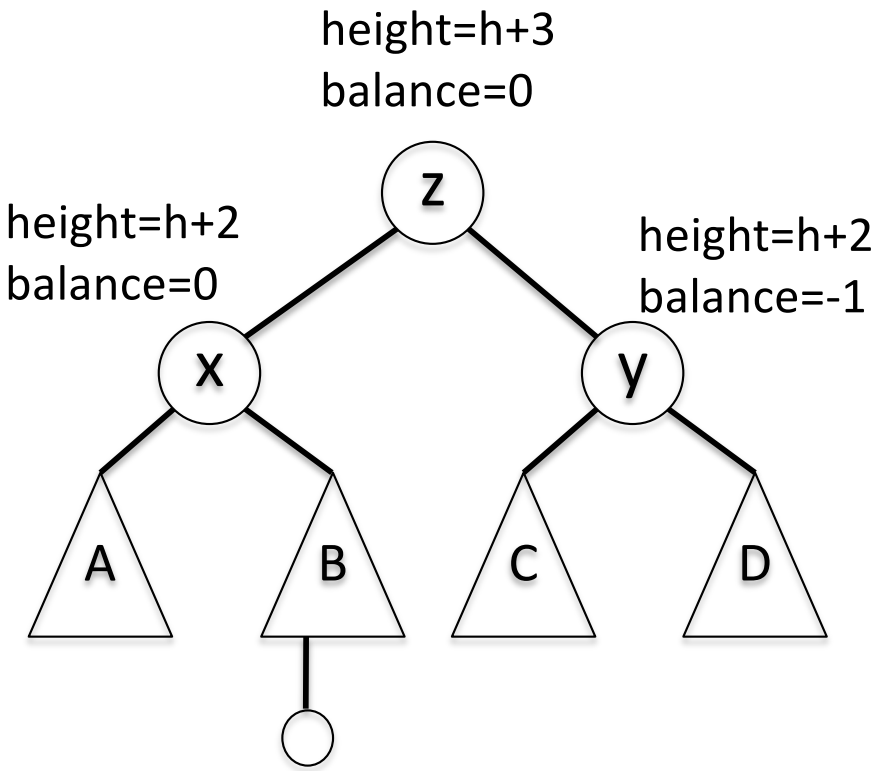
Insertion case 2a: Adding to “inside” (on left)





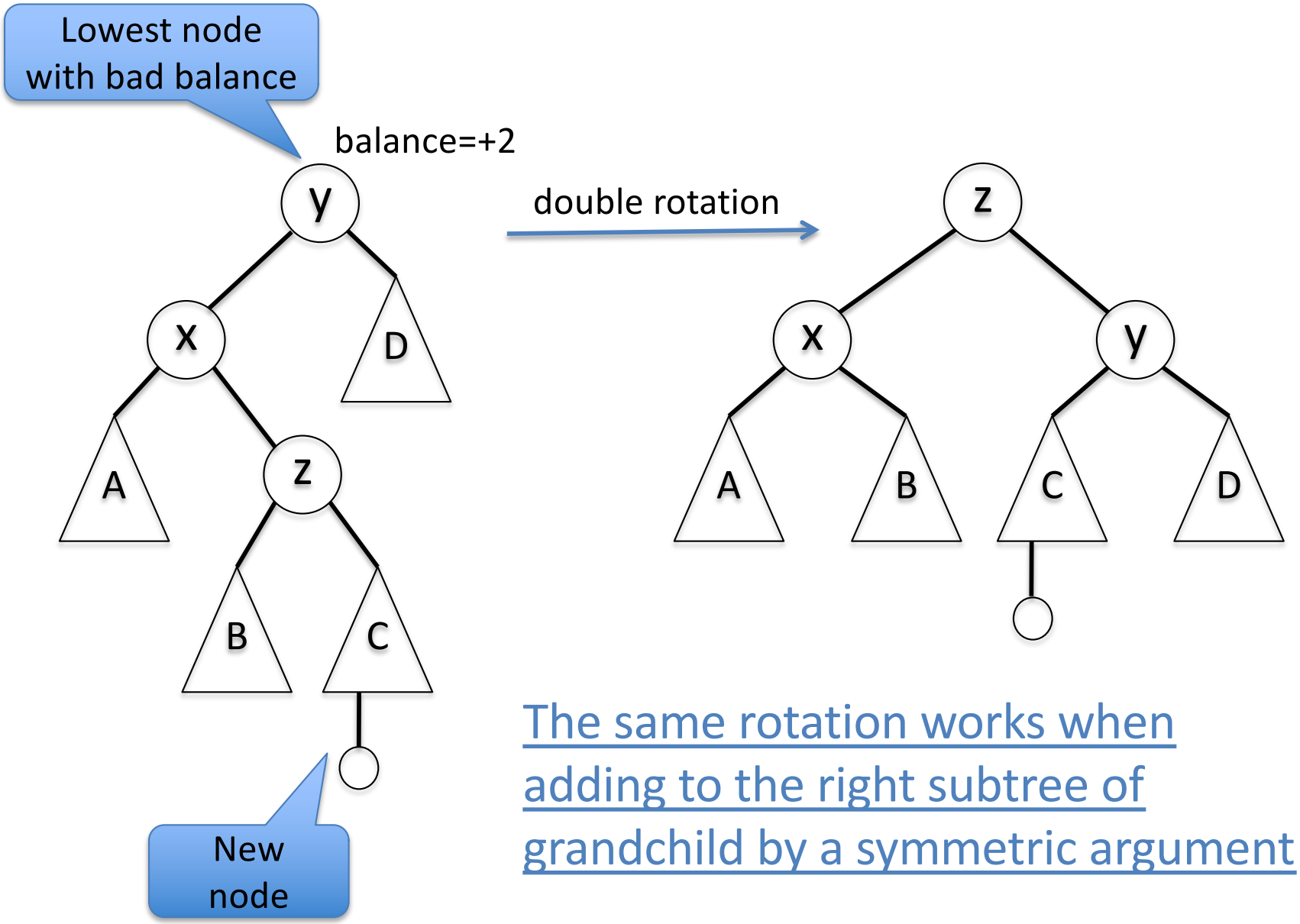
Insertion case 2a: Adding to "inside" (on left)





Claim: No other rotations are needed.
(All nodes already have good balance)

Insertion case 2b: Adding to “inside” (on right)



The same rotation works when adding to the right subtree of grandchild by a symmetric argument

What about deletions?