Algorithm and Data Structure Analysis (ADSA)

AVL-Trees

Overview

AVL-Trees:

- Find, insert, remove

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Runtimes for Binary Search Tree

Find, insert, remove:

Worst case: $\Theta(n)$

Best case: $\Theta(\log n)$

Average case: $\Theta(\log n)$

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Aim: Time O(log n) in the worst case

AVL-Tree

Observation:

- Binary search trees can get imbalanced when applying insert and/or remove operations.

Idea:

- Whenever a subtree rooted at a node v gets imbalanced, apply operations that balance it out in time O(log n).

AVL Tree

Let h(T) be the height of a tree T.

Let v be a node in T and T_l and T_r be the left and right subtree of v.

We denote by b(v) = h(T_l) − h(T_r) the balance degree of v.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Definition: A binary search tree T is called an AVL-tree if for each $v \in T$, $b(v) \in \{-1, 0, 1\}$ holds.

Height of an AVL-tree

Theorem(without proof) Let T be an AVL-tree consisting of n nodes. Then $h(T) \leq 1.44 \log n$
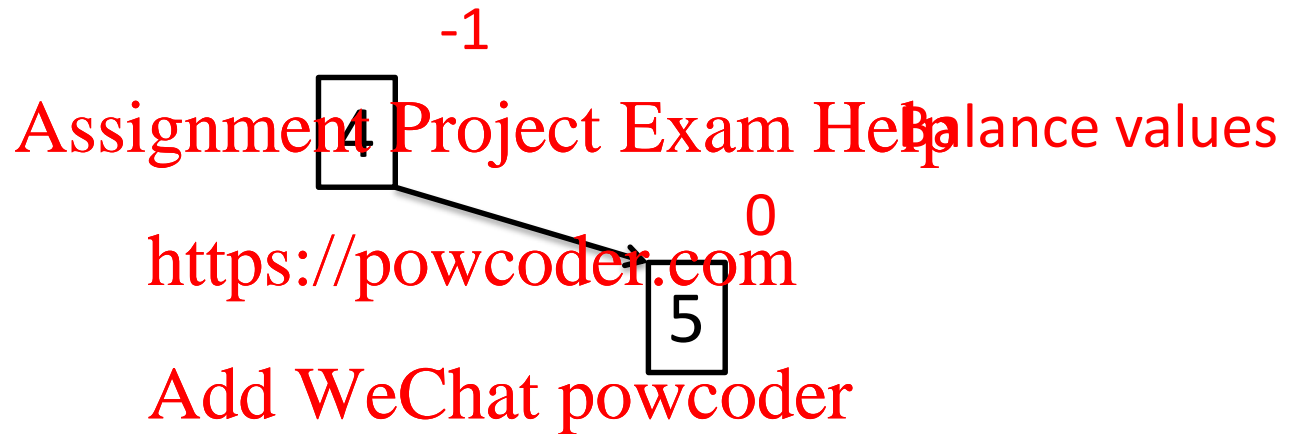
Assignment Project Exam Help

https://powcoder.com

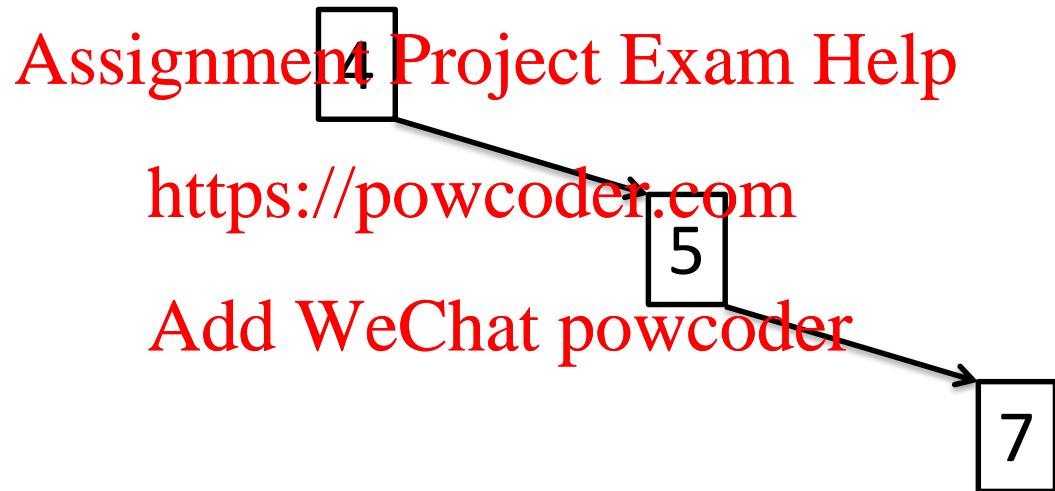We have to consider the operations find, insert, and delete for AVL-trees.

Add WeChat powcoder

- Find is as for Binary Search Trees.

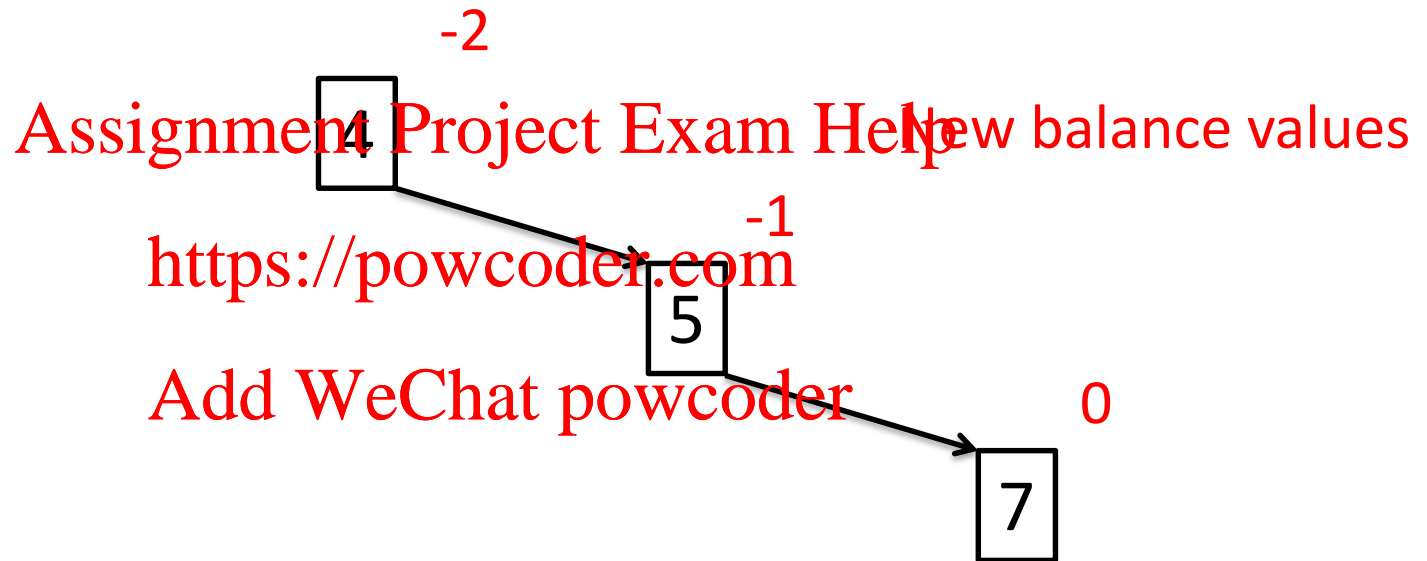- For insert and remove we might have to rebalance the tree.
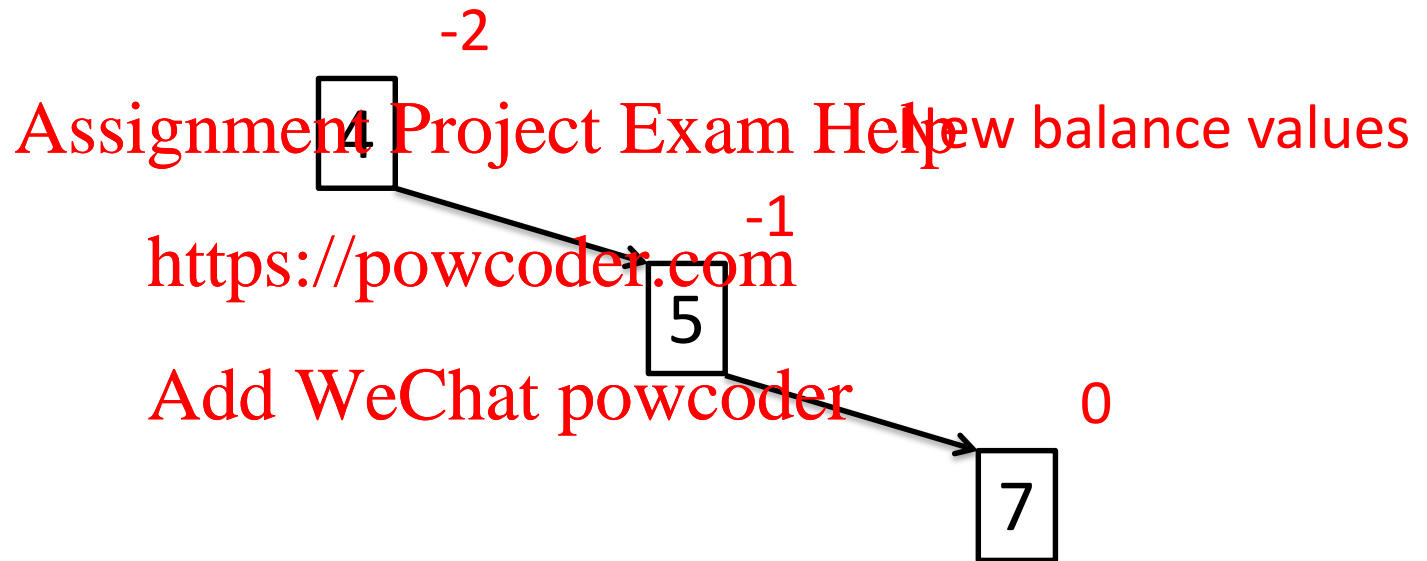
Example Insert

-1

4

balance values

0

5

Insert 7

Example Insert

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Consider path from new leaf
to the root and check balance values

Example Insert

-2

4

-1

5

0

7

New balance values

Example Insert

-2

4

New balance values

-1

5

0

7

AVL-property at node 4 violated

Example Insert

0

5

0   0

4   7

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Rotation establishes AVL-property again
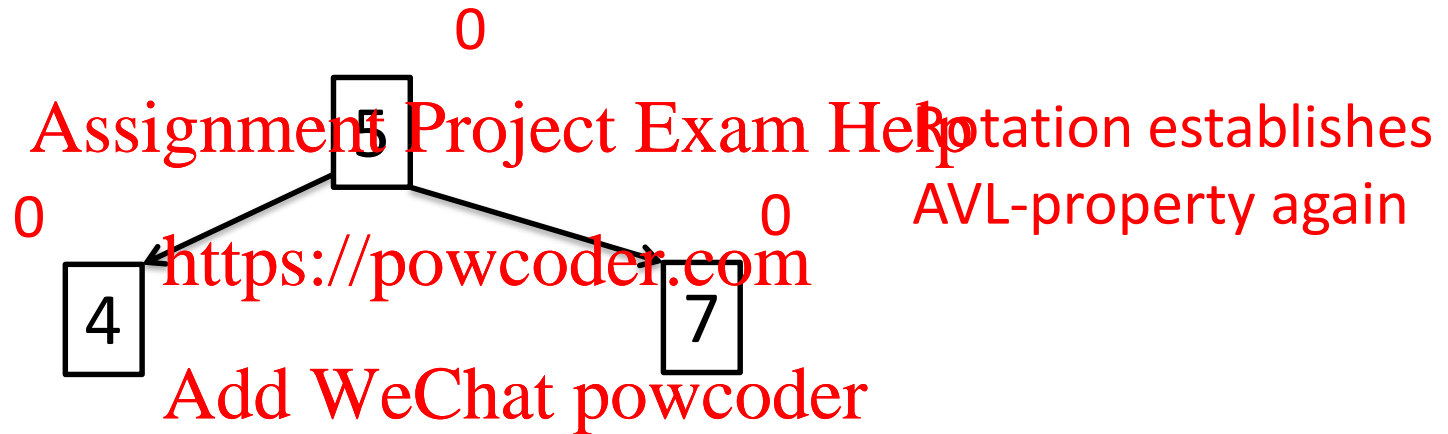
Insertion

Inserting a new element z can violate the AVL-property.

Consider path from the newly inserted leaf z to the root and repair AVL-property.

Rebalancing

Let z be the newly inserted leaf.

Consider the path from z to the root (reverse the insertion path).

Update the balance values.

Repair AVL-property (if necessary).

Insert

- we insert new node z as for Binary Search Trees.

- bal(z)=0 holds after insertion.

- bal(v) might change by 1 for a node v on the path from z to the root.

- If $b(v) \notin \{-1, -0, 1\}$ rebalance

Rebalancing

Start examining for v, where v is the parent of z, and continue with the parent of v (if necessary).

Assume that the right child x of node v is on the path from z to the root.

Before insertion -> After Insertion:

- bal(v) = 1 -> bal(v)=0 (height of tree rooted at v has not changed, stop rebalancing)
- bal(v)=0 -> bal(v) = -1 (height of tree rooted at v has increased by 1, stop rebalancing only if v is root, otherwise examine parent of v)
- bal(v)=-1 -> bal(v) = -2   (AVL-property violated, carry out rotation)

Left Rotation

Assume node v and right child x on the path.
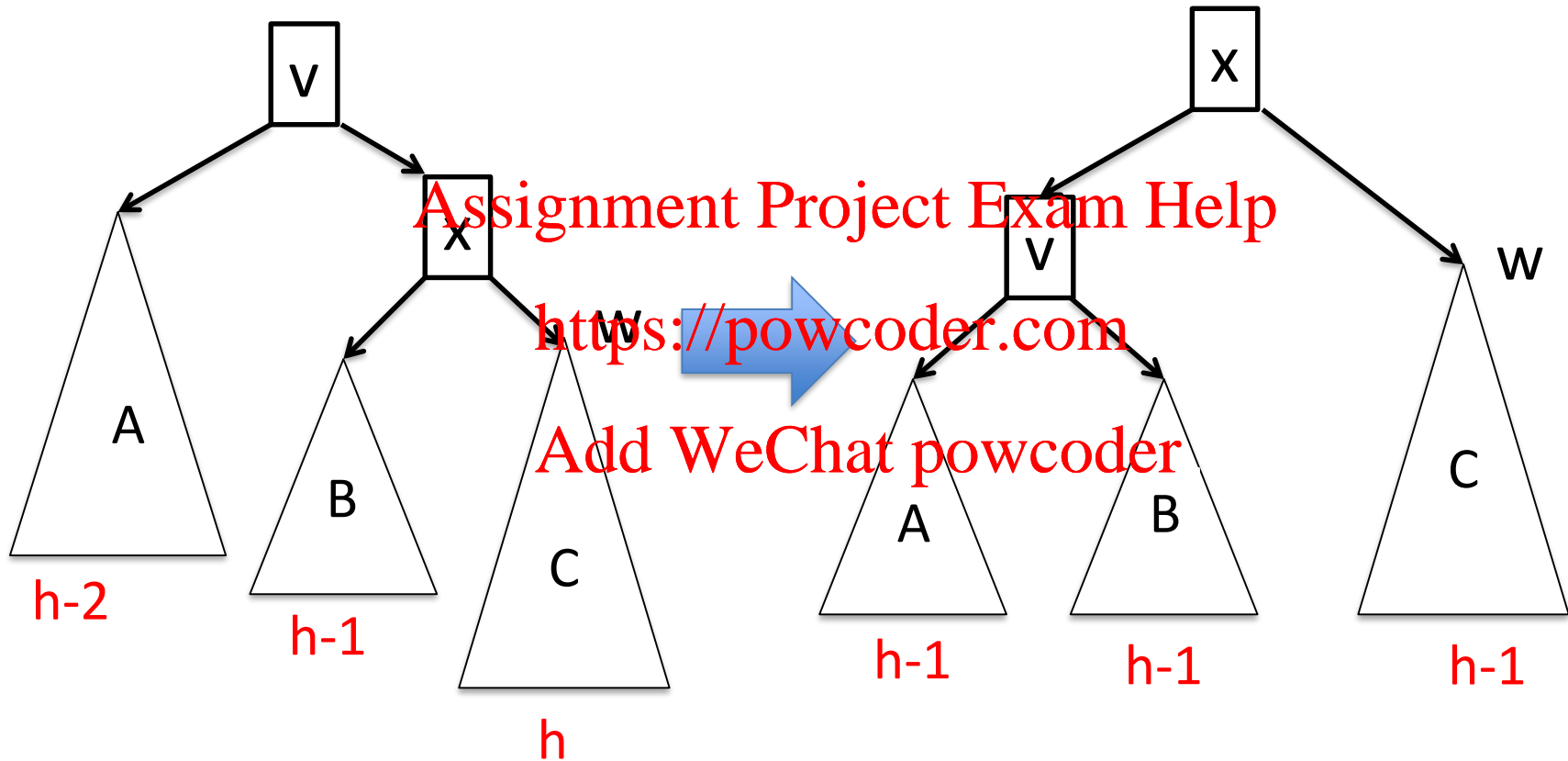
- w is right child of x on the path

    -> Left rotation

    New balance values: bal(x)=0 and bal(v)=0

Analogous: Right Rotation

Left Rotation

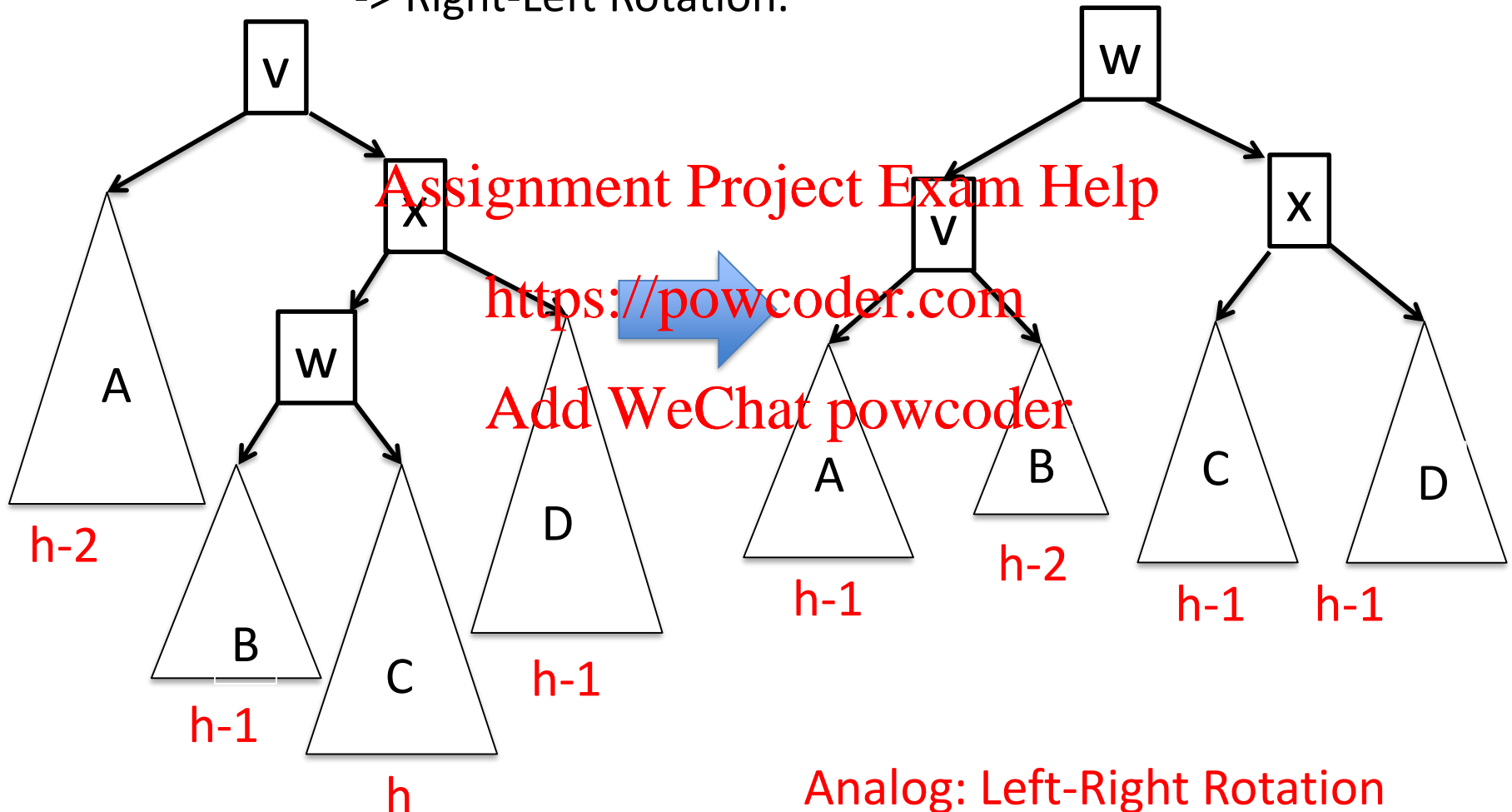Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Analog: Right Rotation

Right-Left Rotation

w is left child of x on the path
-> Right-Left Rotation.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Analog: Left-Right Rotation

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6
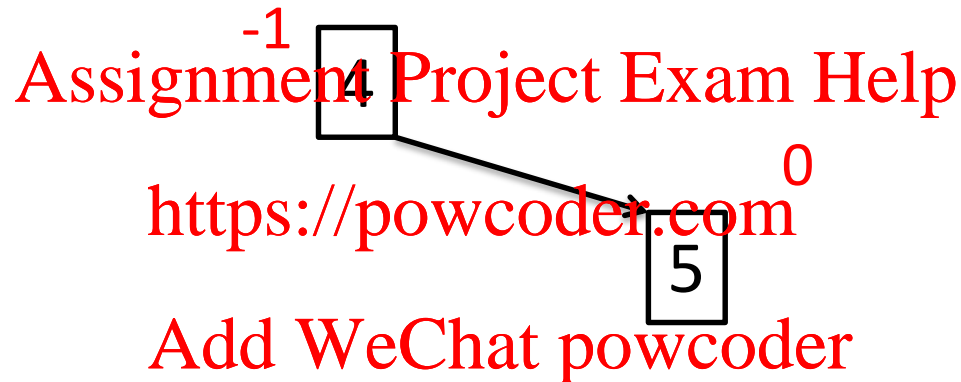
Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

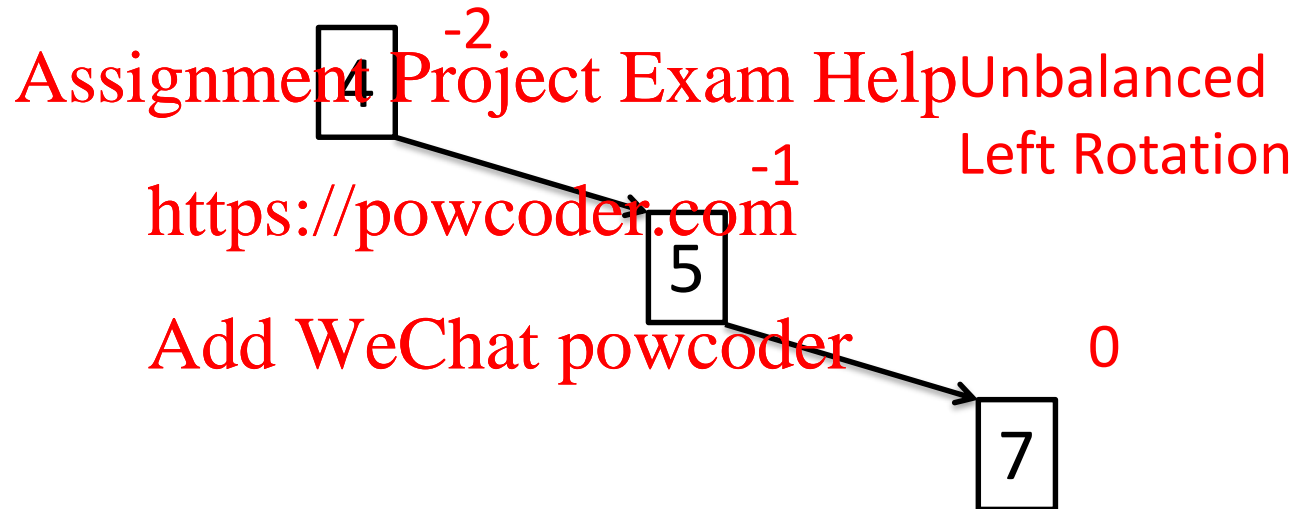# Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

0

4

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

-1

4

Assignment Project Exam Help

https://powcoder.com

0

5

Add WeChat powcoder

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

-2

4

Unbalanced

-1

Left Rotation

5

0

7

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

Assignment Project Exam Help

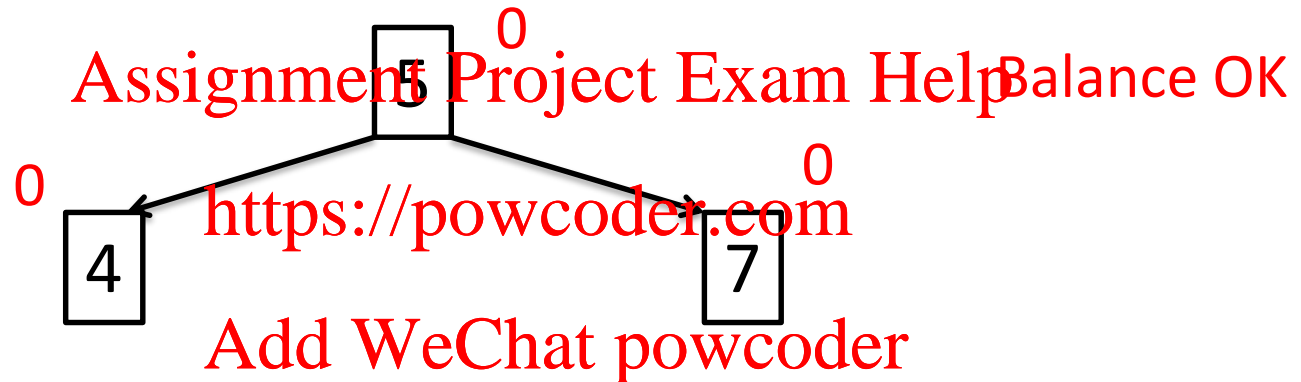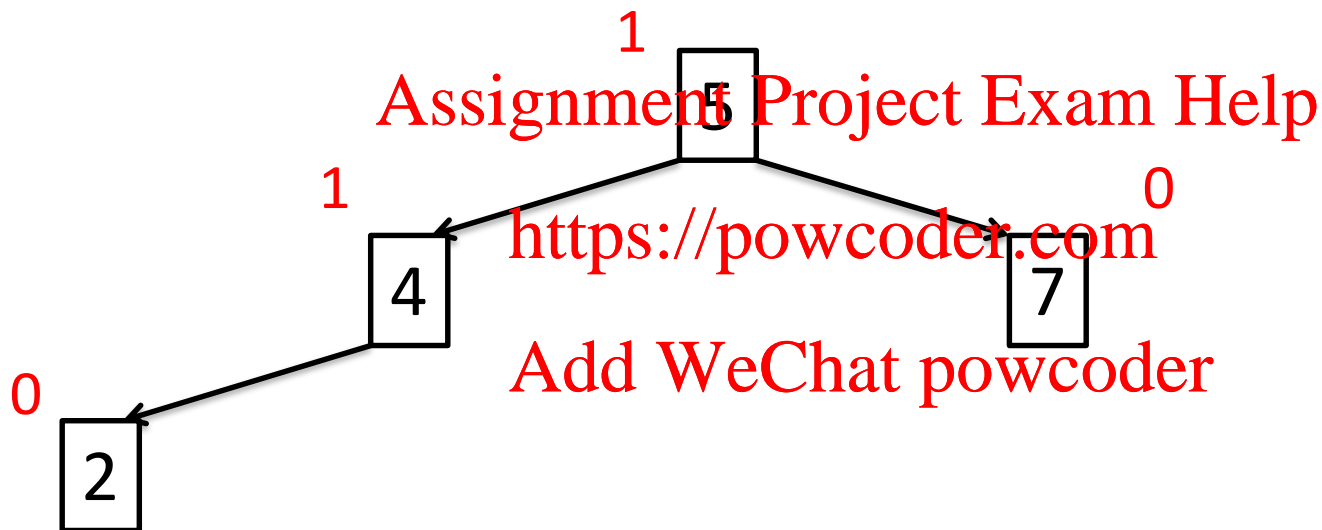Balance OK

https://powcoder.com

Add WeChat powcoder

0

5

0

4

0

7

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

Assignment Project Exam Help

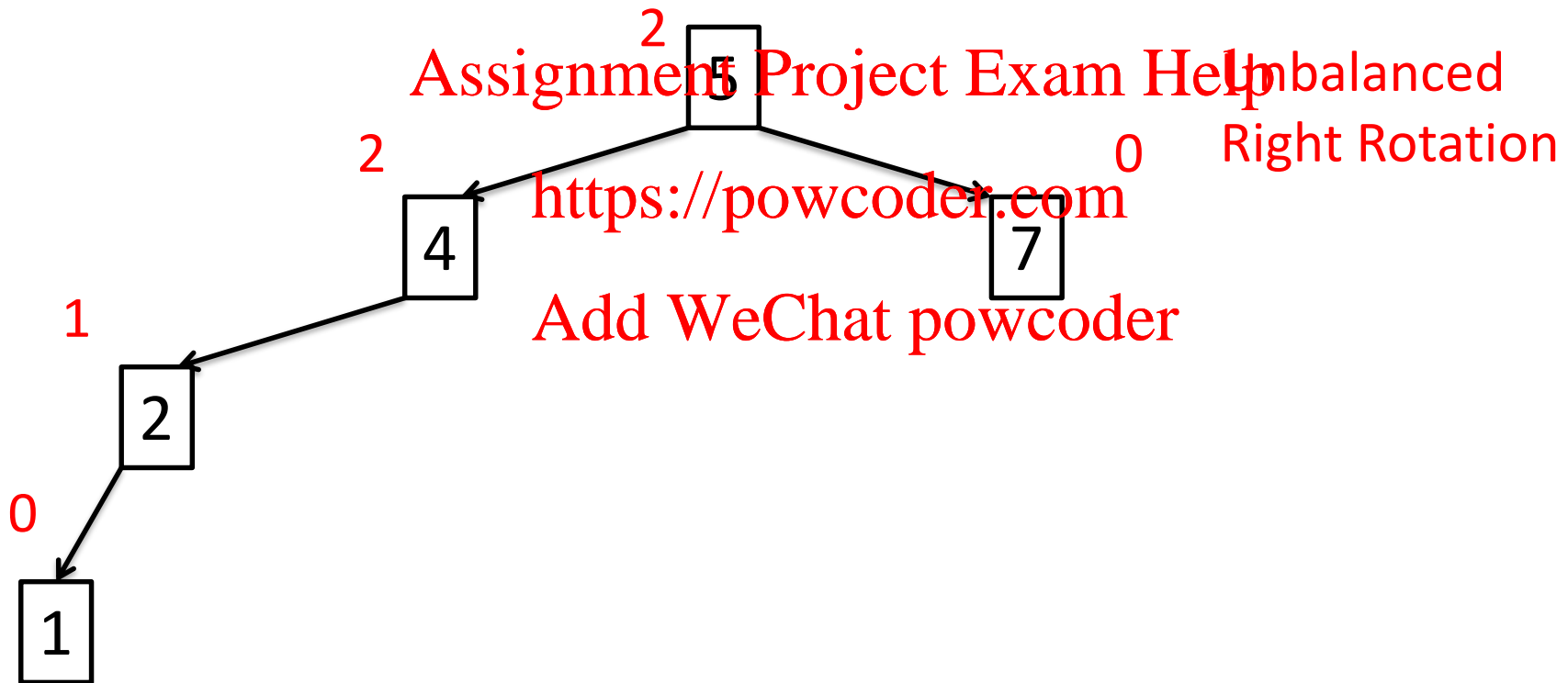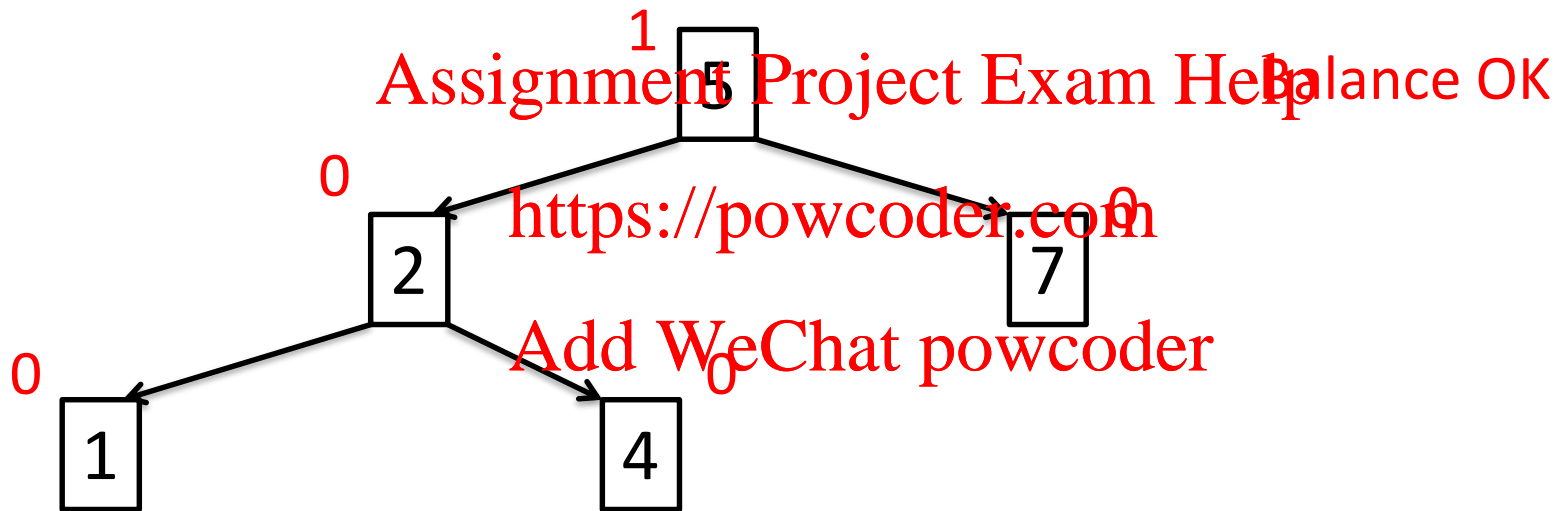https://powcoder.com
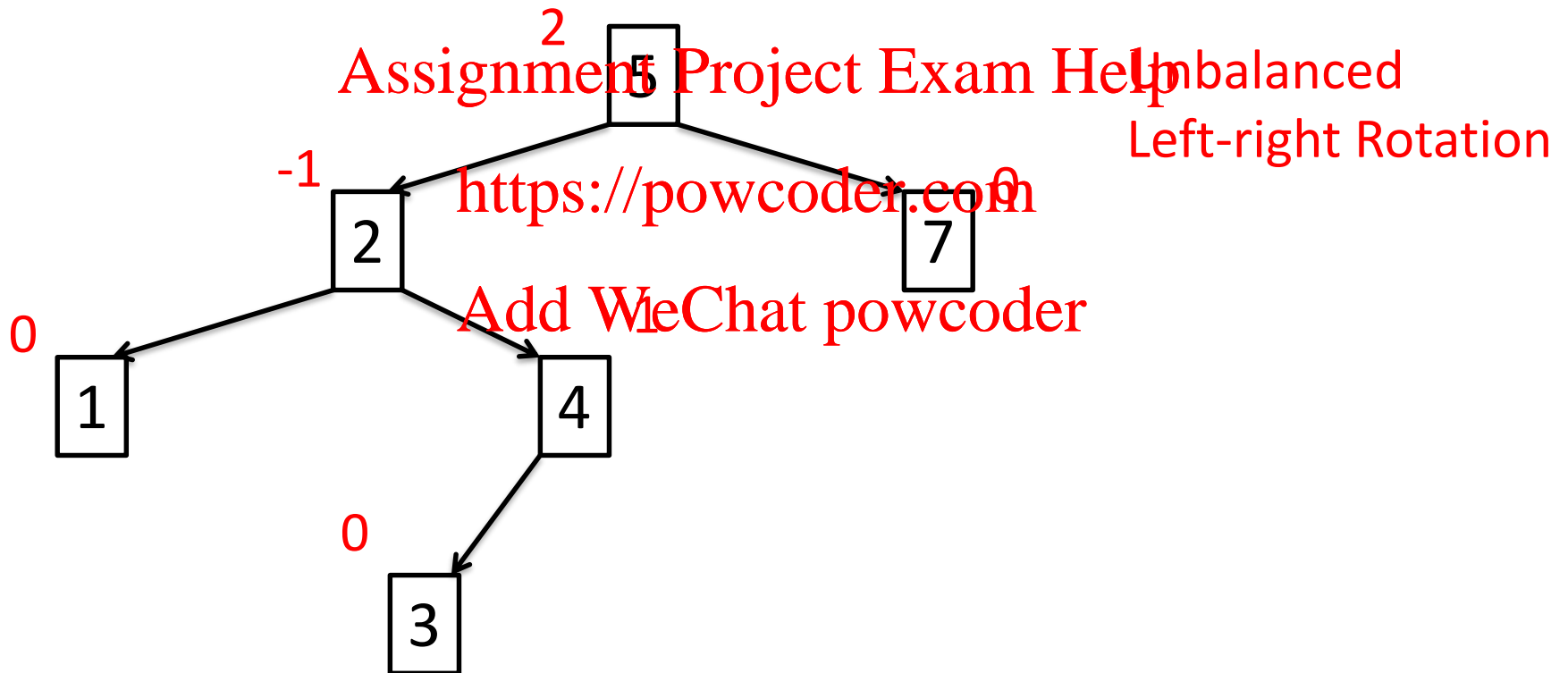
Add WeChat powcoder

Unbalanced
Right Rotation

Example Insert
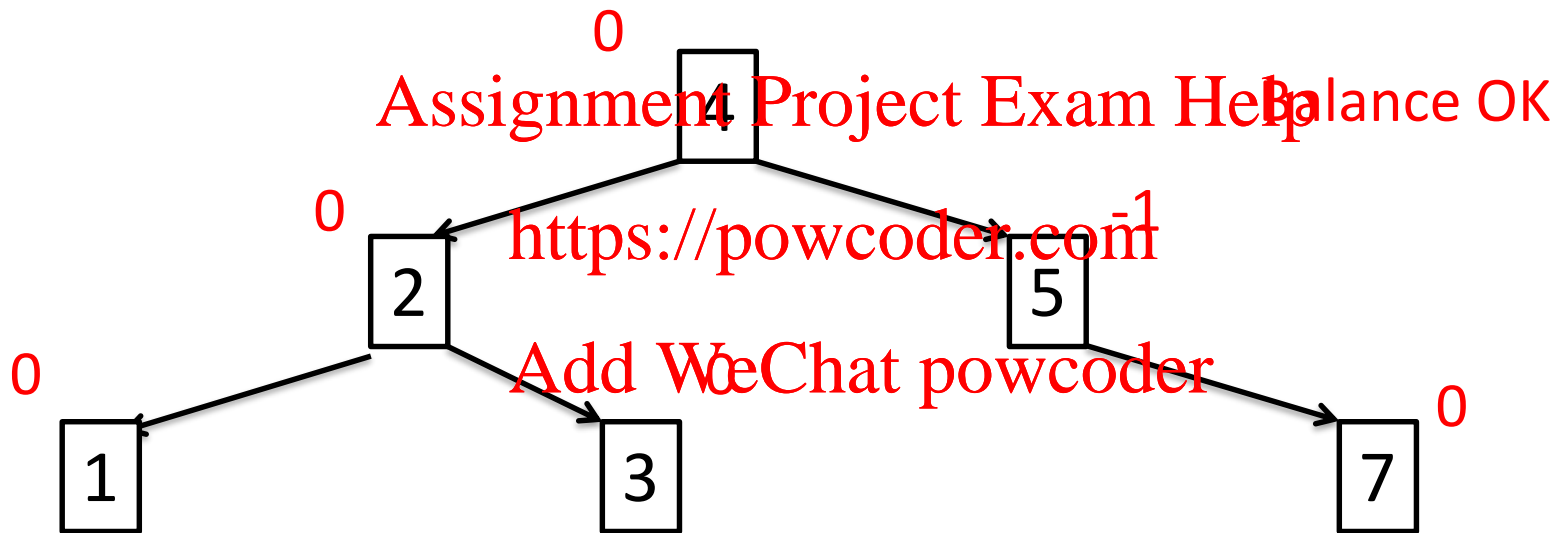
Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Unbalanced
Left-right Rotation

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

0

4   Balance OK

0

2   https://powcoder.com = 1

5

0

1   Add WeChat powcoder   3   0   7

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

-1

4

Assignment Project Exam Help

Unbalanced

Right-left Rotation

0

2

https://powcoder.com

-2

5

0

1

Add WeChat powcoder

3

1

7

0

6

Example Insert

Create AVL-Tree for sequence 4, 5, 7, 2, 1, 3, 6

0

4

Balance OK

0

2

0

6

0

1

0

3

0

5

0

7

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Time complexity of for insertion

Inserting a node involves:

- Finding the location and adding the node to the tree

- Moving up the AVL tree see whether AVL property is violated and perform rotation as needed

O(log n) + O (log n) + O(1)

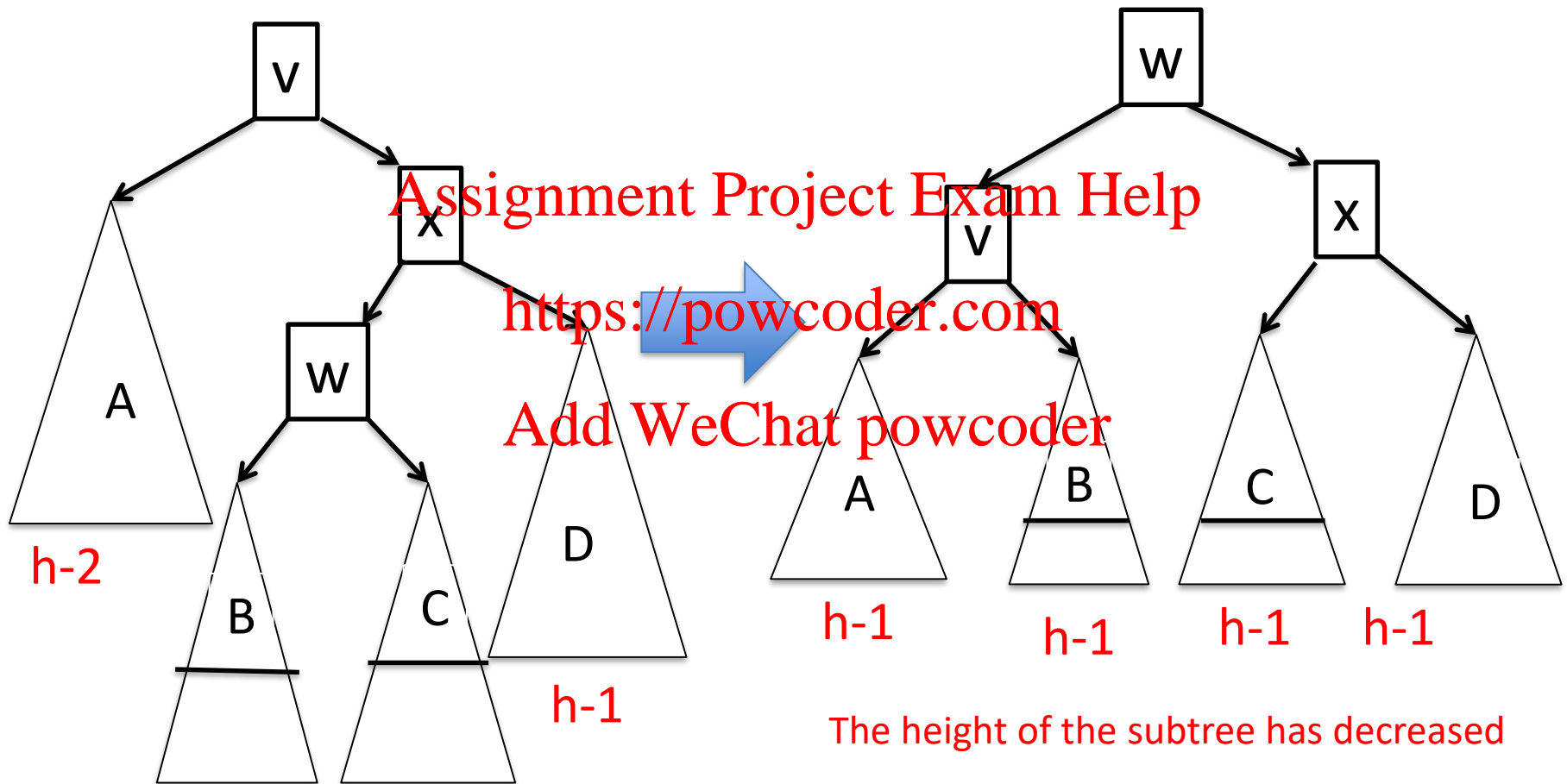Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Remove – Left Rotation

W.l.o.g. assume that the deleted node was in the left subtree of v and height of this tree has decrease by 1.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

h-2

B

C

h

h-1

h-1
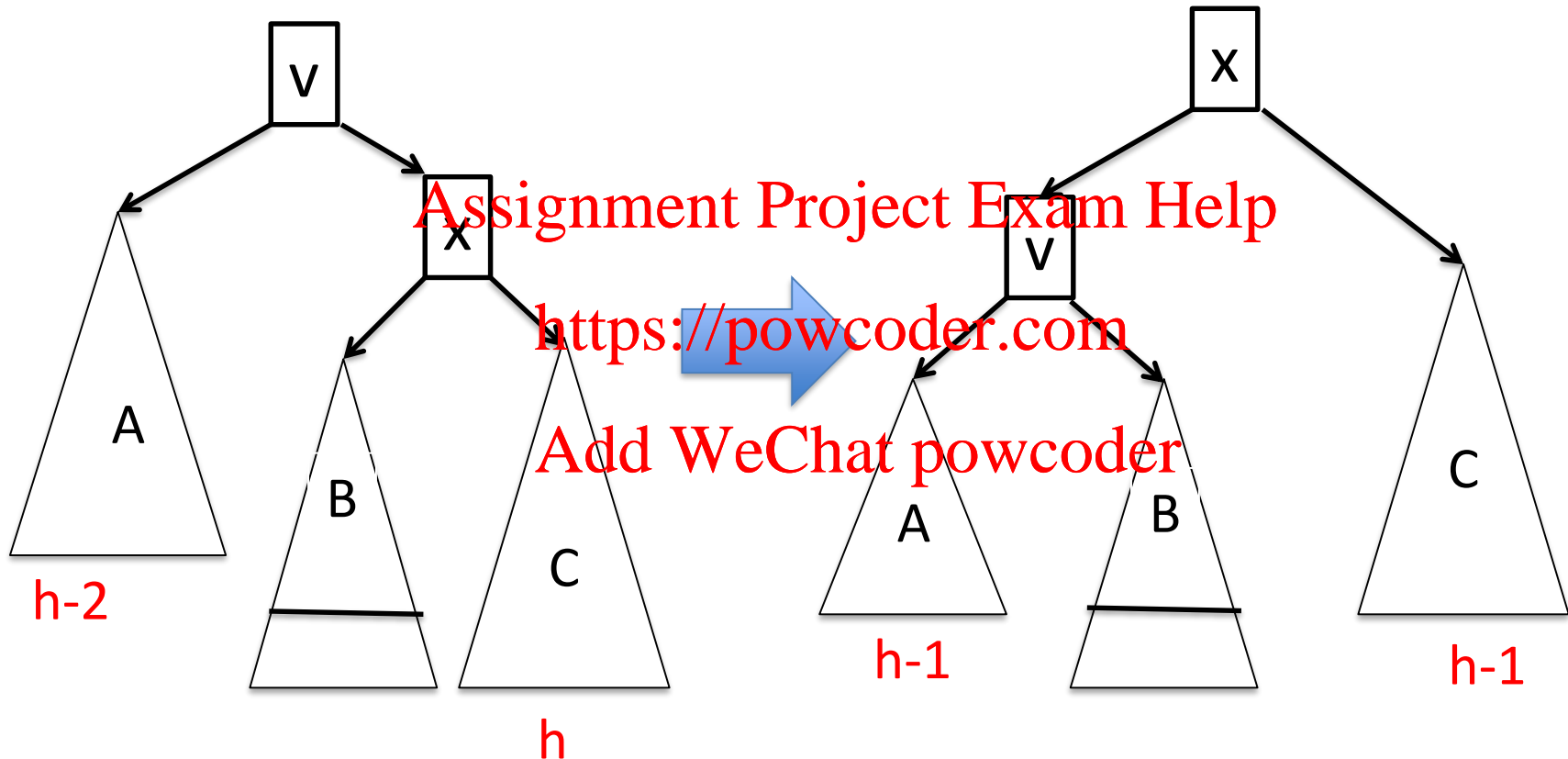
If B had height h-1 before deletion, the height of the subtree has decreased

Analog: Right Rotation

Right-Left Rotation

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

V → X

A: h-2
B
C
D: h-1
h

W → X

A: h-1
B: h-1
C: h-1
D: h-1

The height of the subtree has decreased

Either B or C might have height h-1

Analog: Left-Right Rotation

Rebalancing after Deletion

- After having rebalanced for node v the height of the tree previously rooted at v might have decreased after deleting and rebalancing.

- If this is the case old parent of v might be imbalanced.

- We might have to continue rebalancing until the root has been reached.

Runtime AVL-trees

Theorem: The operations find, insert, and delete can be implemented for AVL-trees in worst-case time O(log n).

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder