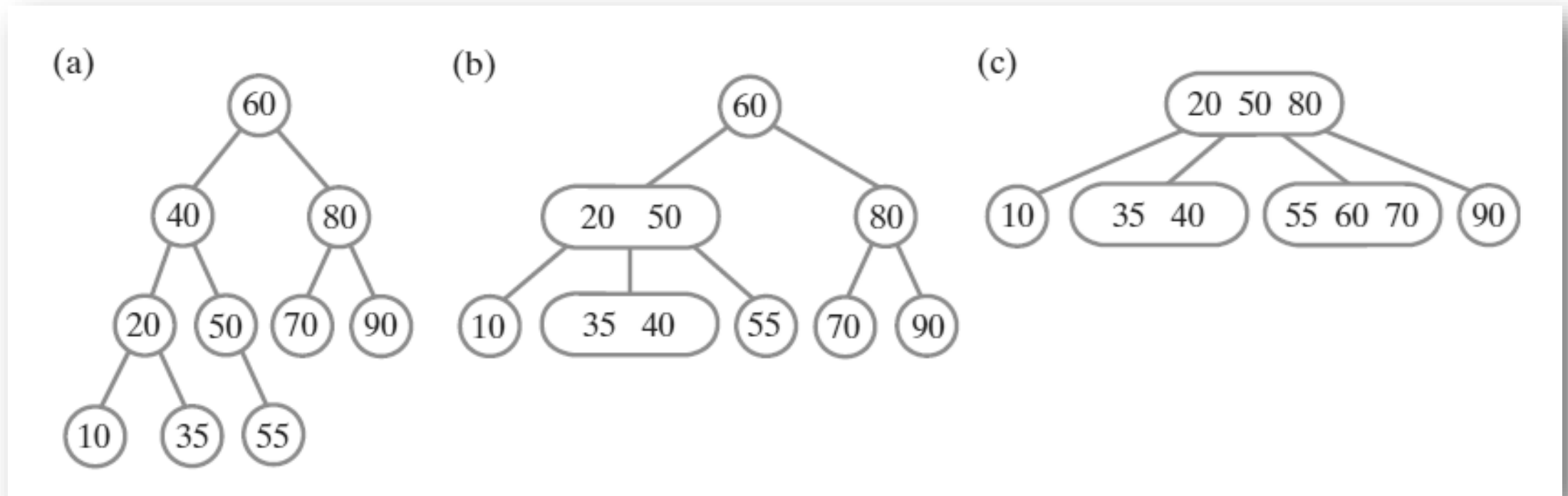


Comparing AVL, 2-3, and 2-4 Trees



Three balanced search trees obtained by adding 60, 50, 20, 80, 90, 70, 55, 10, 40, and 35: (a) AVL tree; (b) 2-3 tree; (c) 2-4 tree.

Question

- What 2-4 tree results when you make the following additions to an initially empty 2-4 tree?

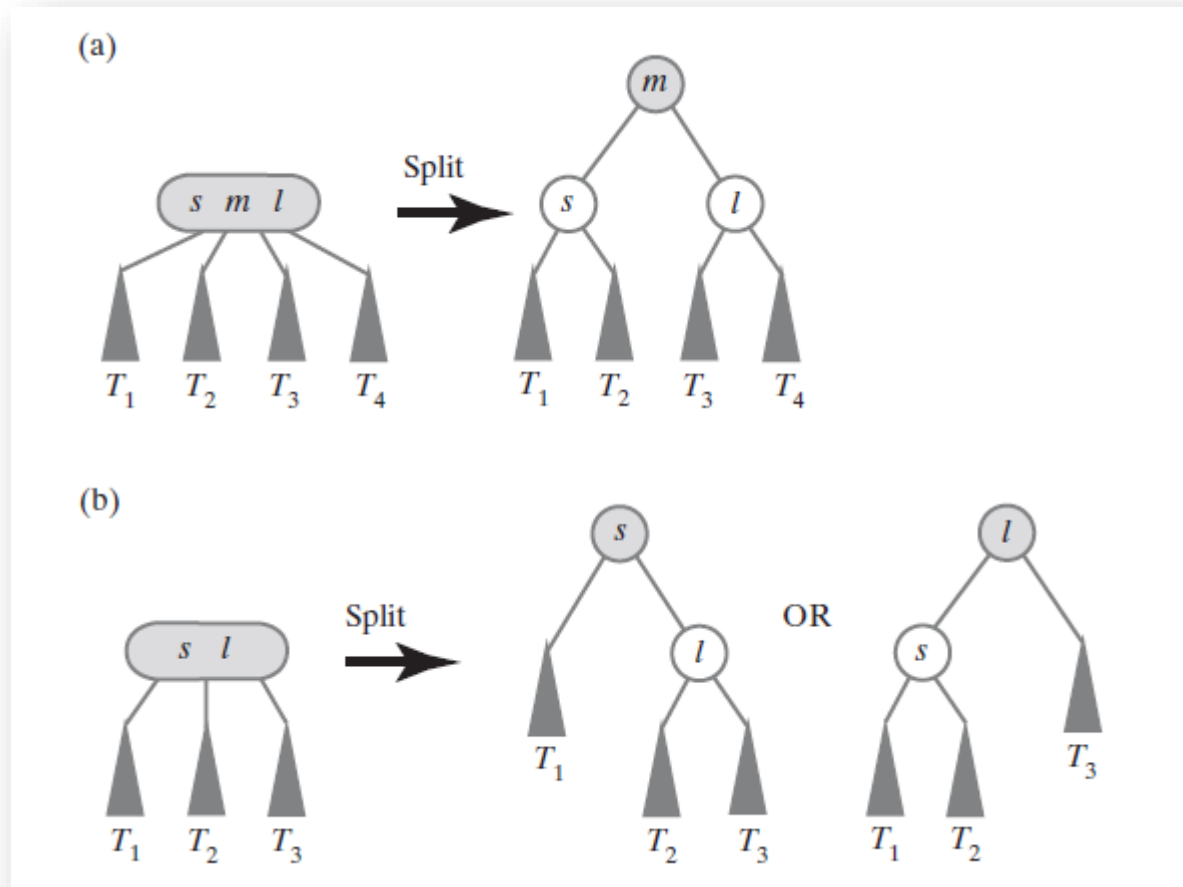
7, 8, 9, 2, 1, 5, 6, 4, 3

Red-Black Trees

Red-Black Trees

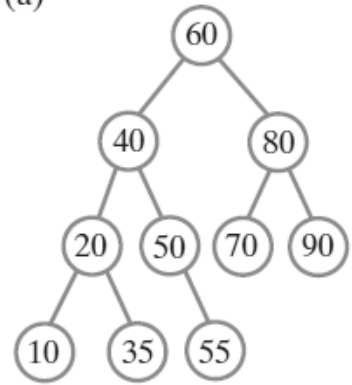
- A **red-black tree** is a binary tree that is equivalent to a 2-4 tree.
 - Conceptually, a red-black is more involved than a 2-4 tree, but its implementation uses only 2-nodes and so is easier to implement.
- Adding an entry to a red-black tree is like adding an entry to a 2-4 tree.

Red-Black Trees (cont.)

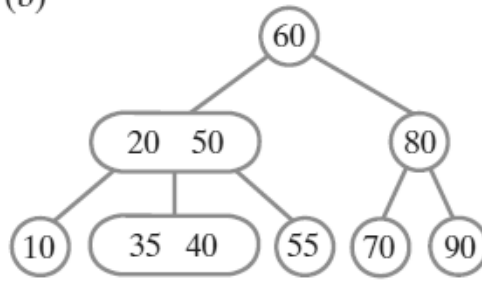


Using a 2-nodes to represent (a) a 4-node; (b) a 3-node.

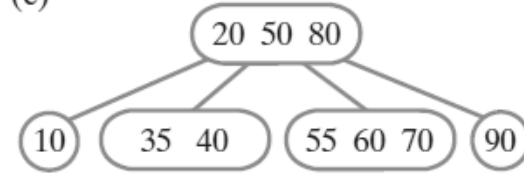
(a)



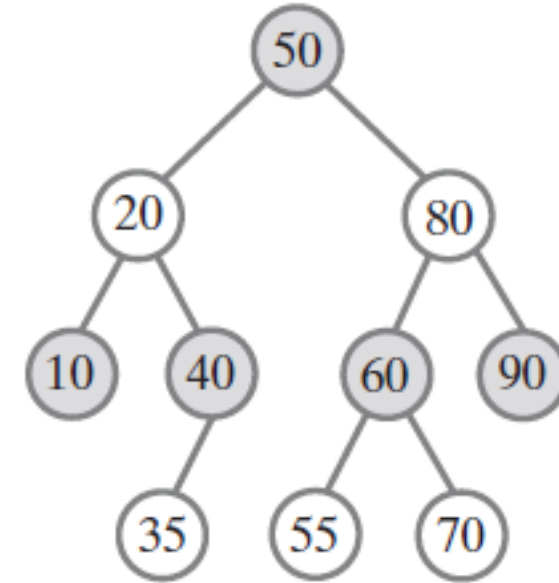
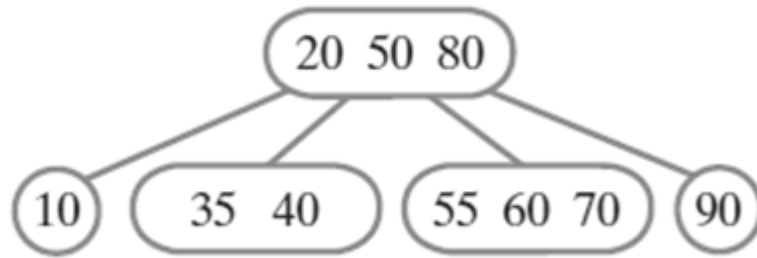
(b)



(c)



Three balanced search trees obtained by adding 60, 50, 20, 80, 90, 70, 55, 10, 40, and 35: (a) AVL tree; (b) 2-3 tree; (c) 2-4 tree.



A red-black tree, on right, that is equivalent to the 2-4 tree in left.

Properties of a red-black tree (Ref1)

1. The root is black.
2. Every red node has a black parent.
3. Any children of a red node are black; that is, a red node cannot have a red children.
4. Every path from the root to a leaf contains the same number of black nodes.

Properties of a red-black tree (Ref2)

A binary search tree is a red-black tree if it satisfies the following ***red-black properties***:

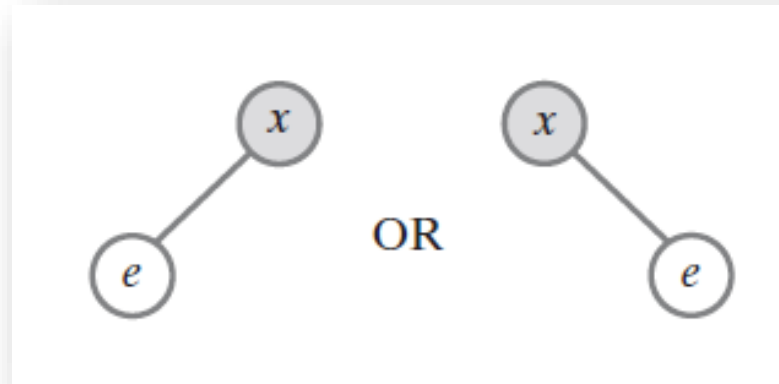
1. Every node is either red or black.
2. The root is black.
3. Every leaf (NIL) is black.
4. If a node is red, then both its children are black.
5. For each node, all paths from the node to descendant leaves contain the same number of black nodes.

Adding entries to a Red-Black Tree

- If we add a node to an empty red-black tree, the node must be black because it is the root.
- Adding an entry to a nonempty red-black tree results in a new red leaf.
 - The color of this leaf can change later when other entries are added or removed.

Adding entries to a Red-Black Tree:

Adding a leaf

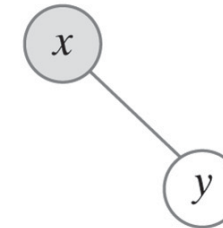


The result of adding a new entry e to a one-node red-black tree.

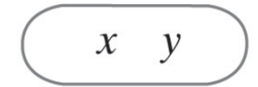
Adding entries to a Red-Black Tree: Adding a leaf (cont.)

(a) Before addition

Red-black tree



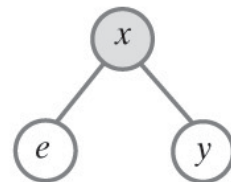
Equivalent 2-4 tree



© 2019 Pearson Education, Inc.

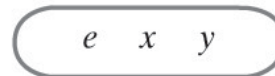
(b) Case 1:
The tree is balanced

**After adding e to
the red-black tree**

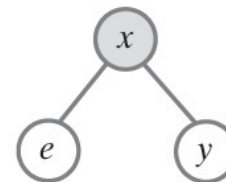


© 2019 Pearson Education, Inc.

**After adding e to
the 2-4 tree**



**Red-black equivalent
of the 2-4 tree**



**Action after addition
to transform
column 1 into column 3**

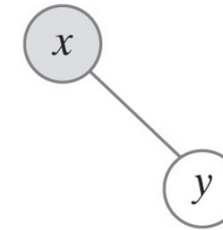
None

The possible results of adding a new entry e to a two-node red-black tree.

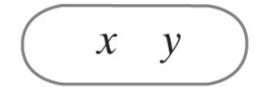
Adding entries to a Red-Black Tree: Adding a leaf (cont.)

(a) Before addition

Red-black tree

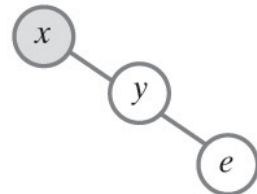


Equivalent 2-4 tree

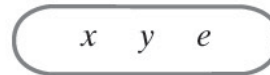


© 2019 Pearson Education, Inc.

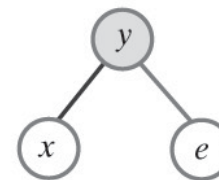
**After adding e to
the red-black tree**



**After adding e to
the 2-4 tree**



**Red-black equivalent
of the 2-4 tree**



**Action after addition
to transform
column 1 into column 3**

Single left rotation
and color flip

(c) Case 2:
A red node has a
red right child

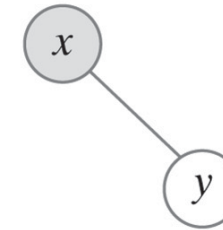
© 2019 Pearson Education, Inc.

The possible results of adding a new entry e to a two-node red-black tree.

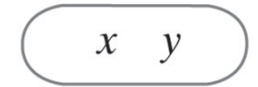
Adding entries to a Red-Black Tree: Adding a leaf (cont.)

(a) Before addition

Red-black tree

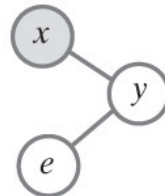


Equivalent 2-4 tree

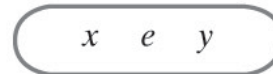


© 2019 Pearson Education, Inc.

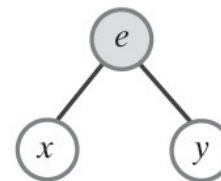
**After adding e to
the red-black tree**



**After adding e to
the 2-4 tree**



**Red-black equivalent
of the 2-4 tree**



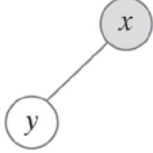
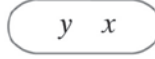
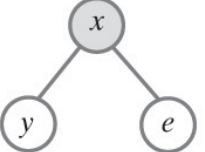
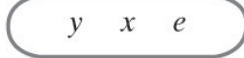
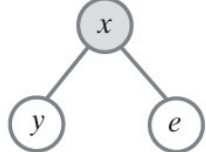
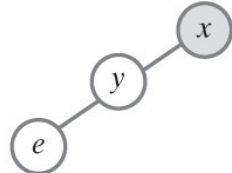
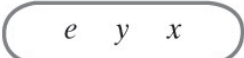
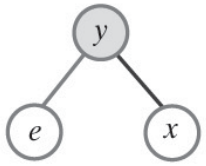
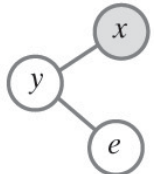
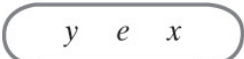
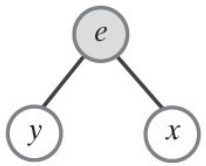
**Action after addition
to transform
column 1 into column 3**

Right-left double
rotation and color flip

(d) Case 3:
A red node has a
red left child

© 2019 Pearson Education, Inc.

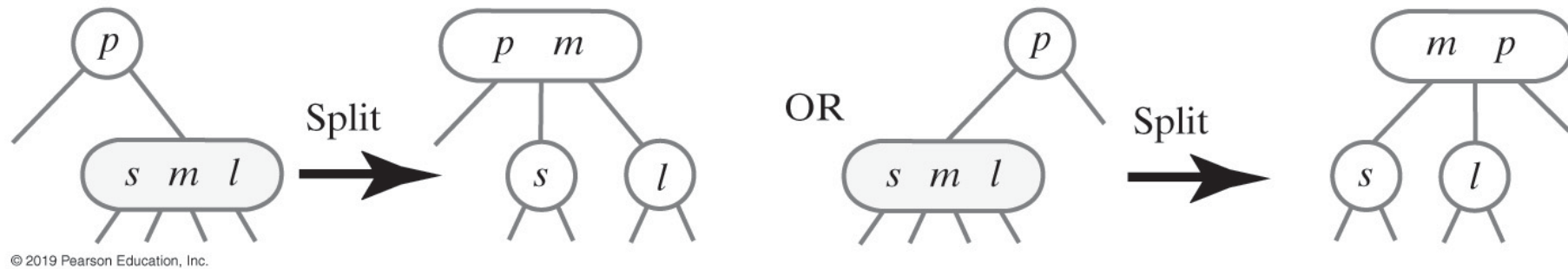
The possible results of adding a new entry e to a two-node red-black tree.

	Red-black tree	Equivalent 2-4 tree		
(a) Before addition				
	<small>© 2019 Pearson Education, Inc.</small>			
	After adding <i>e</i> to the red-black tree	After adding <i>e</i> to the 2-4 tree	Red-black equivalent of the 2-4 tree	Action after addition to transform column 1 into column 3
(b) Case 1: The tree is balanced				None
	<small>© 2019 Pearson Education, Inc.</small>			
(c) Case 2: A red node has a red left child				Single right rotation and color flip
	<small>© 2019 Pearson Education, Inc.</small>			
(d) Case 3: A red node has a red right child				Left-right double rotation and color flip
	<small>© 2019 Pearson Education, Inc.</small>			

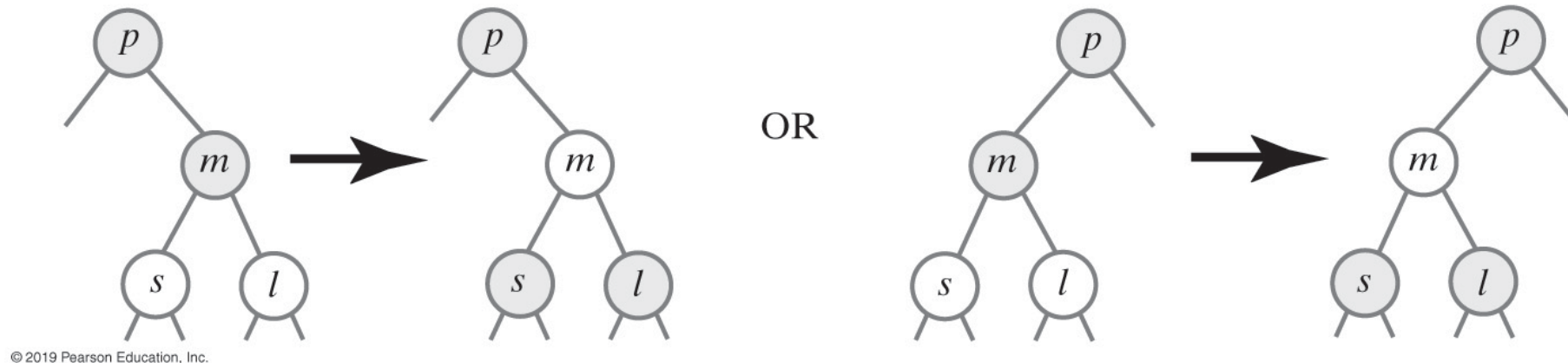
The possible results of adding a new entry *e* to a two-node red-black tree: mirror images of figures shown on previous three slides.

Adding entries to a Red-Black Tree: Splitting a 4-node whose parent is black

(a) In a 2-4 tree



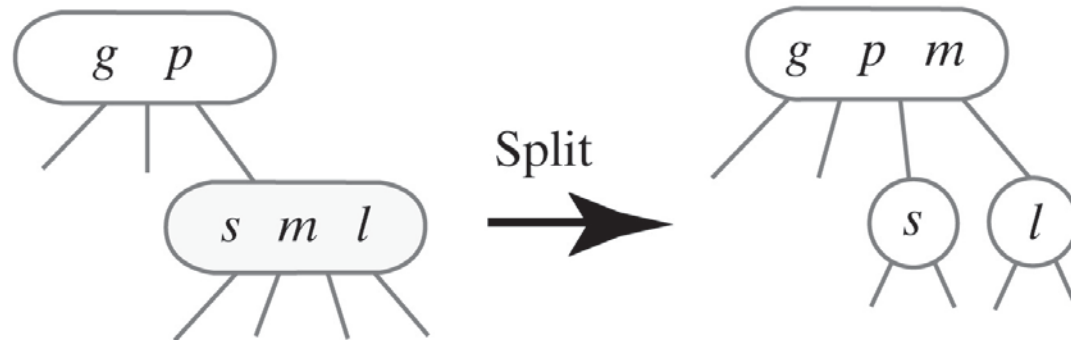
(b) In a red-black tree



Splitting a 4-node whose parent is a 2-node.

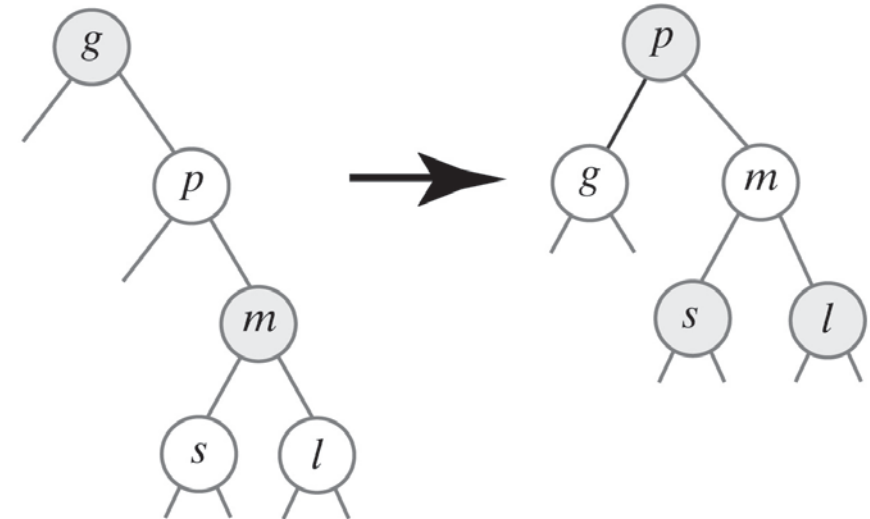
Adding entries to a Red-Black Tree: Splitting a 4-node whose parent is red: Case 1

(a) In a 2-4 tree



© 2019 Pearson Education, Inc.

(b) In a red-black tree

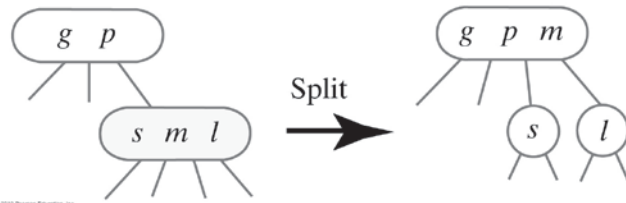


© 2019 Pearson Education, Inc.

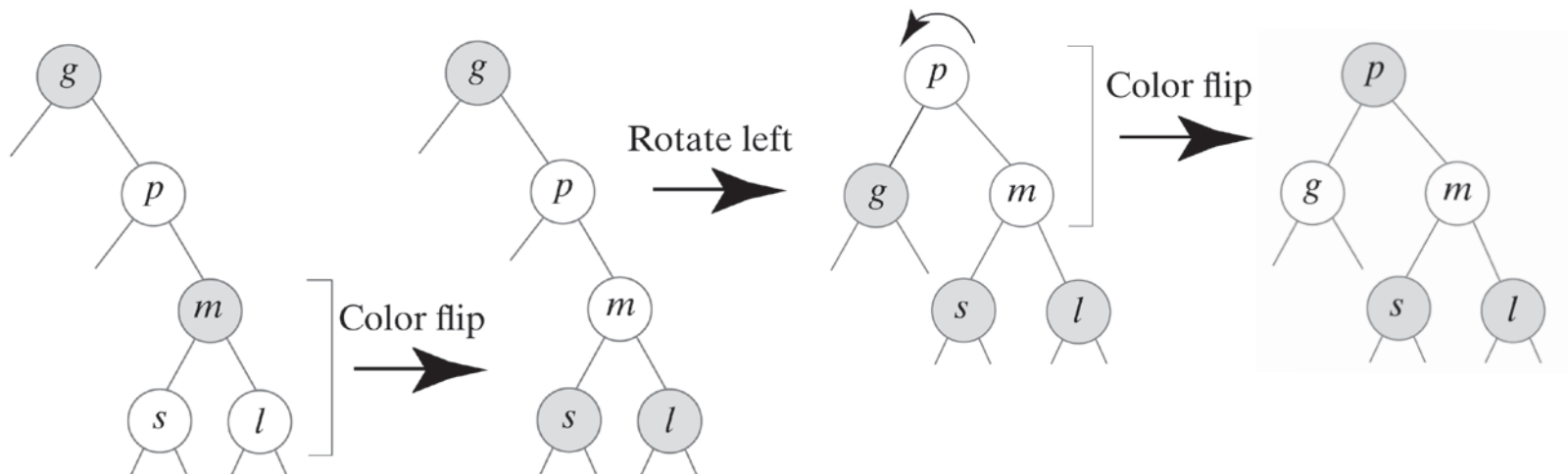
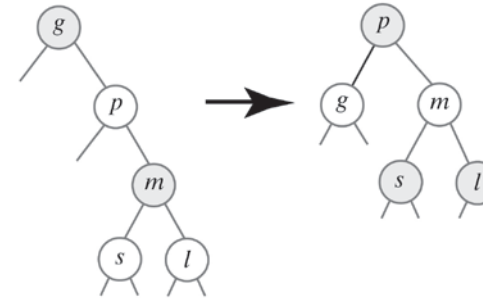
Splitting a 4-node whose parent is a 3-node.

Adding entries to a Red-Black Tree: Splitting a 4-node whose parent is red: Case 1

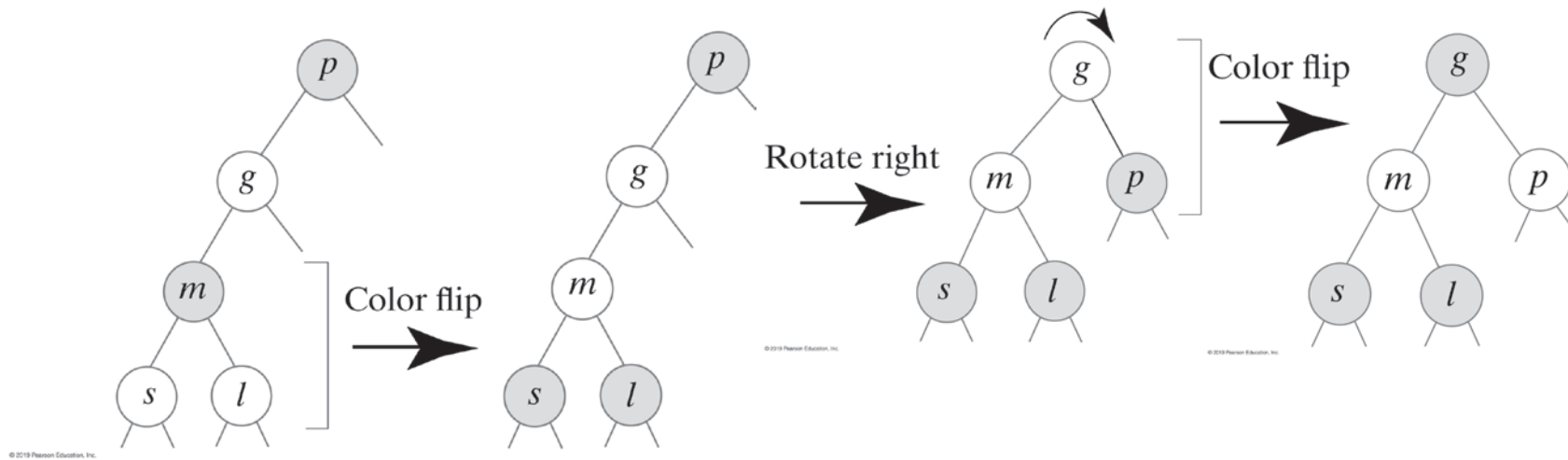
(a) In a 2-4 tree



(b) In a red-black tree



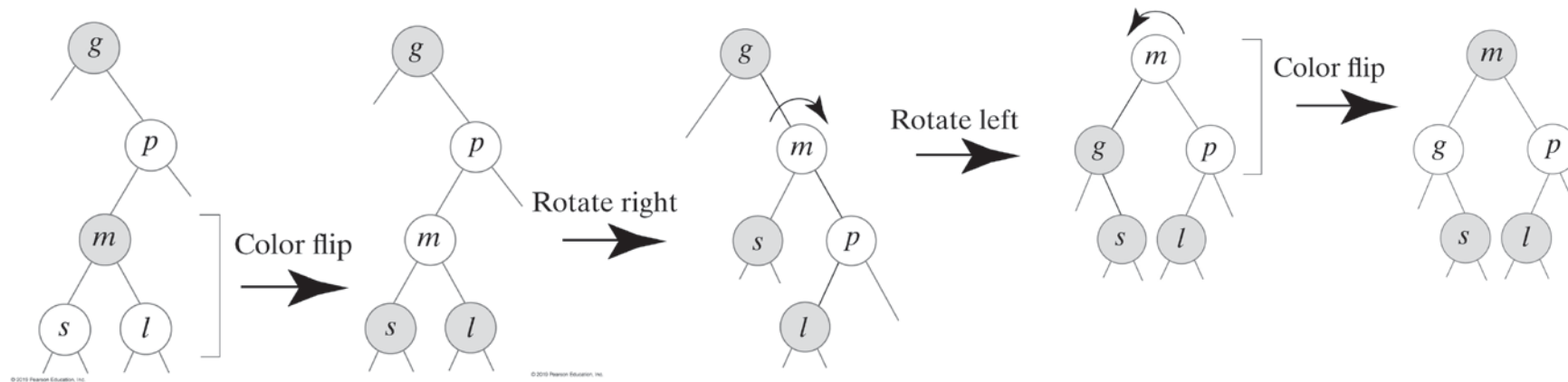
Adding entries to a Red-Black Tree: Splitting a 4-node whose parent is red: Case 2 (a 4-node is a left child of its parent)



Adding entries to a Red-Black Tree:

Splitting a 4-node whose parent is red: Case 3

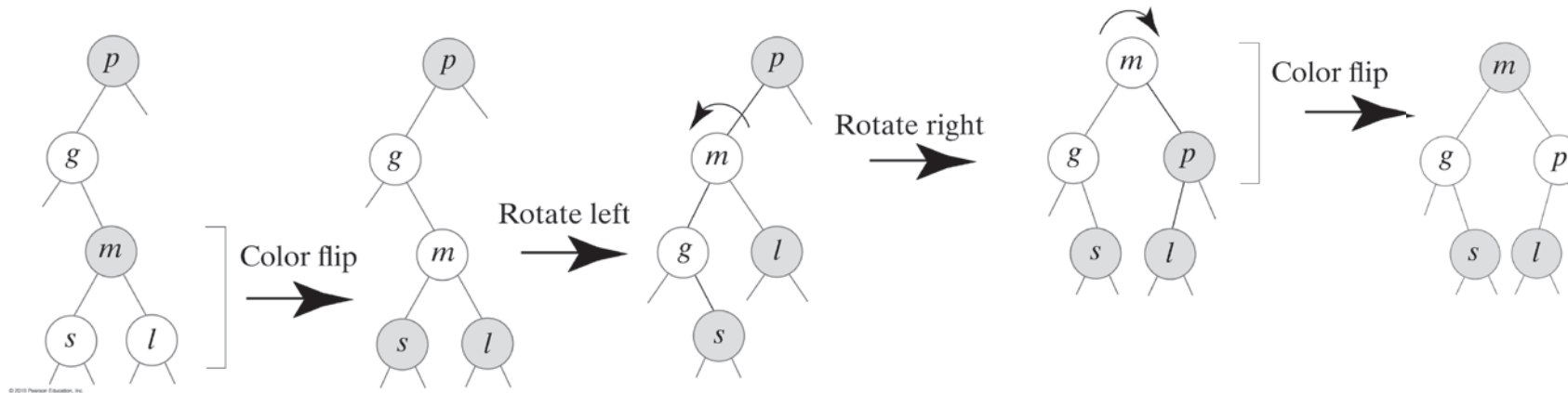
(a 4-node is a middle child of its 3-node parent)



Adding entries to a Red-Black Tree:

Splitting a 4-node whose parent is red: Case 4

(a 4-node is a middle child of its 3-node parent)



References

- F. M. Carrano & T. M. Henry, “Data Structures and Abstractions with Java”, 4th ed., 2015. Pearson Education, Inc.
- T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “Introduction to Algorithms”, 2nd ed., 2001. The Massachusetts Institute of Technology Press.