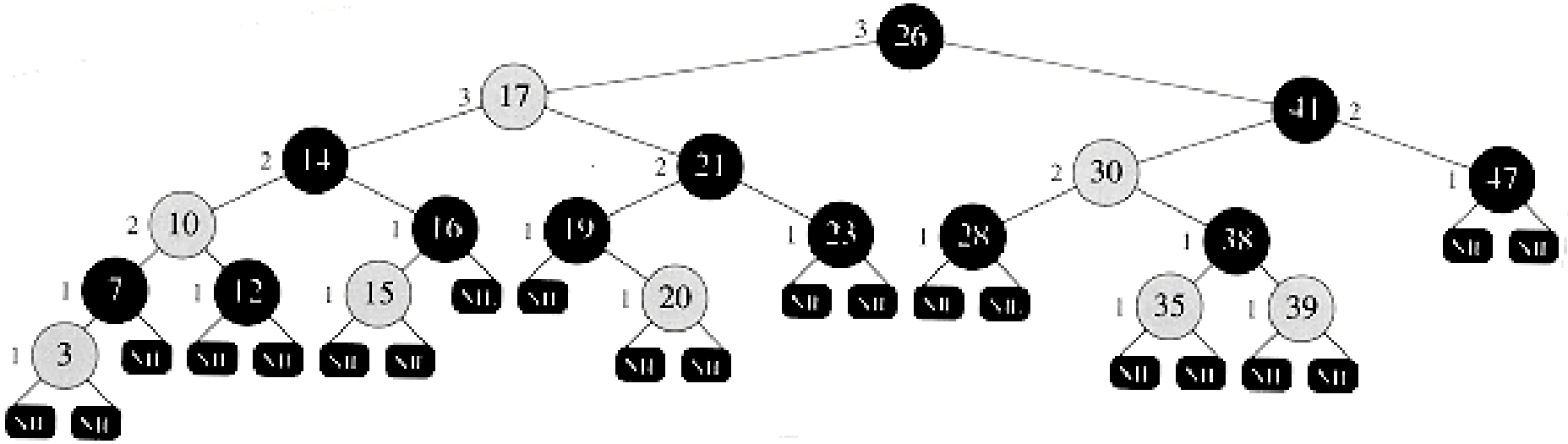
Red-black tree



Cormen

Red-Black tree

A red-black tree is a binary search tree which satisfies:

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. Every path from a node to a descendant leaf contains the same number of black nodes.

(Red-black trees are not for the exam)

Red-Black tree

- **black-height** of a node x: bh(*x*)

  the number of black nodes on any path from *x* to a leaf node
- **black-height of a red-black tree**: the black-height of its root.

Lemma

A red-black tree with *n* internal nodes

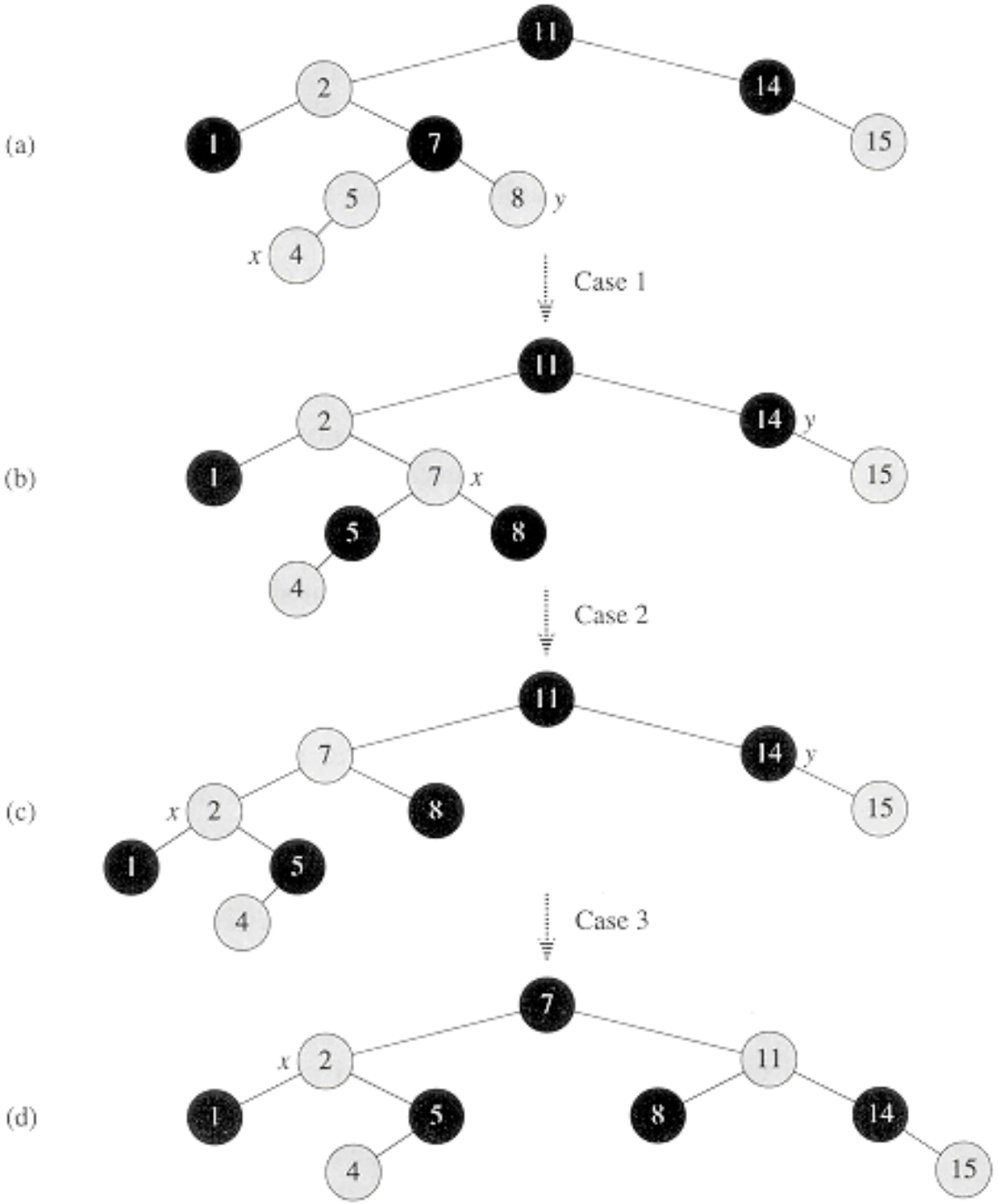
has height at most $2*\log_2(n + 1)$.

Red-black tree: operation insert

- insert in BSTree
- the new node is red
- if the parent of the new node is red

  fix the tree !!

Red-black tree: operation insert



Red-black tree: operation delete

Delete as in BSTree

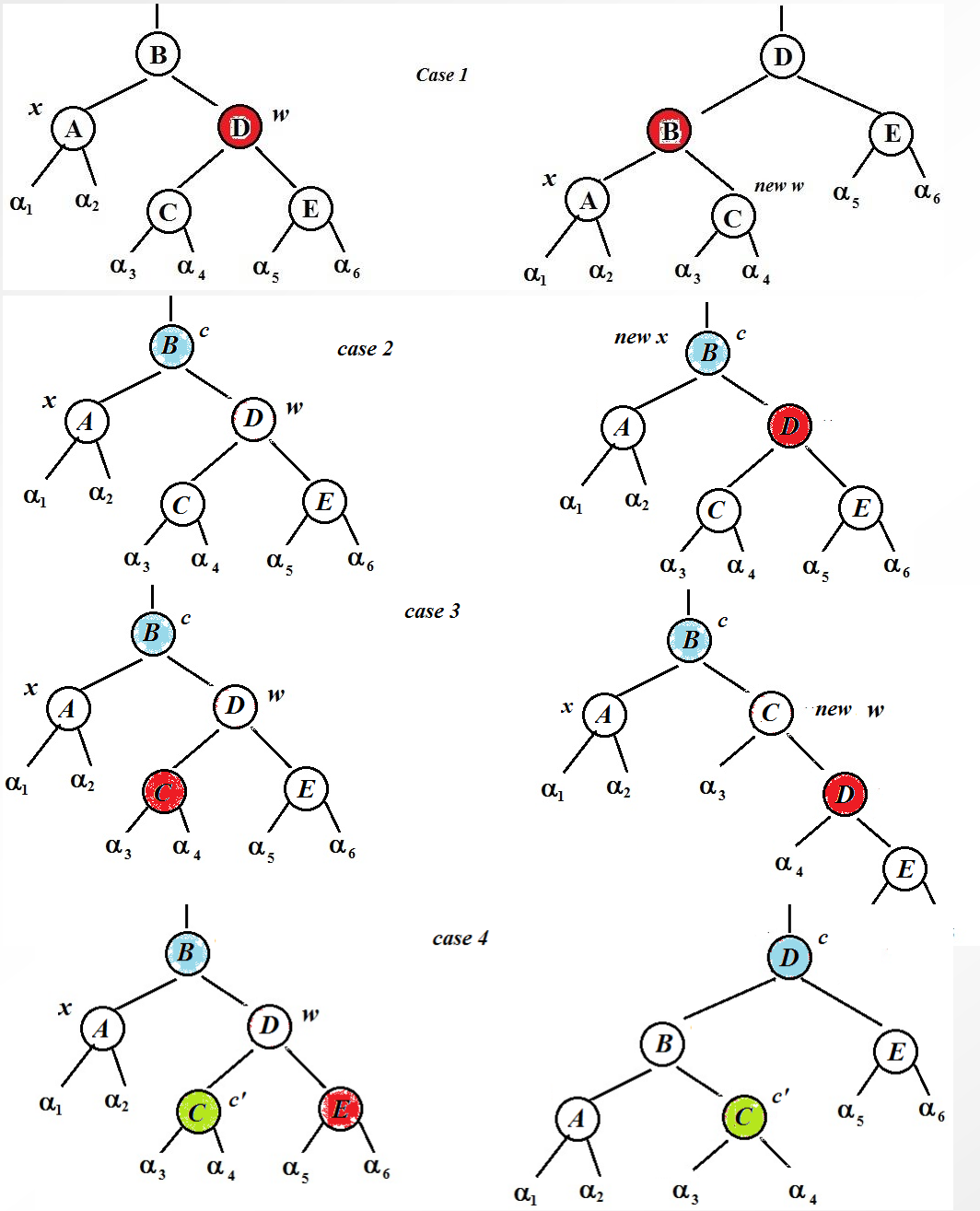
If a discrepancy arises for the red-black tree, fix it !

- If the deleted node is red

  the tree is still a red-black tree

- If the deleted node is black:
  - if its child is red, repaint the child to black.
  - otherwise: fix the tree !!

    mark the child as **double black : x** **(and fix the problem !)**


Case 1
B
x
A
D
w
α1
α2
C
E
α3
α4
α5
α6
D
B
E
x
A
C
new w
α5
α6
α1
α2
α3
α4
case 2
B
c
x
A
D
w
α1
α2
C
E
α3
α4
α5
α6
new x
B
c
A
D
α1
α2
C
E
α3
α4
α5
α6
case 3
B
c
x
A
D
w
α1
α2
C
E
α3
α4
α5
α6
B
c
x
A
C
new w
α1
α2
α3
D
α4
E
case 4
B
x
A
D
w
α1
α2
C
c′
E
α3
α4
α5
α6
D
c
B
E
A
C
c′
α5
α6
α1
α2
α3
α4