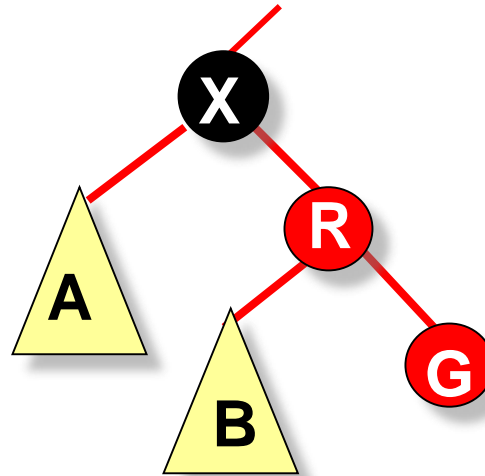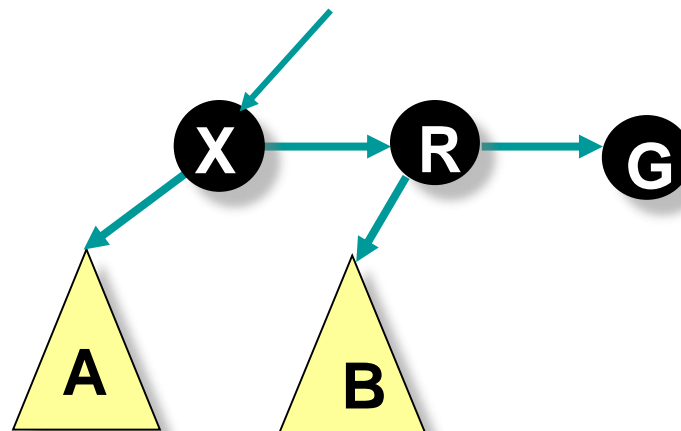# Adjustments to AA-Trees: Split
(Color no longer shown for AA-Trees, since only the level is stored)

Problem: With G inserted, there are two reds in a row

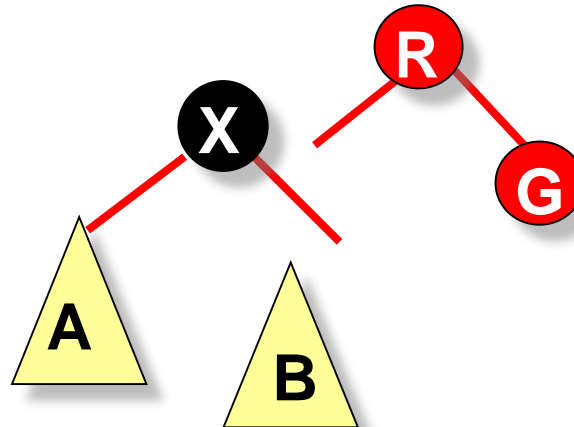Red-Black Tree

The split（左旋） procedure is a simple left rotation between X and R

AA-Tree
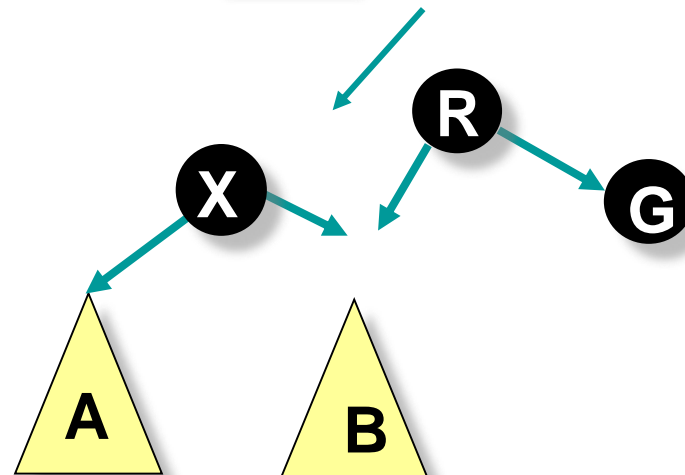
Adjustments to AA-Trees: Split

Problem: With
  G inserted,
  there are two
  reds in a row

The split（左旋）
  procedure is
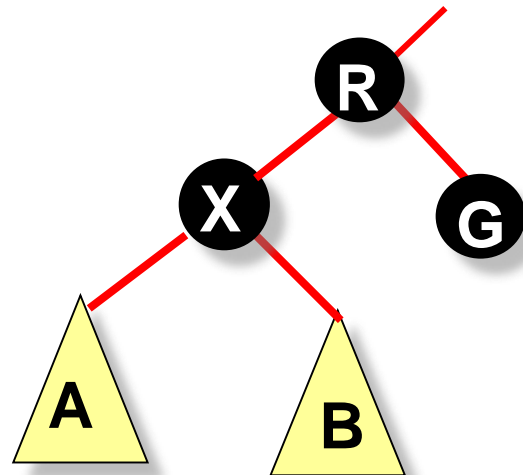 a simple left
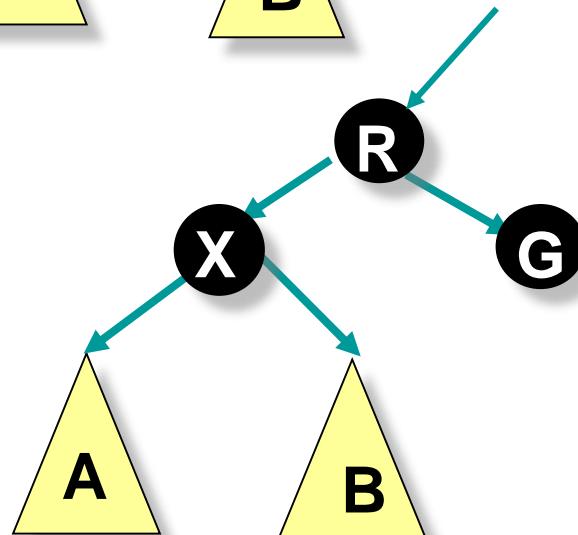 rotation
  between X
  and R

Red-Black
Tree

AA-Tree

Adjustments to AA-Trees: Split

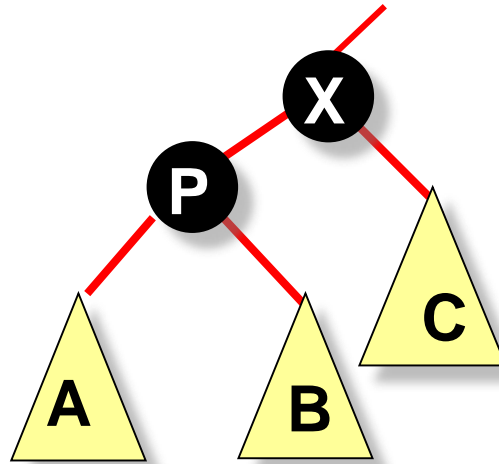Red-Black Tree

Note that R's level increases in the AA-Tree
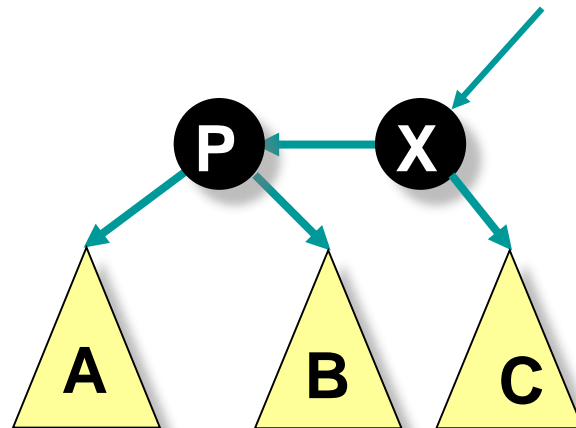
AA-Tree

Adjustments to AA-Trees: Skew

Problem:
Horizontal left link in AA-Tree (too many black nodes on one path)

Red-Black Tree

The skew（右旋）procedure is a simple right rotation between X and P

AA-Tree

Adjustments to AA-Trees: Skew

Problem:
  Horizontal left link in AA-Tree (too many black nodes on one path)

Red-Black Tree

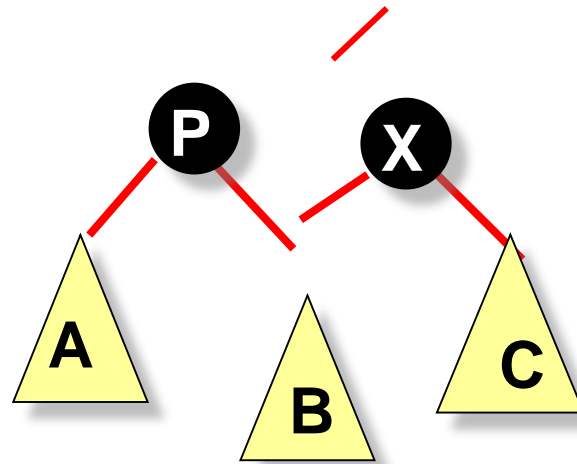The skew（右旋）procedure is a simple right rotation between X and P
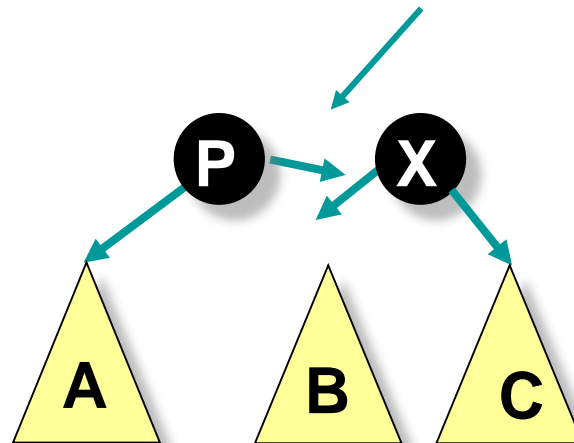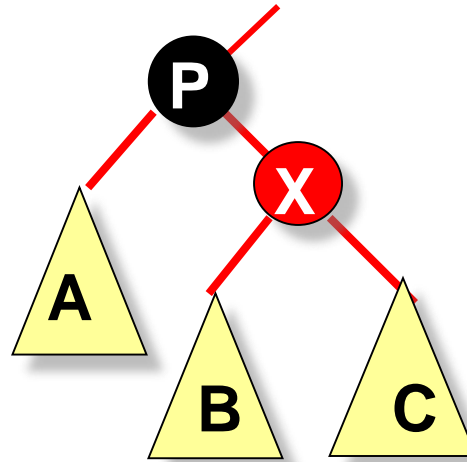
AA-Tree

Adjustments to AA-Trees: Skew
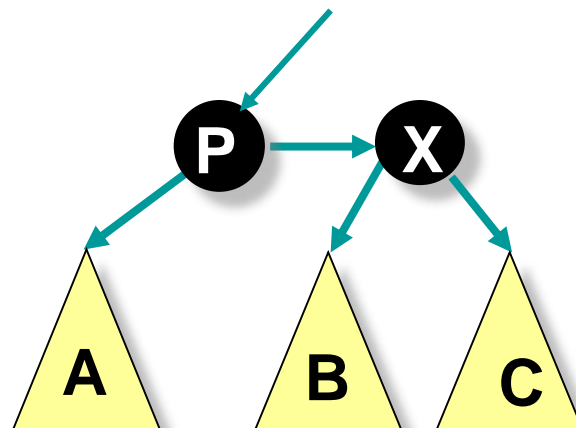
Problem:
Horizontal left
link in AA-Tree
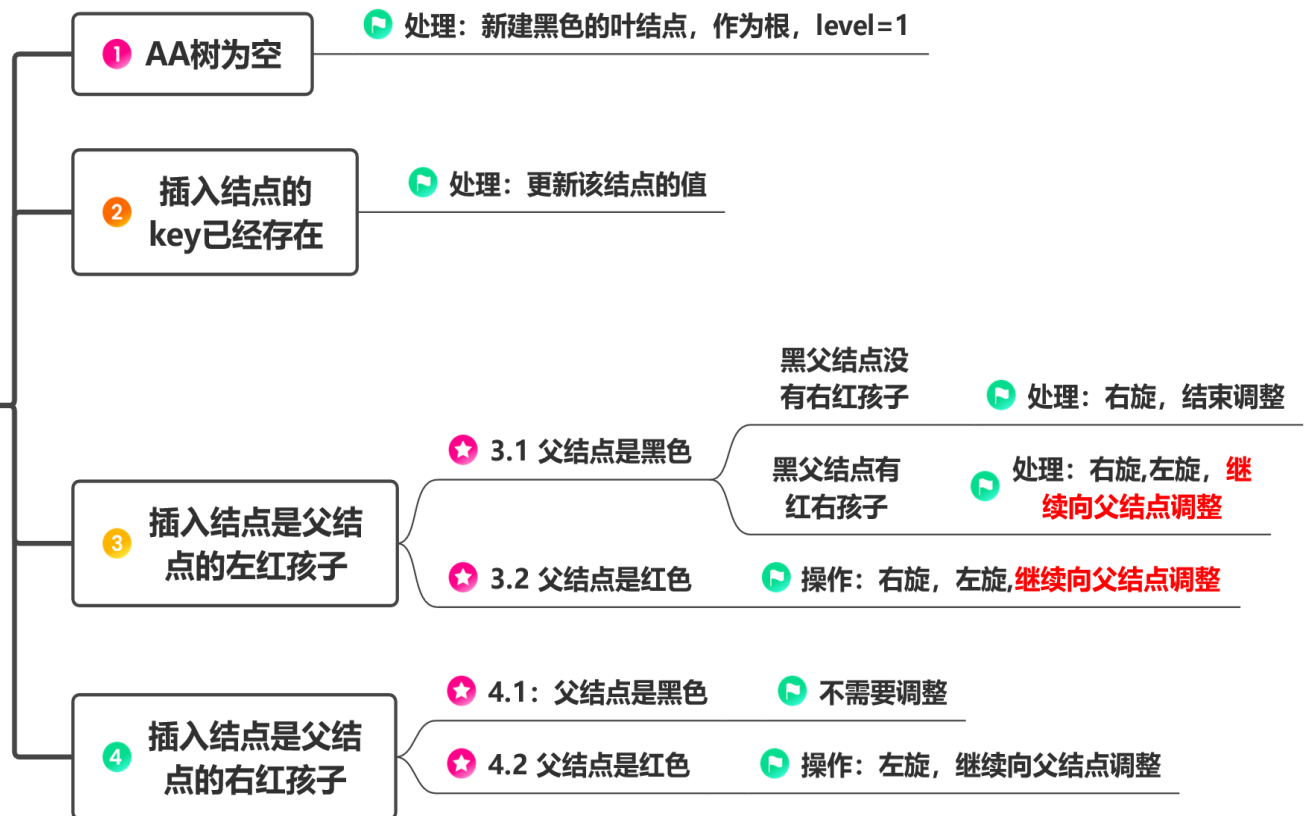(too many black
nodes on one
path)

Red-Black
Tree

The skew（右旋）
procedure is a
simple right
rotation between
X and P

AA-Tree

AA树结点的插入

AA树的插入操作

❶ **AA树为空** — 🏴 处理：新建黑色的叶结点，作为根，level=1

❷ **插入结点的key已经存在** — 🏴 处理：更新该结点的值

❸ **插入结点是父结点的左红孩子**

⭐ **3.1 父结点是黑色**
- 黑父结点没有右红孩子 — 🏴 处理：右旋，结束调整
- 黑父结点有红右孩子 — 🏴 处理：右旋,左旋，**继续向父结点调整**

⭐ **3.2 父结点是红色** — 🏴 操作：右旋，左旋,**继续向父结点调整**

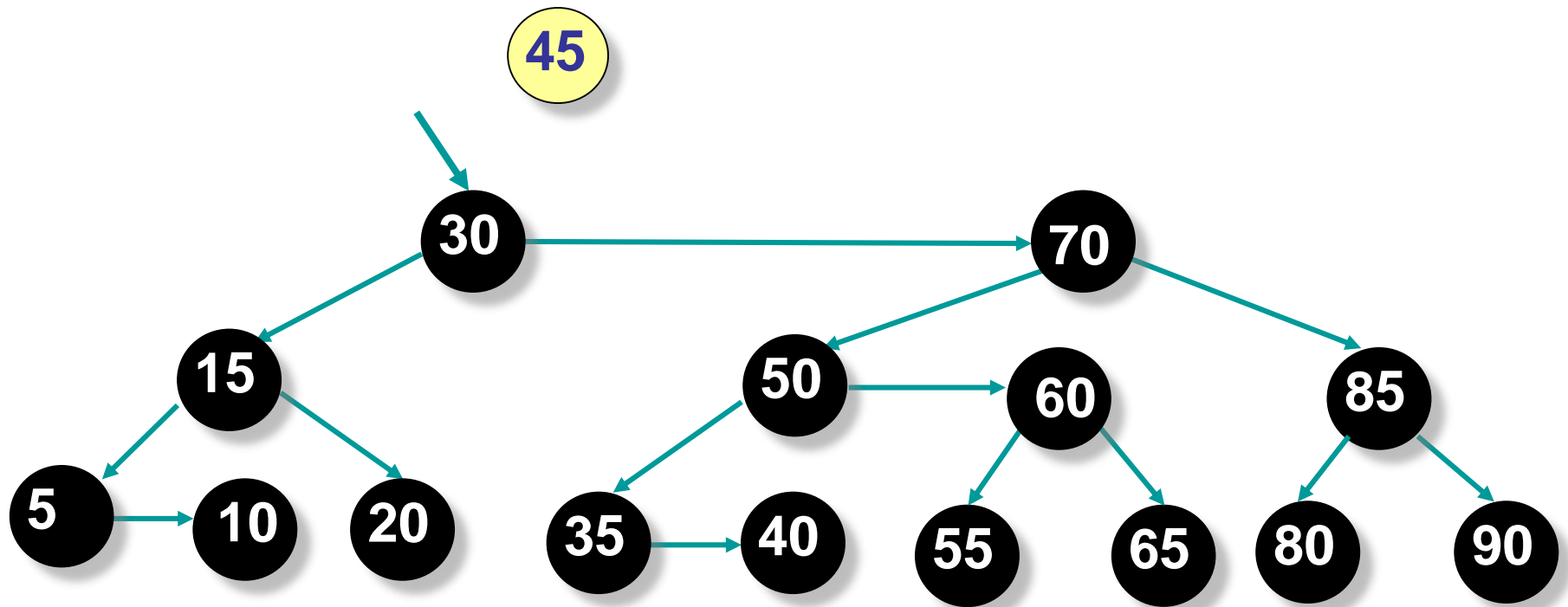❹ **插入结点是父结点的右红孩子**

⭐ **4.1：父结点是黑色** — 🏴 不需要调整

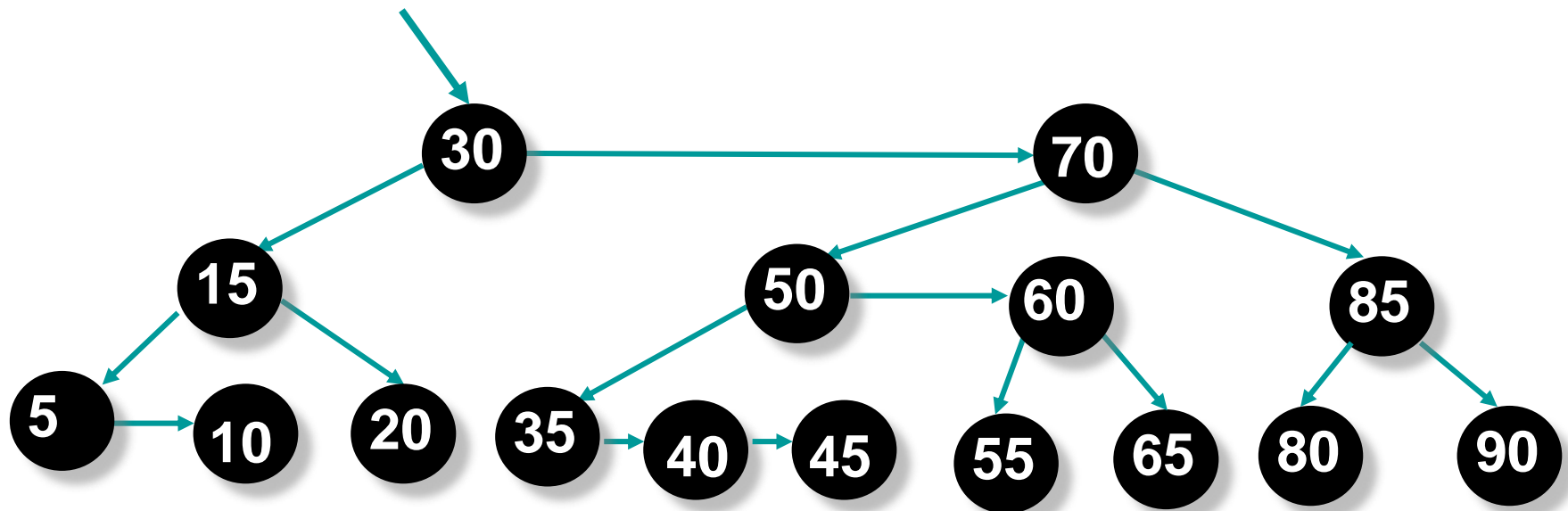⭐ **4.2 父结点是红色** — 🏴 操作：左旋，继续向父结点调整

Example: Insert 45

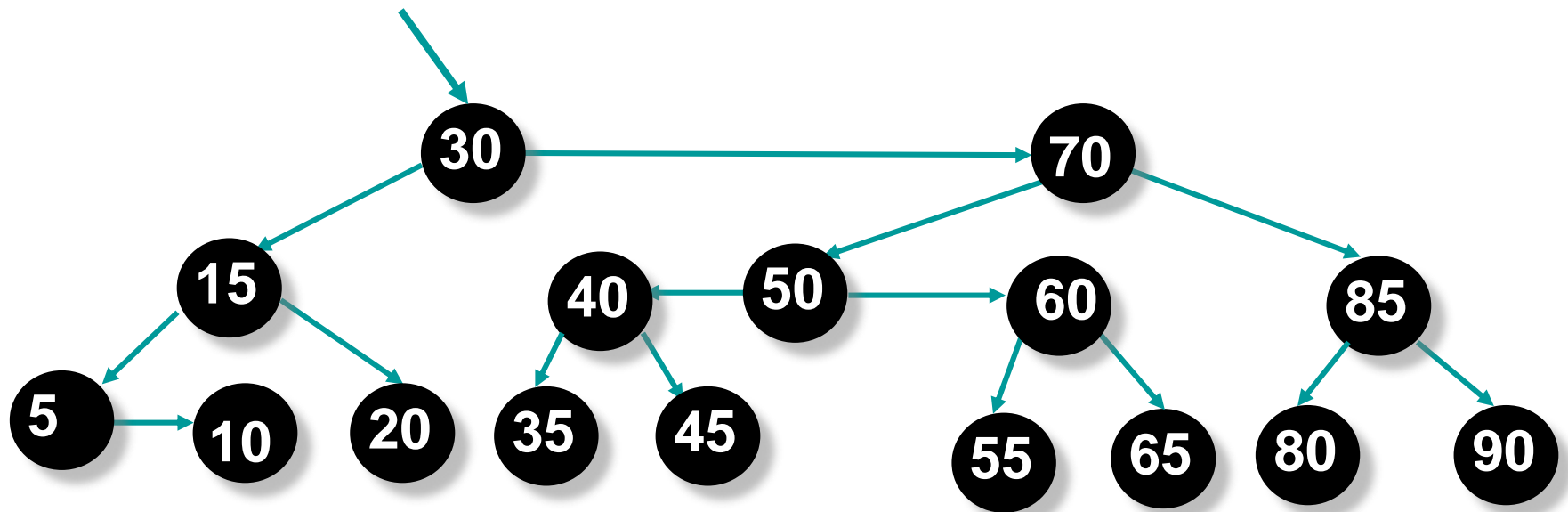First, insert as for simple binary search tree

Example: Insert 45

After insert to right of 40:

Problem: Consecutive horizontal links starting at 35, so need split
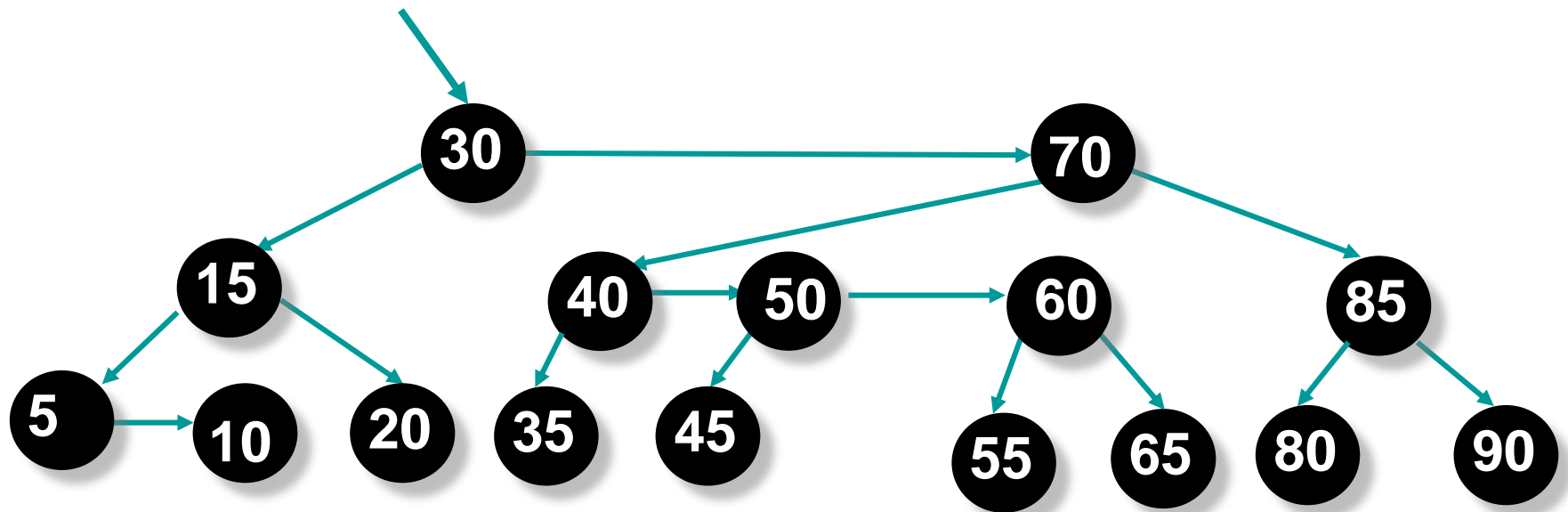
Example: Insert 45

After split at 35:

Problem: Left horizontal link at 50 is introduced, so need skew

Example: Insert 45
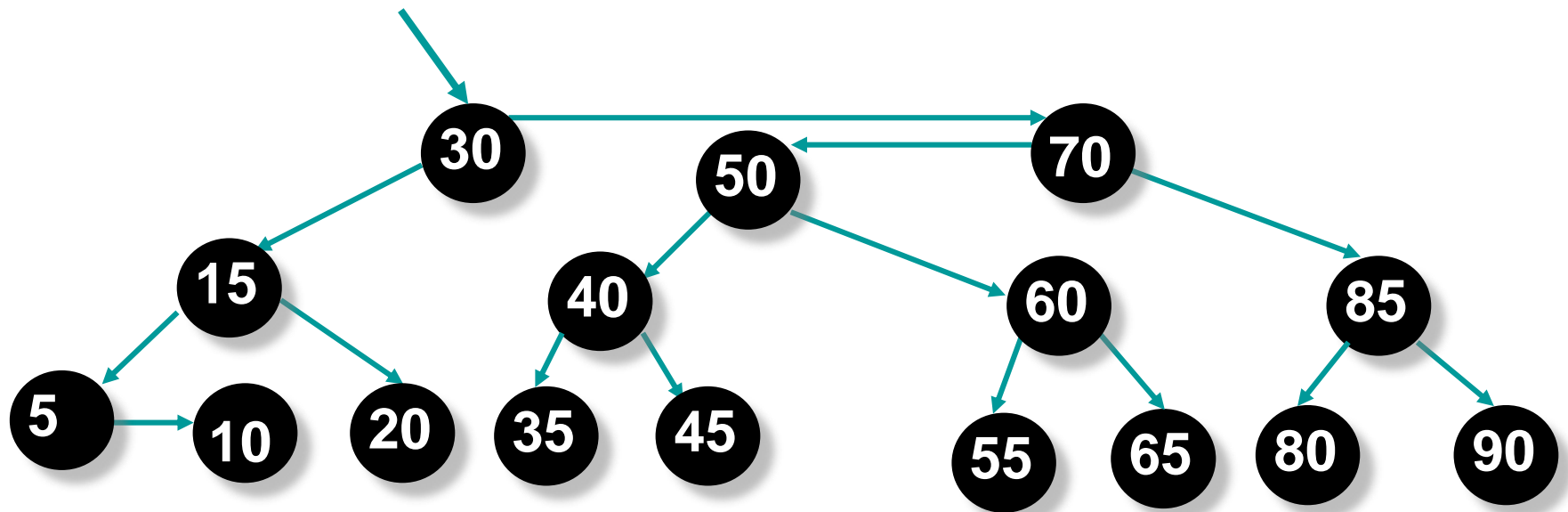
After skew at 50:

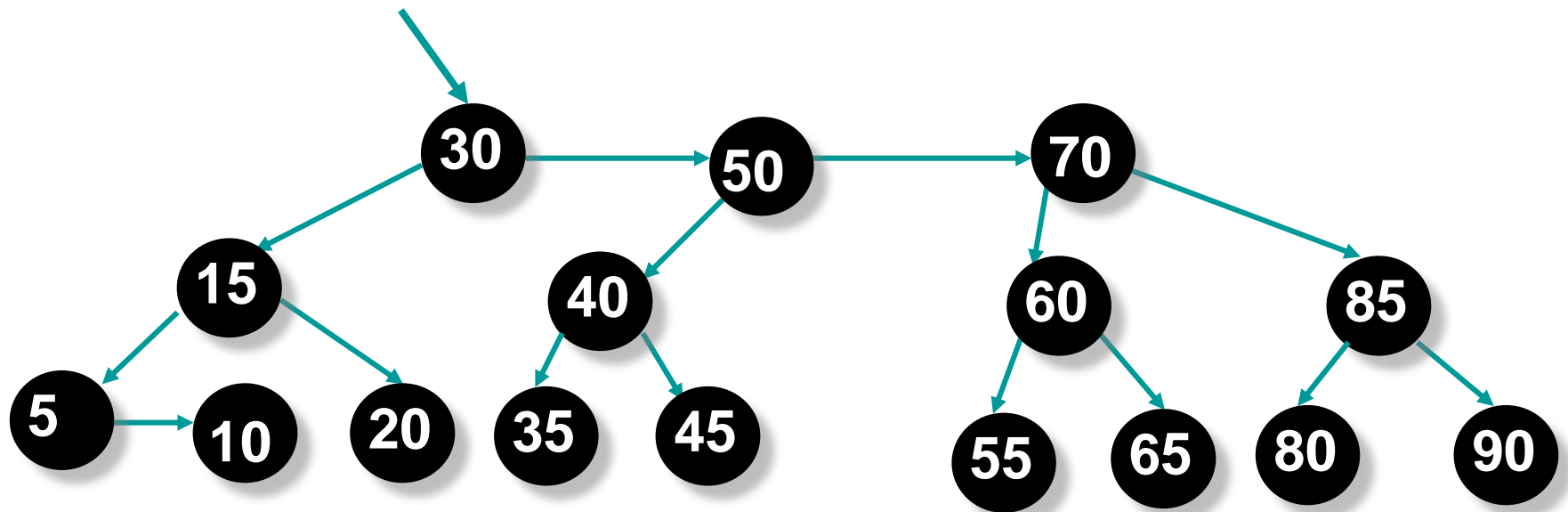Problem: Consecutive horizontal links starting at 40, so need split

Example: Insert 45

After split at 40:

Problem: Left horizontal link at 70 is introduced (50 is now on same level as 70), so need skew
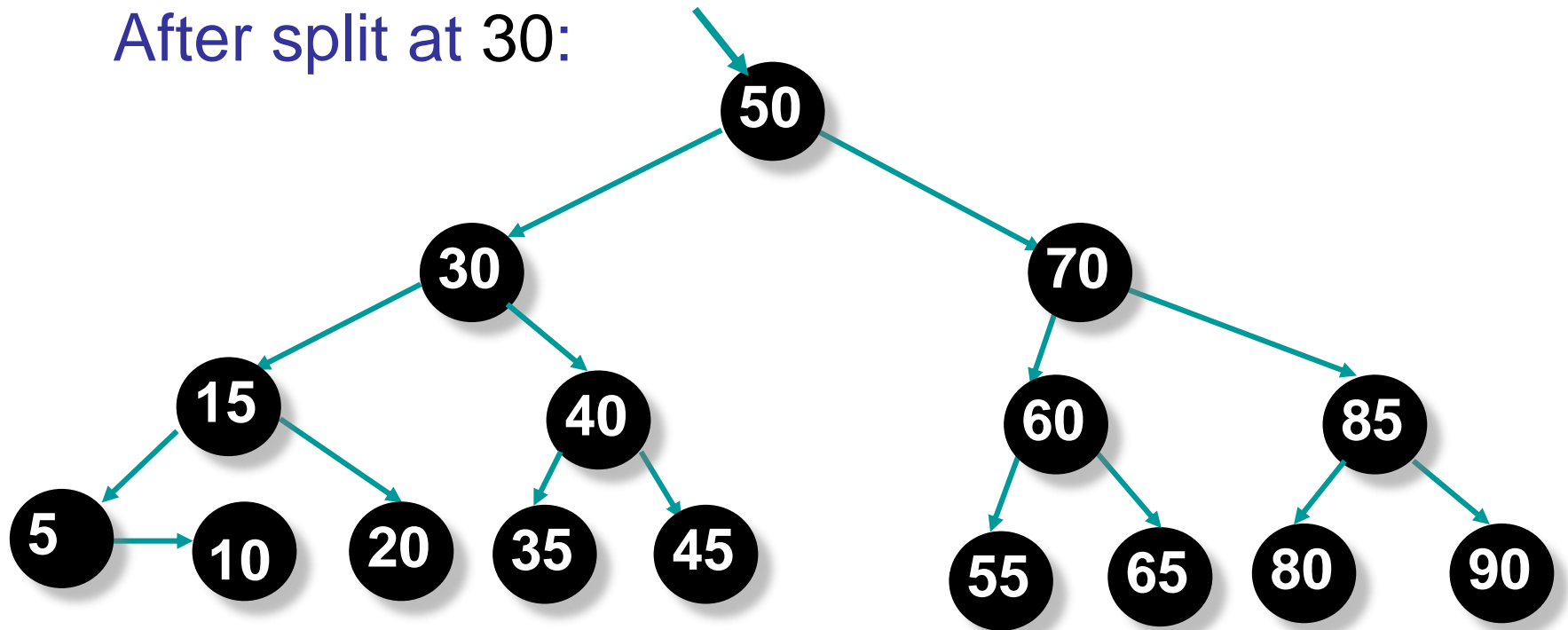
Example: Insert 45

After skew at 70:

Problem: Consecutive horizontal links starting at 30, so need split

Example: Insert 45

After split at 30:

Insertion is complete (finally!)

AA-Tree Insertion Algorithm

// Inserts node y into AA-Tree rooted at node x

// Only for tree nodes with no pointer to parent

AAInsert (x, y)

        if (x = NIL) // have found where to insert y

                then x ← y

        else if key[y] < key[x]

                then AAInsert(left[x], y)

        else if key[y] > key[x]

                then AATInsert(right[x], y)

        else
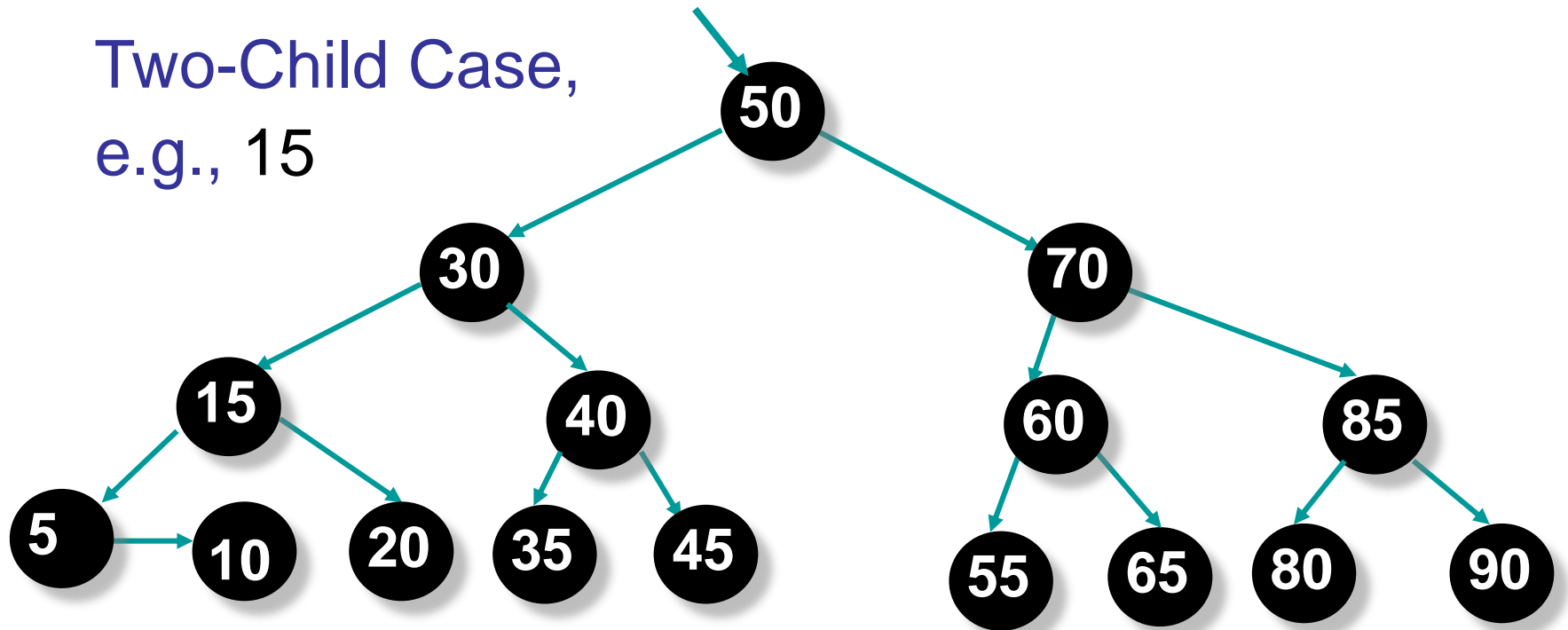
                y is a duplicate; handle duplicate case

        skew (x)   // Do skew and split at each level

        split (x)

Deletion
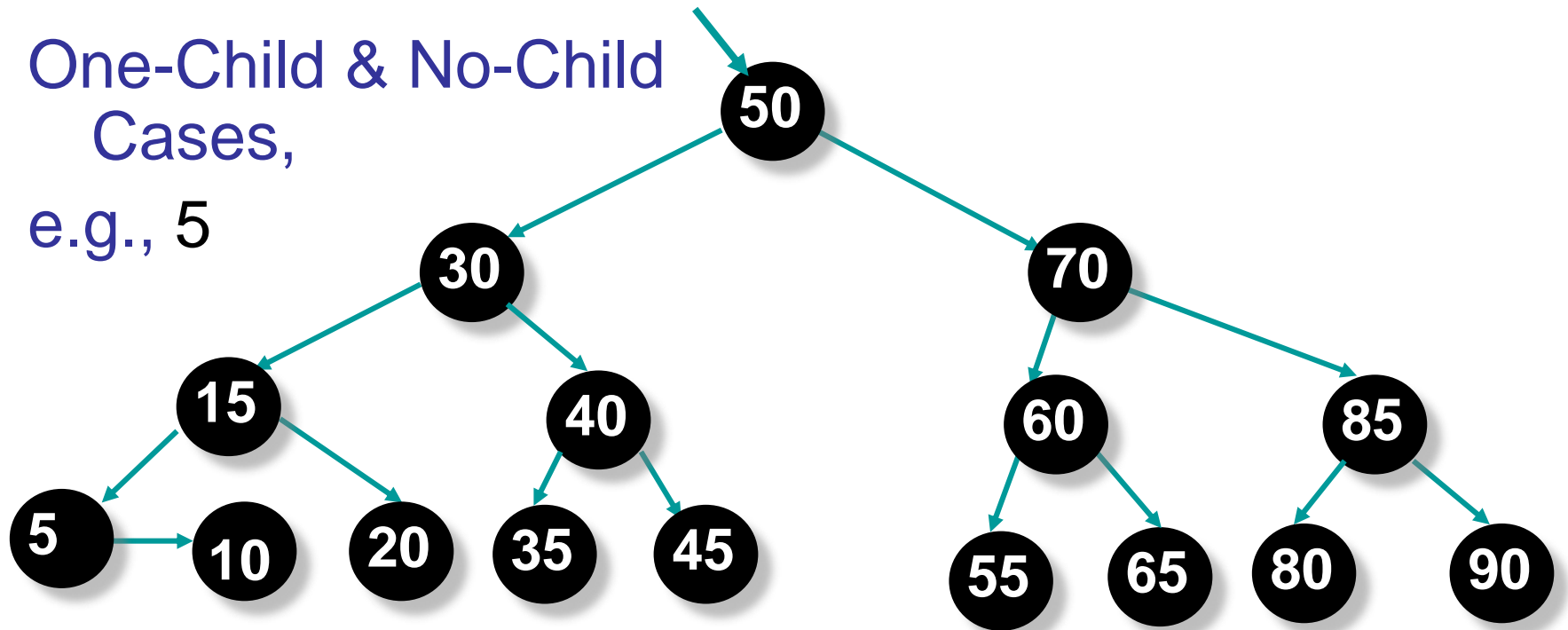
Two-Child Case,
e.g., 15

Same as for simple BST: replace with smallest right child or largest left child and recursively call delete
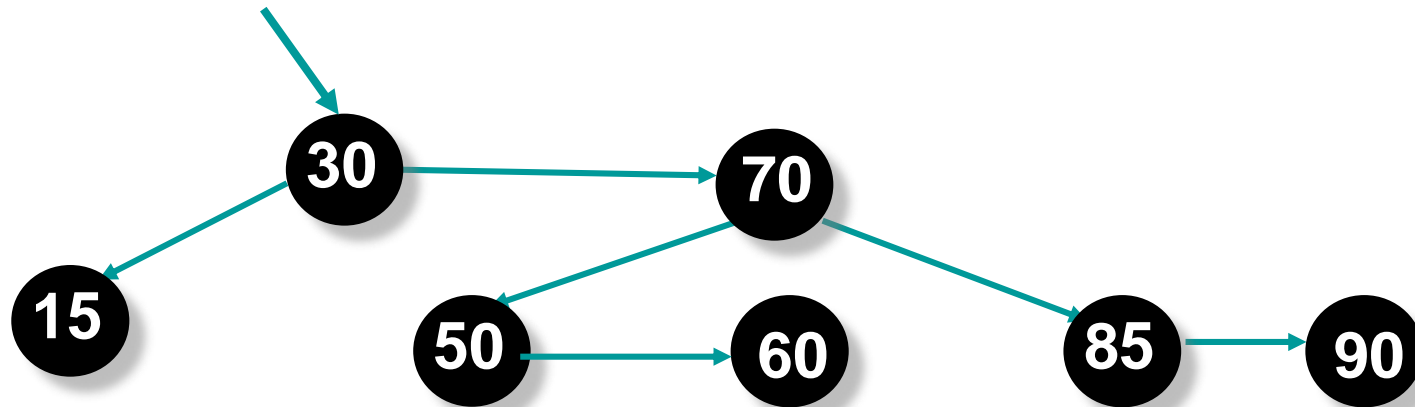
Deletion

One-Child & No-Child Cases,

e.g., 5

Note that these are all at level one, so everything boils down to deleting a level one node
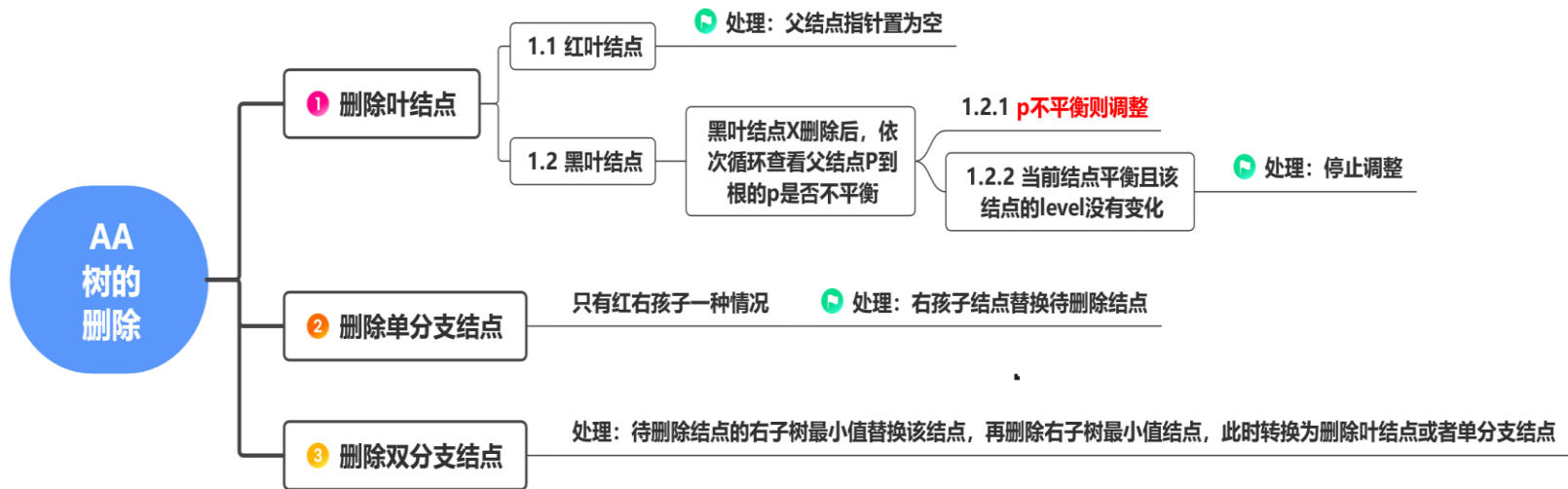
Deletion at Level 1

In the worst case, deleting one leaf node, e.g., 15, could cause six nodes to all become at one level, introducing horizontal left links.
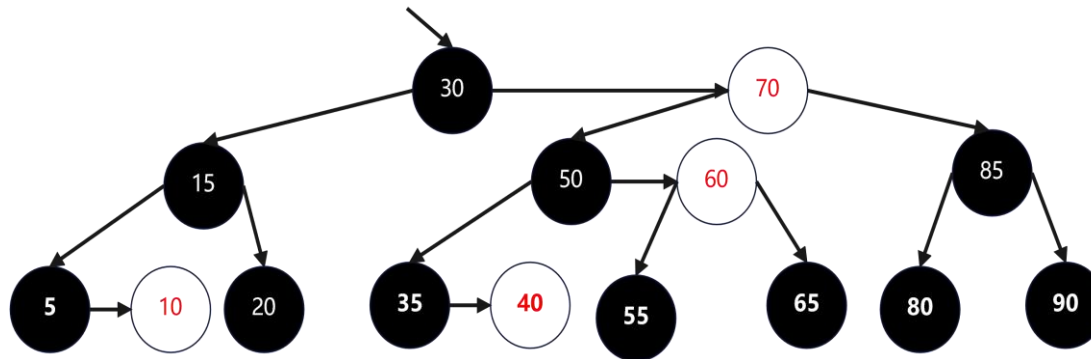

However, it turns out that all cases can be handled by three calls to skew, followed by two calls to split (implementation can be found in various texts if you need it someday).
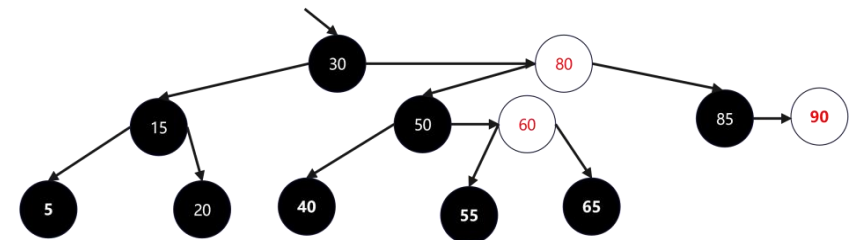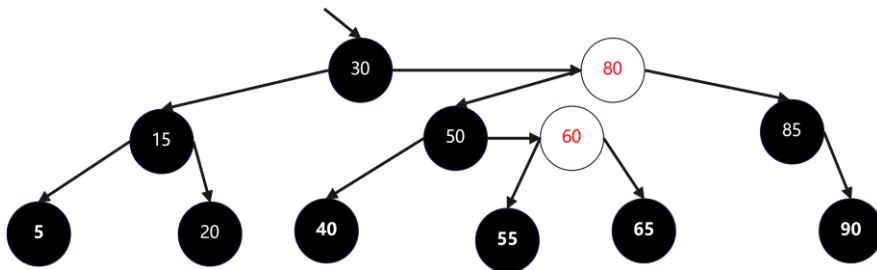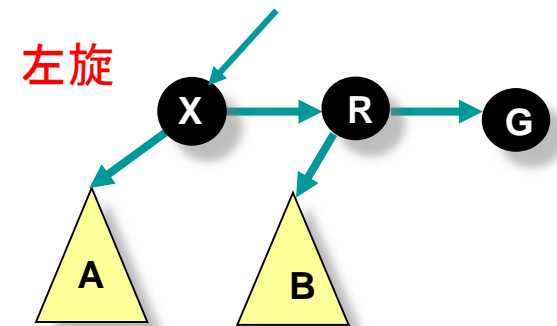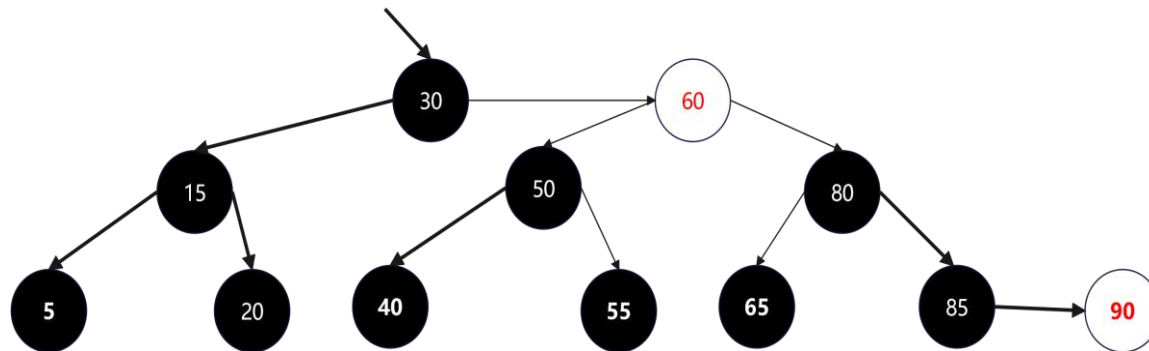
AA树结点的删除

AA 树的删除

❶ 删除叶结点
  1.1 红叶结点 —— 🔵 处理：父结点指针置为空
  1.2 黑叶结点 —— 黑叶结点X删除后，依次循环查看父结点P到根的p是否不平衡
    1.2.1 **p不平衡则调整**
    1.2.2 当前结点平衡且该结点的level没有变化 —— 🔵 处理：停止调整

❷ 删除单分支结点 —— 只有红右孩子一种情况 🔵 处理：右孩子结点替换待删除结点

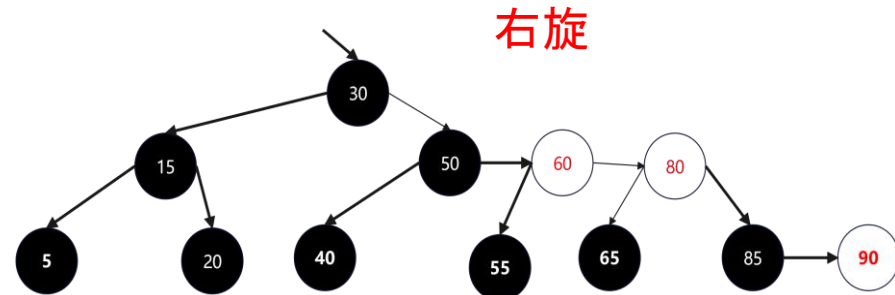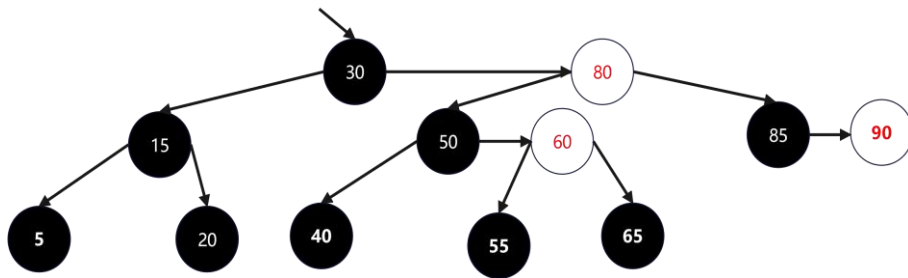❸ 删除双分支结点 —— 处理：待删除结点的右子树最小值替换该结点，再删除右子树最小值结点，此时转换为删除叶结点或者单分支结点

AA树结点的删除

- 删除10, 35和70
- 85的level减少1

AA树结点的删除

右旋

左旋

BST Applets

http://people.ksp.sk/~kuko/bak/index.html

http://www.site.uottawa.ca/~stan/csi2514/applets/avl/BT.html

http://www.cis.ksu.edu/~howell/viewer/viewer.html