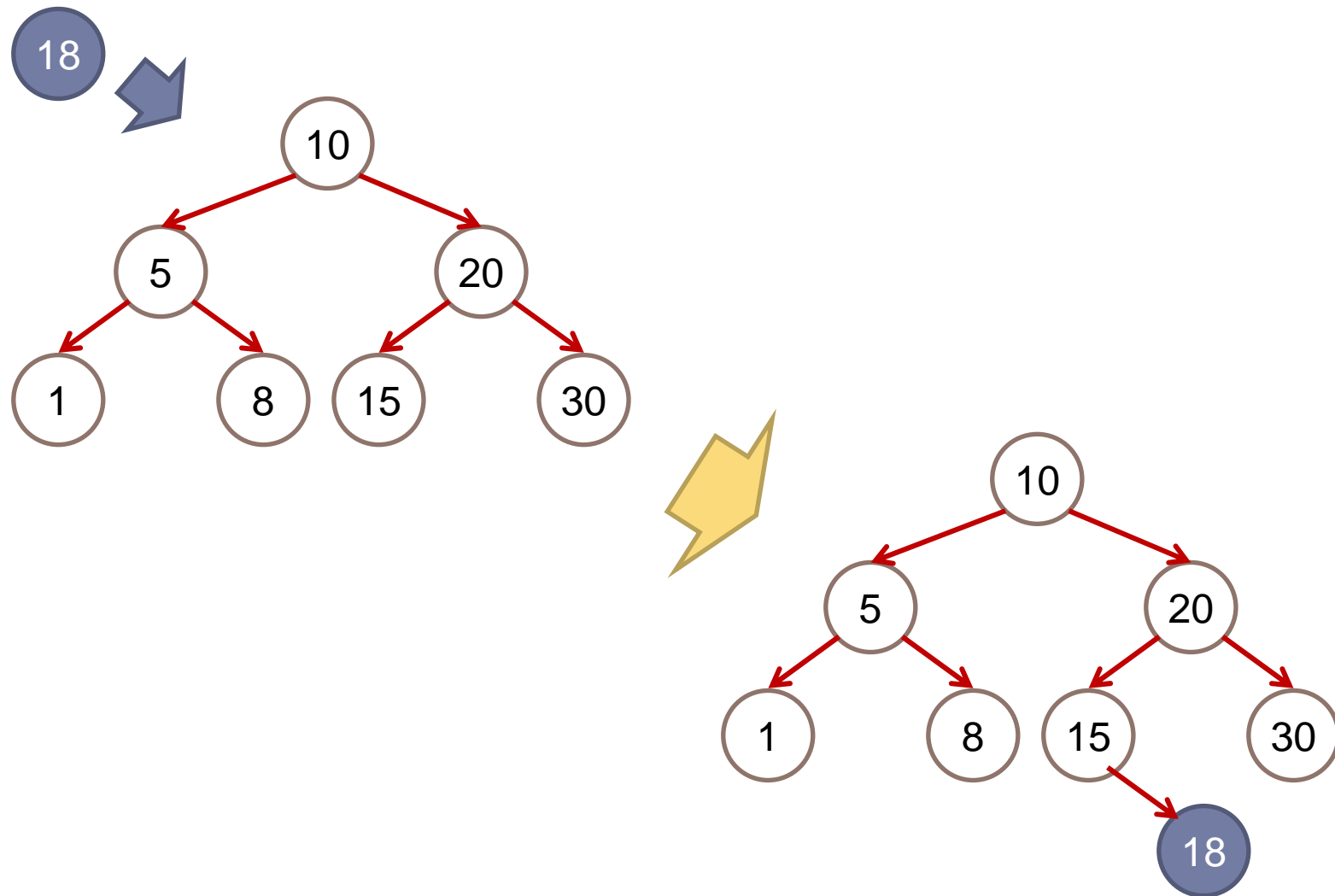


二元搜尋樹

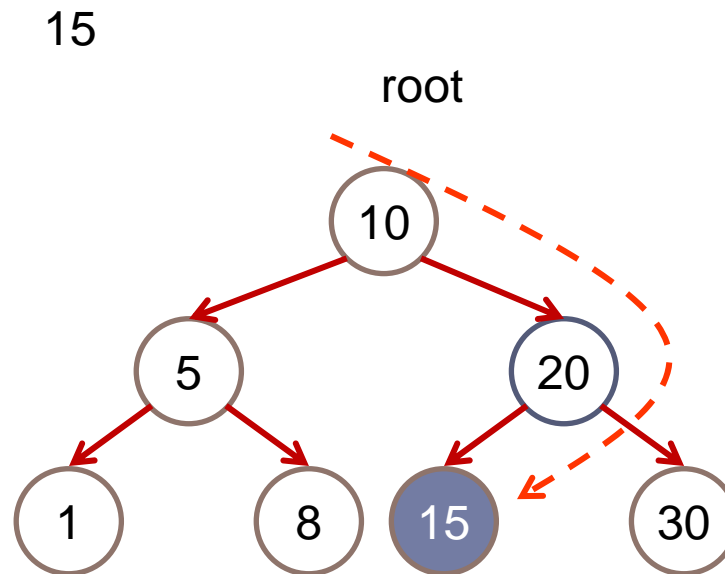
- ▶ 「二元搜尋樹」 (Binary Search Trees) 是一種二元樹，其節點資料的排列擁有一些特性，如下所示：
 - 二元樹的每一個節點值都不相同，在整棵二元樹中的每一個節點都擁有不同值。
 - 每一個節點的資料大於左子節點的資料，但是小於右子節點的資料。
 - 節點的左、右子樹也是一棵二元搜尋樹。

二元搜尋樹: 插入



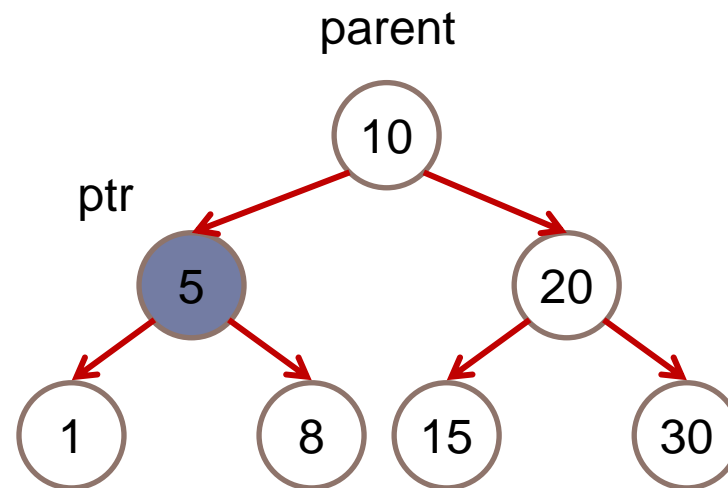
二元搜尋樹: 搜尋

- ▶ 比較：一般二元樹的搜尋



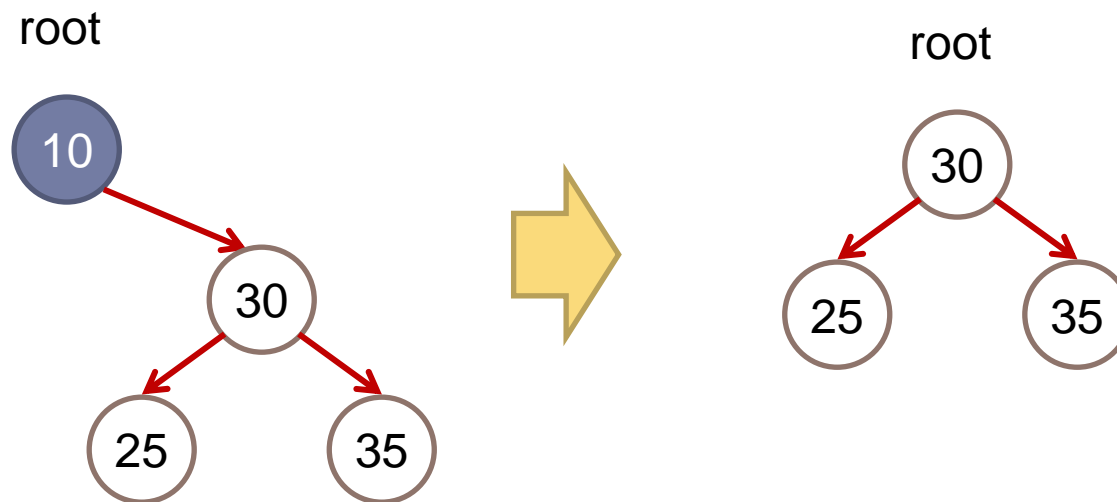
二元搜尋樹: 刪除

- ▶ 必須先取得欲刪節點之父節點 (parent)
- ▶ 判斷此節點 (ptr) 為父節點之左節點或右節點
- ▶ 考量三種情況：
 - 情況1: 節點沒有左子樹
 - 情況2: 節點沒有右子樹
 - 情況3: 節點有左右子樹



二元搜尋樹: 刪除

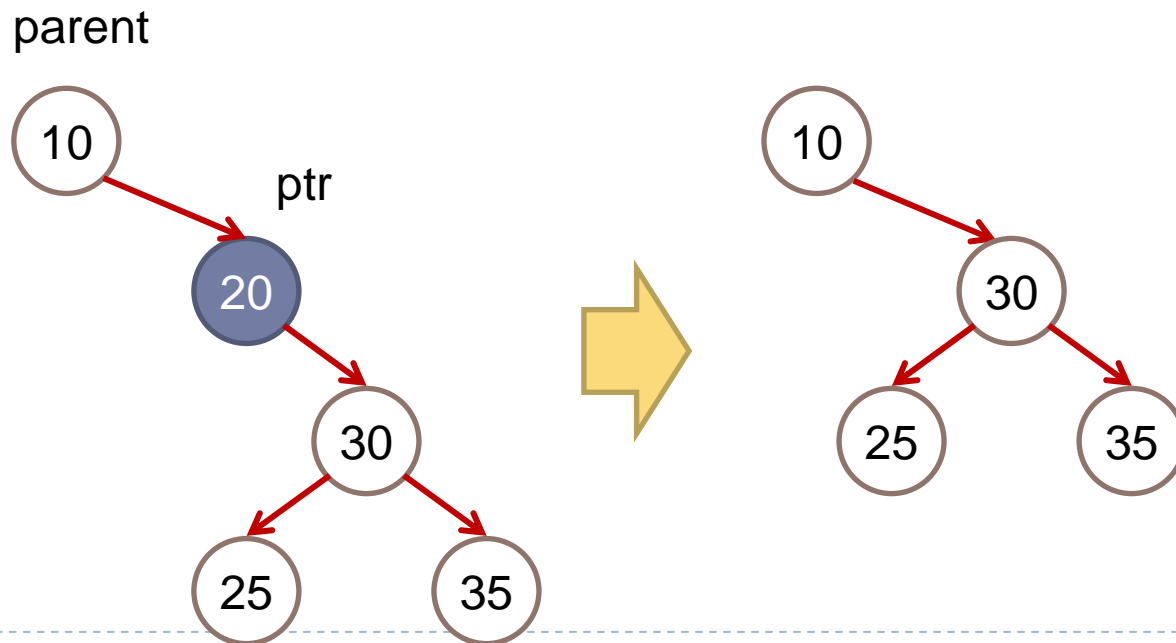
- ▶ 情況1: 節點沒有左子樹
 - 如果要刪的是根節點
 - 其他
 - ▶ 要刪除的節點在父節點右方
 - ▶ 要刪除的節點在父節點左方



二元搜尋樹: 刪除

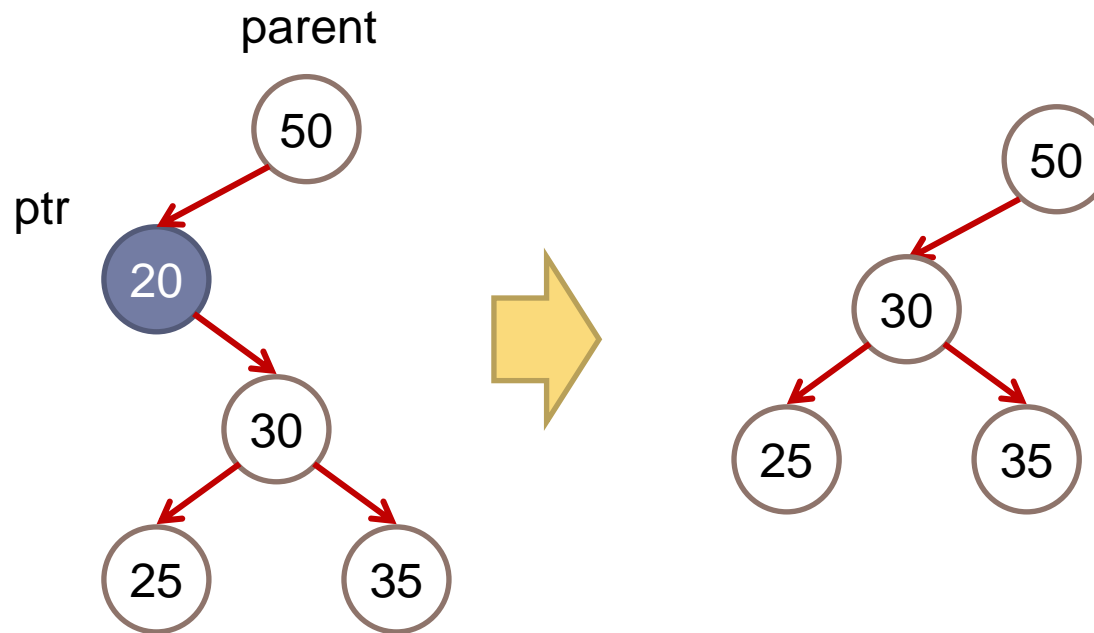
► 情況1: 節點沒有左子樹

- 如果要刪的是根節點
- 其他
 - 要刪除的節點在父節點右方
 - 要刪除的節點在父節點左方



二元搜尋樹: 刪除

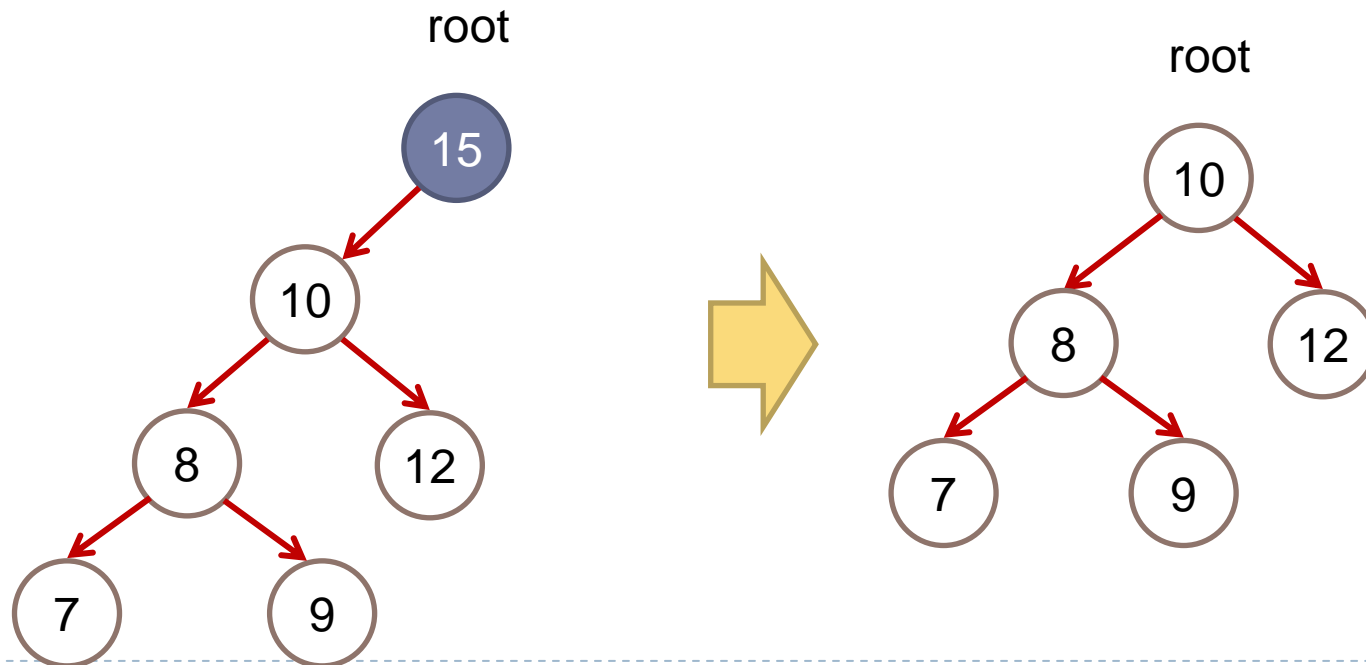
- ▶ 情況1: 節點沒有左子樹
 - 如果要刪的是根節點
 - 其他
 - ▶ 要刪除的節點在父節點右方
 - ▶ 要刪除的節點在父節點左方



二元搜尋樹: 刪除

► 情況2: 節點沒有右子樹

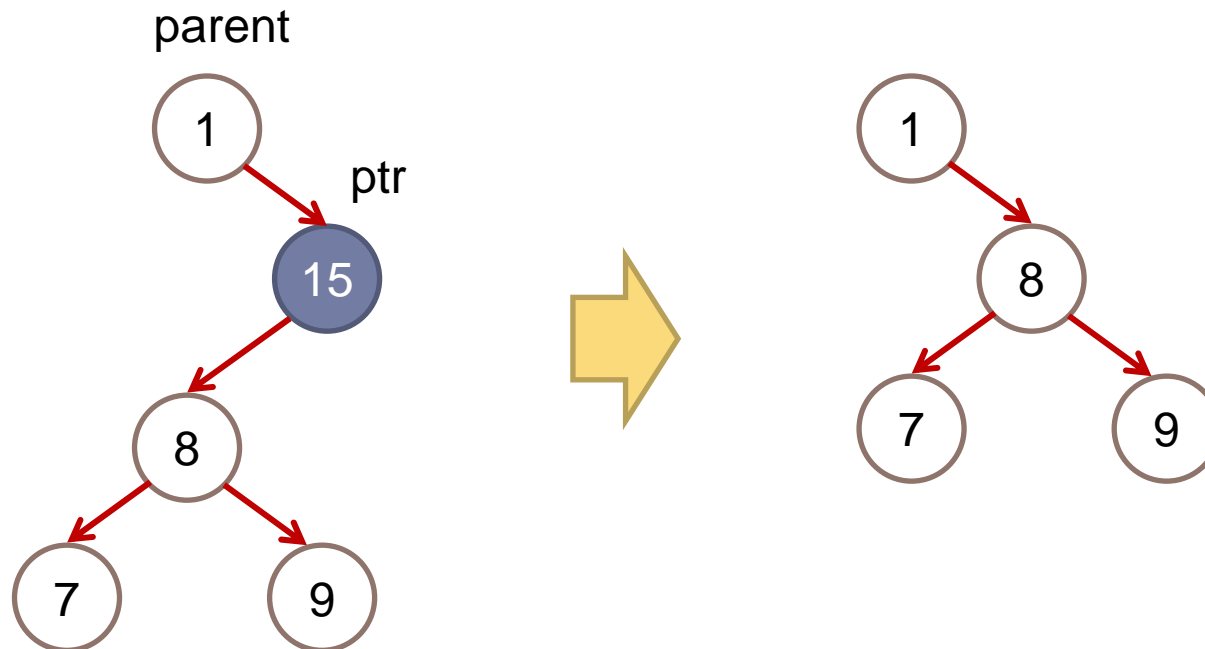
- 如果要刪的是根節點
- 其他
 - 要刪除的節點在父節點右方
 - 要刪除的節點在父節點左方



二元搜尋樹: 刪除

► 情況2: 節點沒有右子樹

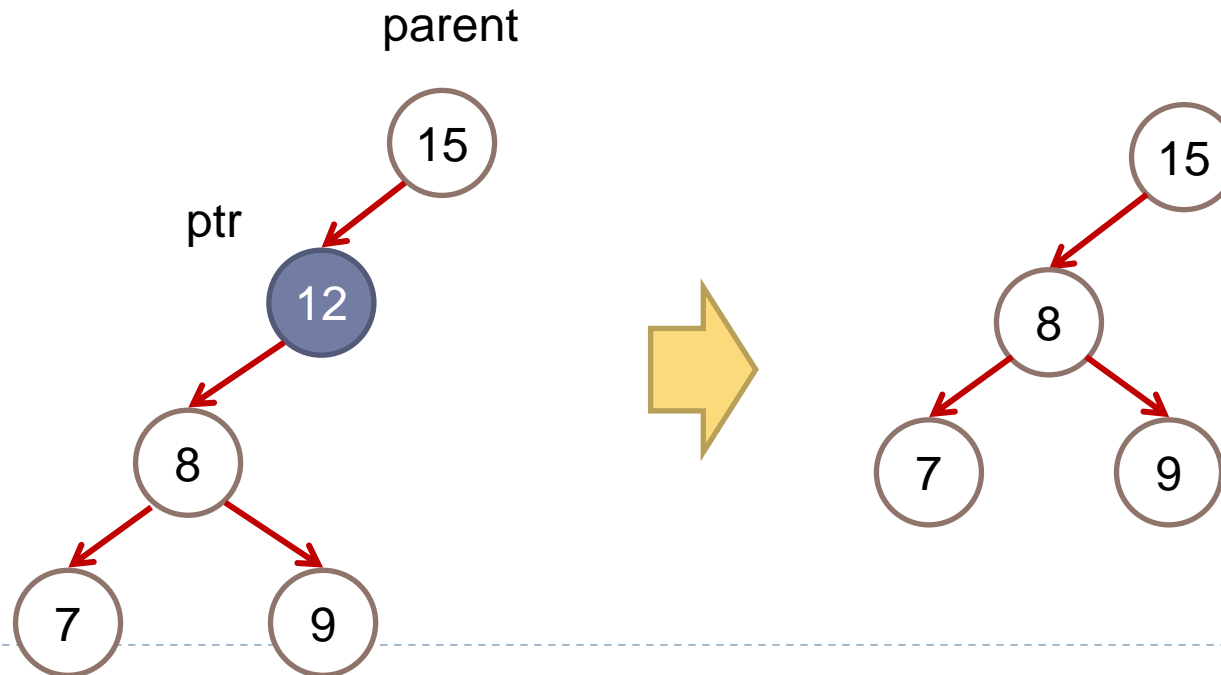
- 如果要刪的是根節點
- 其他
 - 要刪除的節點在父節點右方
 - 要刪除的節點在父節點左方



二元搜尋樹: 刪除

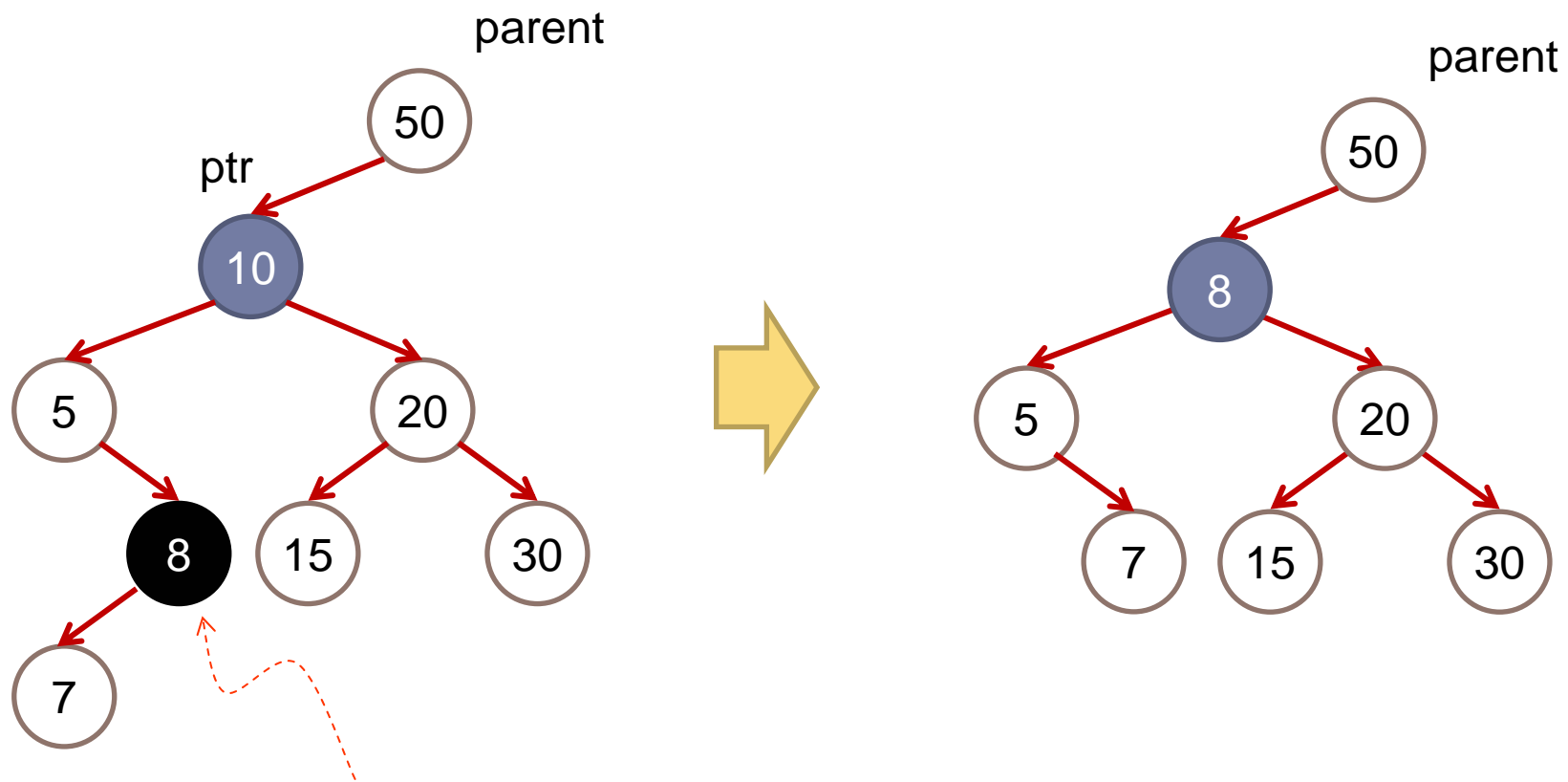
► 情況2: 節點沒有右子樹

- 如果要刪的是根節點
- 其他
 - 要刪除的節點在父節點右方
 - 要刪除的節點在父節點左方



二元搜尋樹: 刪除

- ▶ 情況3: 節點有左右子樹
 - 往左子樹找最大值當取代點



一定沒有右子樹,一定在其父節點的右方