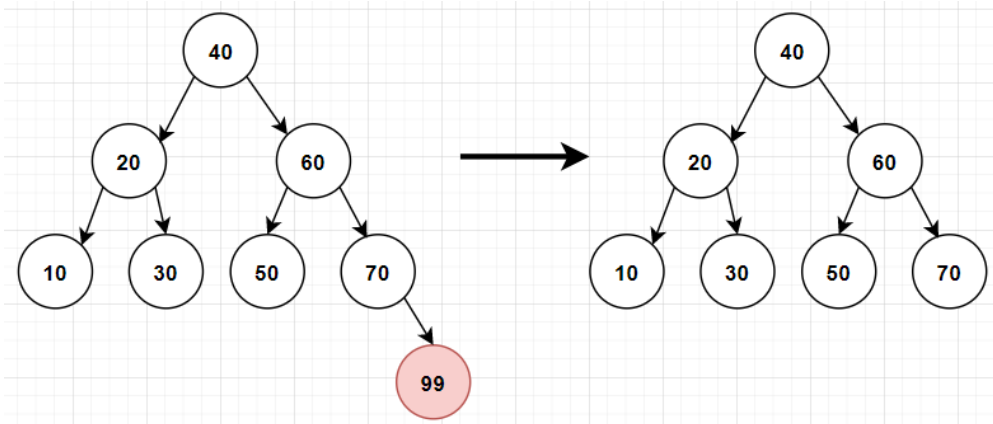
Bài 5.7: Xóa node trên cây BST

- ✓ Xóa node lá
- ✓ Xóa node có 1 cây con
- ✓ Xóa node có 2 cây con
- ✓ Ví dụ minh họa & bài tập

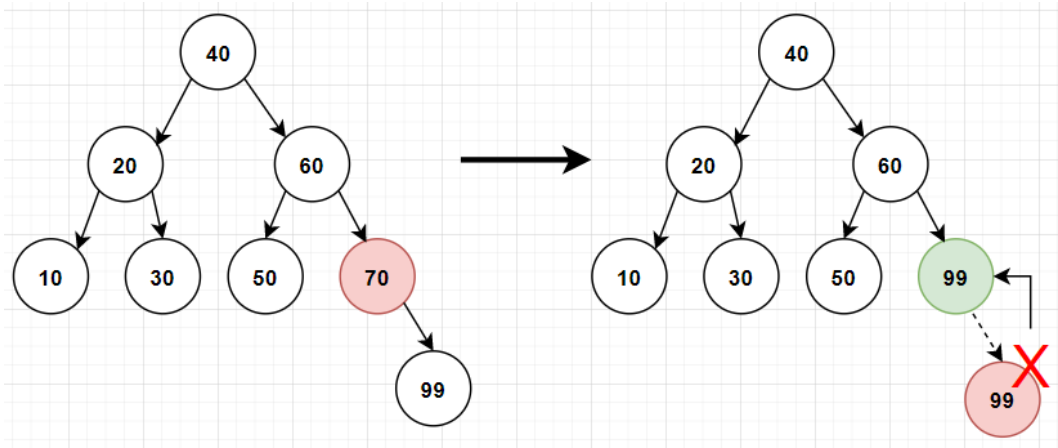
Xóa node lá

- Nếu node cần xóa là node lá, đơn giản ta chỉ cần xóa bỏ node đó.
- Thay liên kết trỏ đến node đó bằng null.



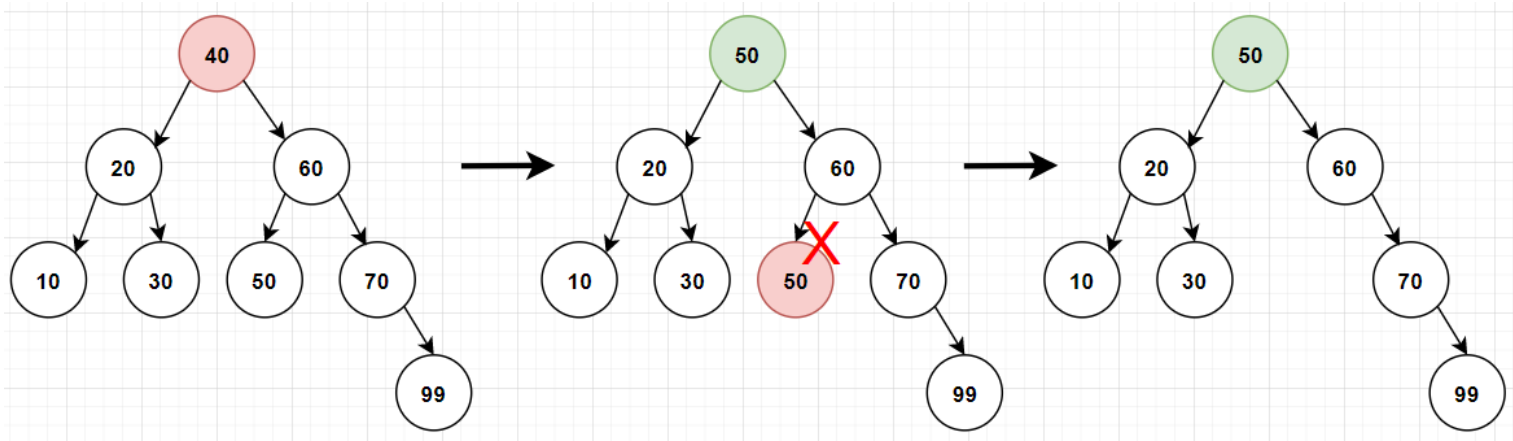
Xóa node có 1 cây con

- Đưa giá trị node con lên thế chỗ cho node bị xóa.
- Xóa bỏ node con.



Xóa node có 2 cây con

- Tìm node nhỏ nhất bên cây con phải, copy giá trị vào node cần xóa r.
- Xóa bỏ node có giá trị nhỏ nhất ở cây con phải.



Code mẫu

```
private Node<T> remove(Node<T> r, T x) {
    if (r == null) { // nếu cây rỗng
        return null;
    }
    if (x.compareTo(r.data) < 0) { // xóa bên trái
        r.leftNode = remove(r.leftNode, x);
    } else if (x.compareTo(r.data) > 0) {
        r.rightNode = remove(r.rightNode, x);
    } else { // nếu giá trị node r trùng với x, node r là node cần xóa
        // xóa node có 0 hoặc 1 cây con
        if (r.leftNode == null) {
            r = r.rightNode; // gán cây con phải thế chỗ cho r
        } else if (r.rightNode == null) {
            r = r.leftNode; // gán cây con trái thế chỗ cho r
        } else { // node cần xóa có 2 cây con
            // tìm giá trị nhỏ nhất ở cây con phải của r
            r.data = findMinNode(r.rightNode); // cập nhật giá trị node r
            r.rightNode = remove(r.rightNode, r.data); // xóa bỏ node đã sử dụng
        }
    }
    return r;
}
```

Code mẫu

```
// tìm node có giá trị nhỏ nhất của cây con phải
private T findMinNode(Node<T> r) {
    while (r.leftNode != null) {
        r = r.leftNode;
    }
    return r.data;
}
```

Nội dung tiếp theo

Tìm hiểu lớp TreeSet