

CHƯƠNG 4

CÂY AVL

(G.M. Adelson-Velsky và E.M. Landis)

Bộ môn Công Nghệ Phần Mềm



CANTHO UNIVERSITY

Nội dung

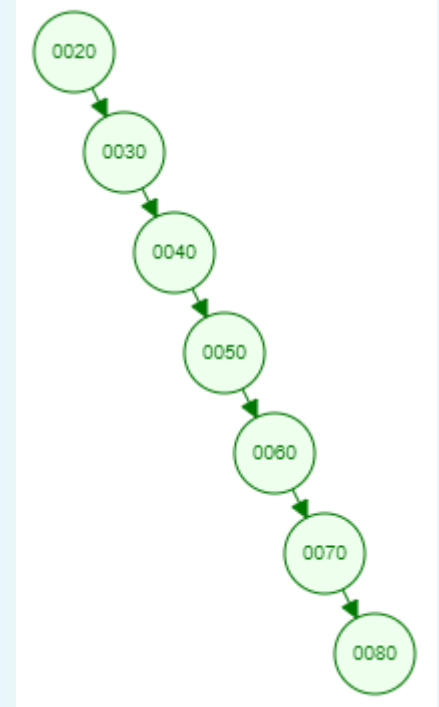
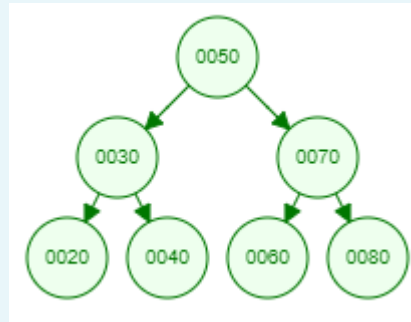
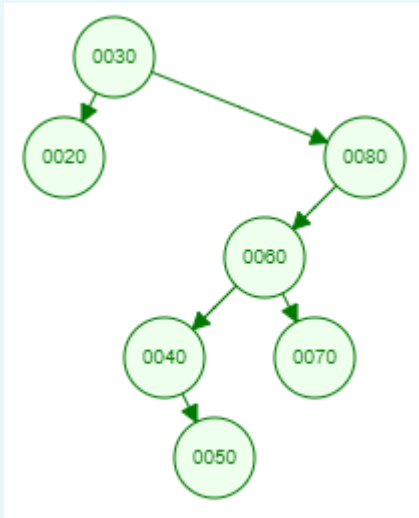
- Khái niệm về cây tìm kiếm nhị phân cân bằng
- Khái niệm về cây AVL
- Các thuật toán trên cây AVL
- Ý tưởng cài đặt cây bằng con trỏ



CANTHO UNIVERSITY

Khái niệm về cây tìm kiếm nhị phân cân bằng

- Cây TKNP được dựng từ các khóa:
20, 30, 40, 50, 60, 70, 80



- Nhận xét: Độ phức tạp trong các giải thuật trên cây tìm kiếm nhị phân trường hợp xấu nhất là $O(n)$ và trung bình $O(\log n)$.



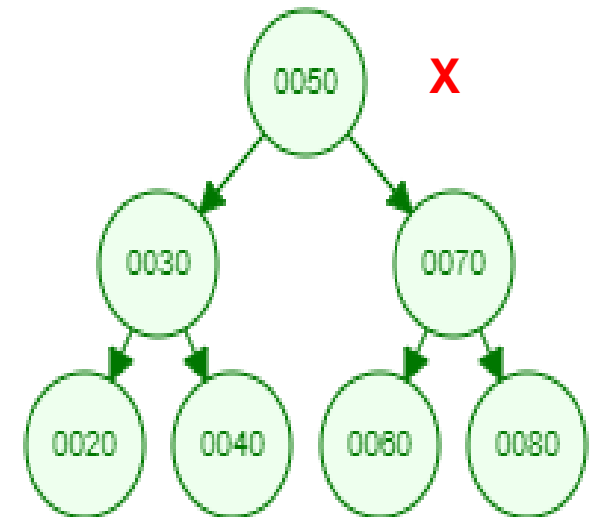
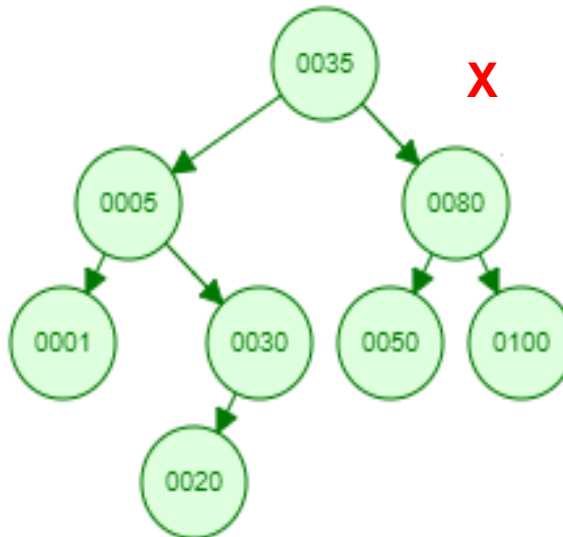
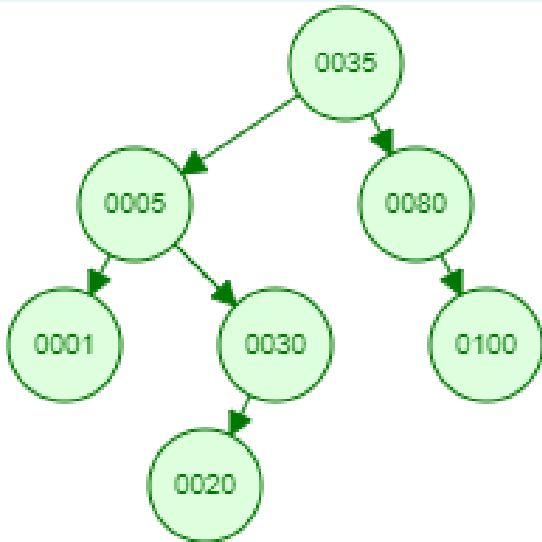
Khái niệm về cây tìm kiếm nhị phân cân bằng

- Quá trình tìm kiếm khóa, thêm nút, xóa nút trên cây TKNP là quá trình di chuyển từ nút gốc ra nút lá. ➔ Cây càng cao thì giải thuật càng kém hiệu quả.
- Do vậy rất cần thiết phải xây dựng cây TKNP mà trong đó chiều cao của cây càng nhỏ càng tốt để cho các giải thuật được hiệu quả nhất. Cây ở dạng này được gọi là cây cân bằng.



Cây TKNP cân bằng hoàn toàn

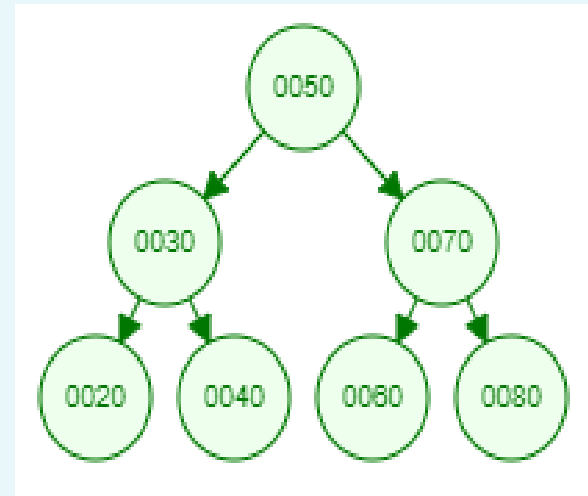
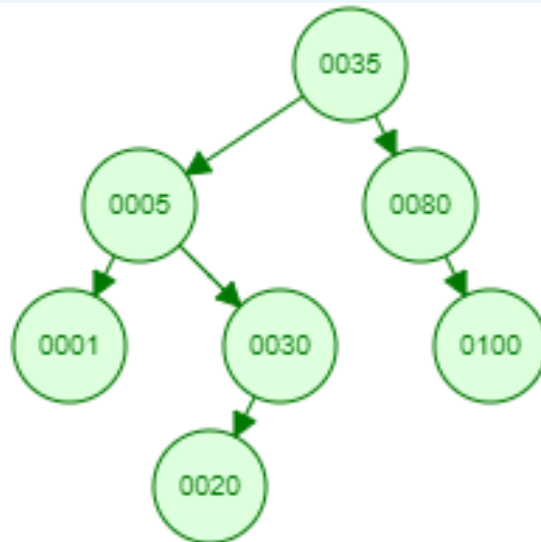
- Cây TKNP cân bằng hoàn toàn là cây TKNP mà tại mỗi nút có **tổng số nút của cây con trái và con phải lệch nhau không quá một**.
- Cây nào là cây cân bằng hoàn toàn?





Cây TKNP cân bằng tương đối (cân bằng về chiều cao)

- Cây TKNP **cân bằng về chiều cao** là cây TKNP mà trong đó mỗi nút đều có chiều cao con trái và con phải lệch nhau tối đa là 1.
- Cây cân bằng về chiều cao còn gọi là **cân bằng tương đối**.
- Ví dụ:





Khái niệm về cây AVL

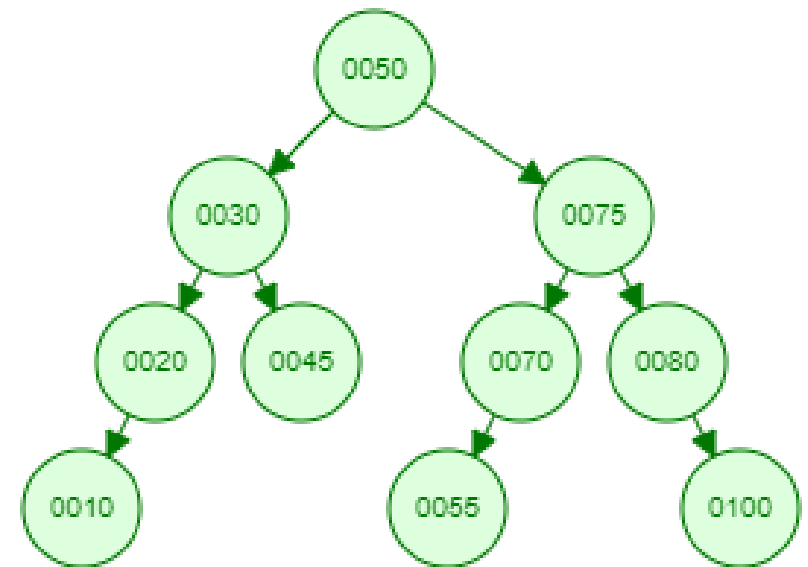
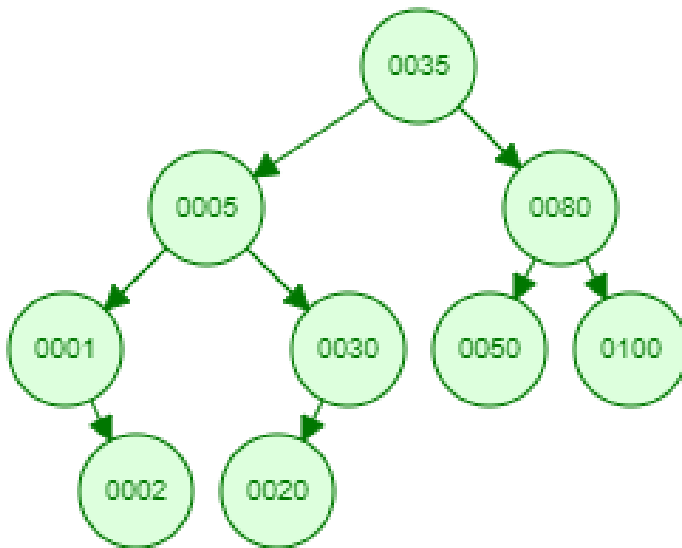
- Cây AVL được gọi theo tên của hai người đề xuất chúng, G.M. Adelson-Velsky và E.M. Landis, được công bố trong bài báo của họ vào năm 1962: "An algorithm for the organization of information." (Một thuật toán về tổ chức thông tin)
- **Cây AVL là cây TKNP mà chiều cao của hai cây con của mọi nút chênh lệch tối đa là 1.**



CANTHO UNIVERSITY

Khái niệm về cây AVL

- Ví dụ:





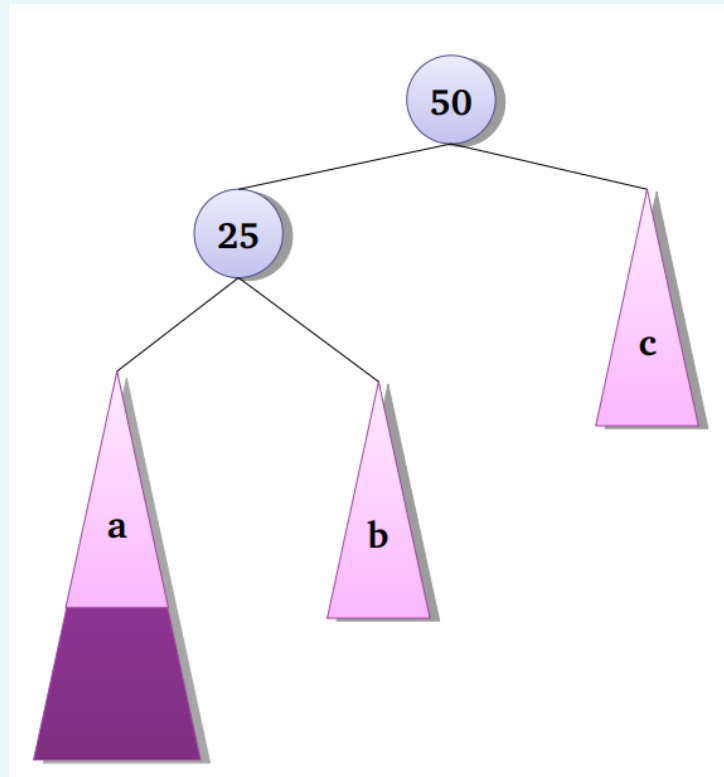
Các thao tác trên cây AVL

- Cây AVL là cây tìm kiếm nhị phân nên các thao tác cơ bản trên cây AVL tương tự như các thao tác cơ bản trên cây TKNP.
- Trong đó, thao tác thêm và xóa nút trên cây có thể làm cây mất cân bằng nên phải có thao tác cân bằng lại cây sau khi thực hiện thêm hoặc xóa nút.



Các trường hợp cây mất cân bằng

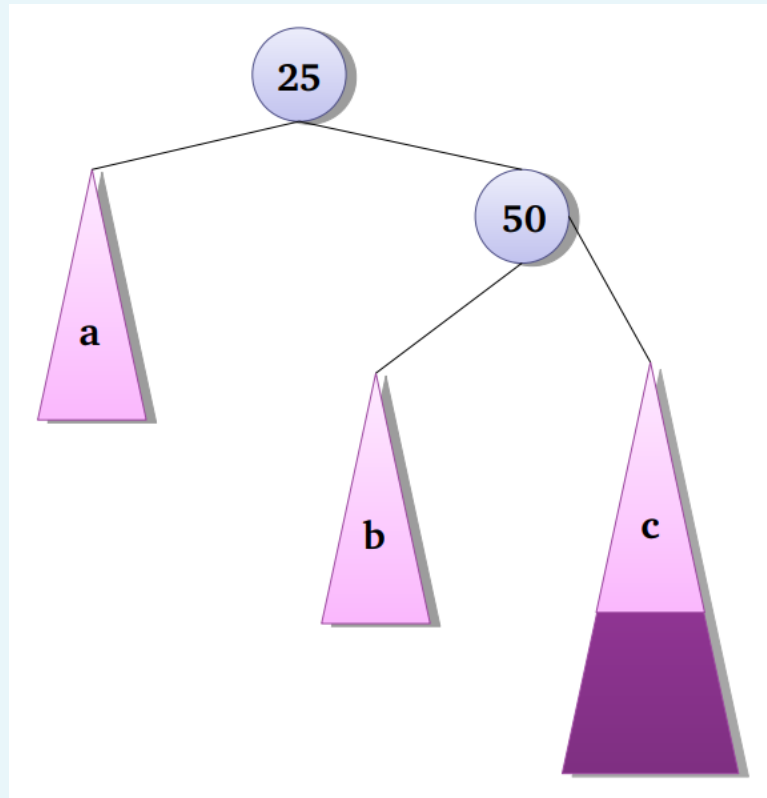
- Trường hợp 1: Cây mất cân bằng bên trái của con trái (L-L)





Các trường hợp cây mất cân bằng

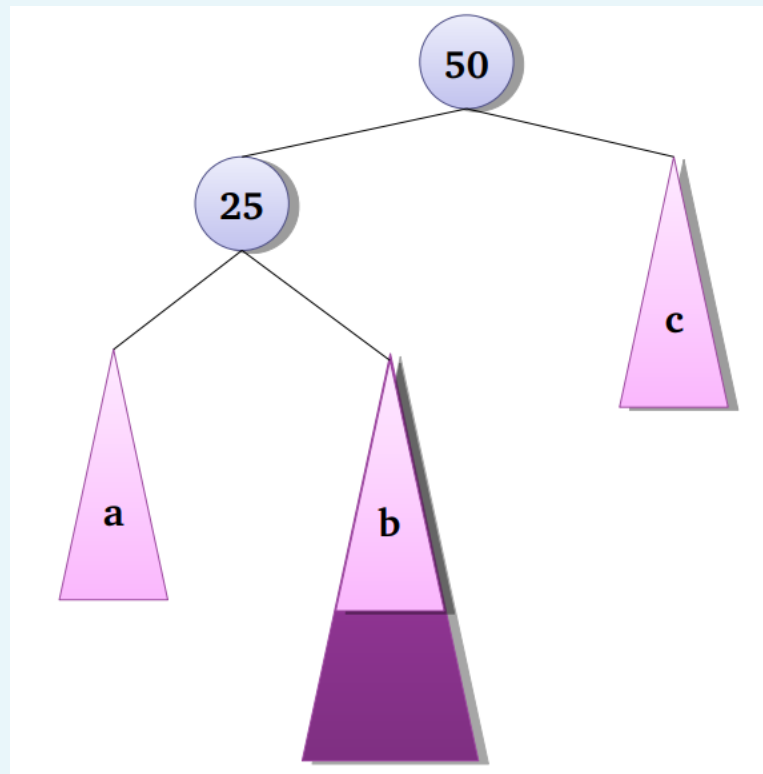
- Trường hợp 2: Cây mất cân bằng bên phải của con phải (R-R)





Các trường hợp cây mất cân bằng

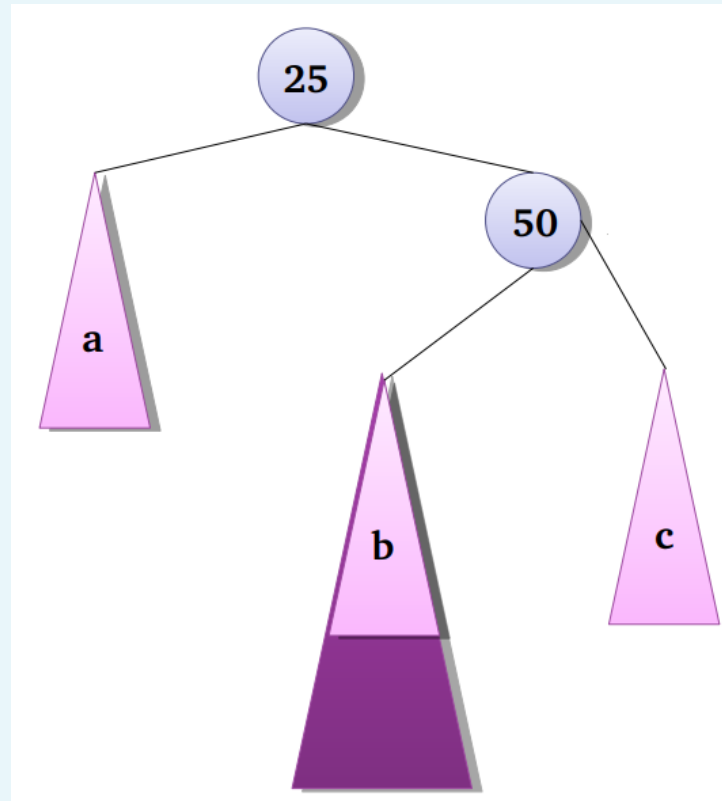
- Trường hợp 3: Cây mất cân bằng bên phải của con trái (R-L)





Các trường hợp cây mất cân bằng

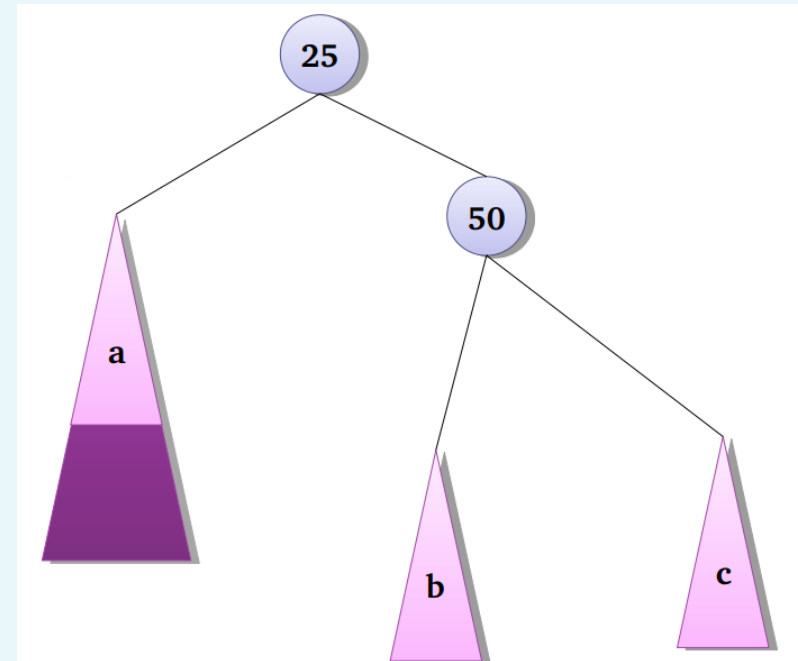
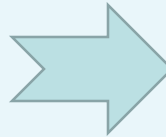
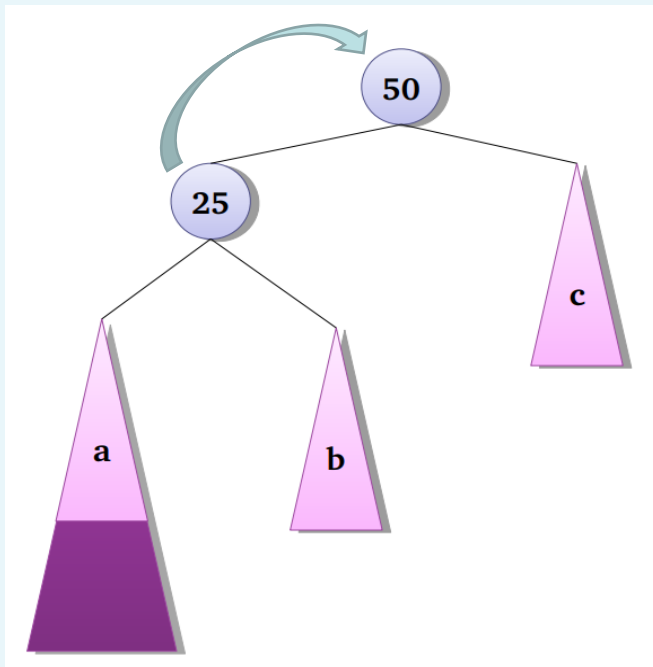
- Trường hợp 4: Cây mất cân bằng bên trái của con phải (L-R)





Các quy tắc xử lý khi cây mất cân bằng

- Đối với trường hợp 1 (L-L) ta thực hiện quay đơn qua phải (right rotate) như sau:

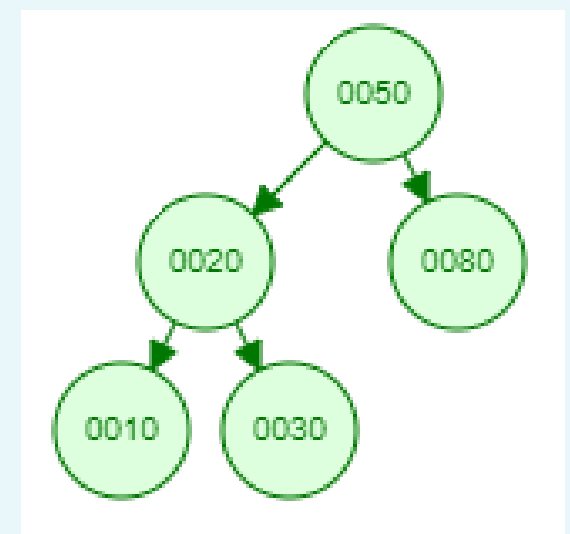
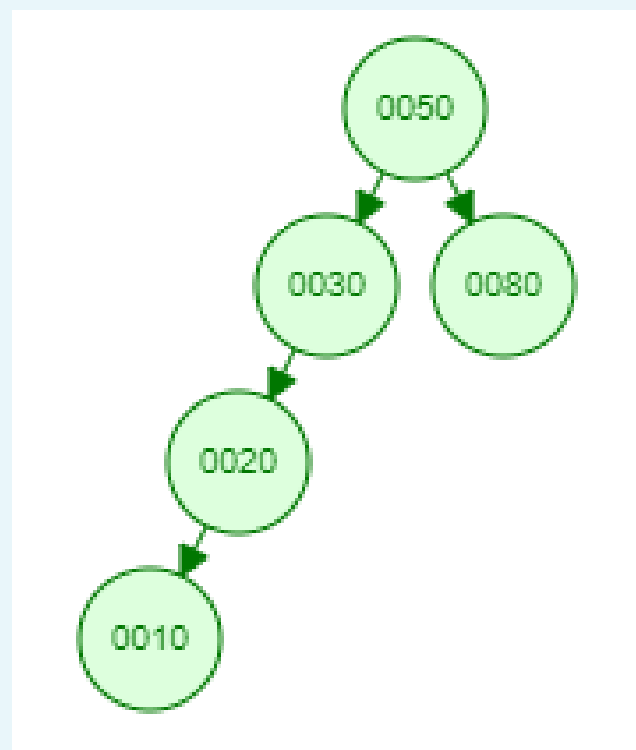
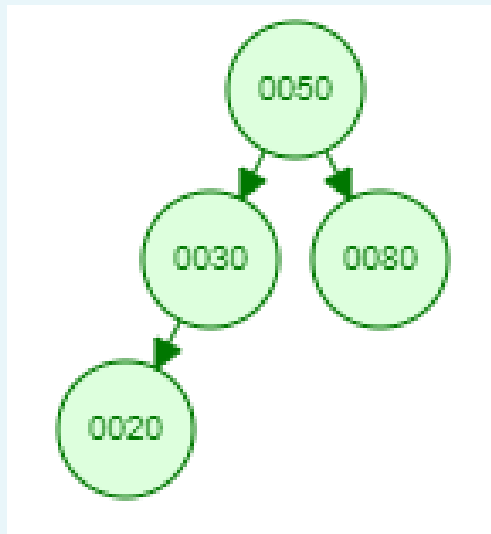




CANTHO UNIVERSITY

Ví dụ

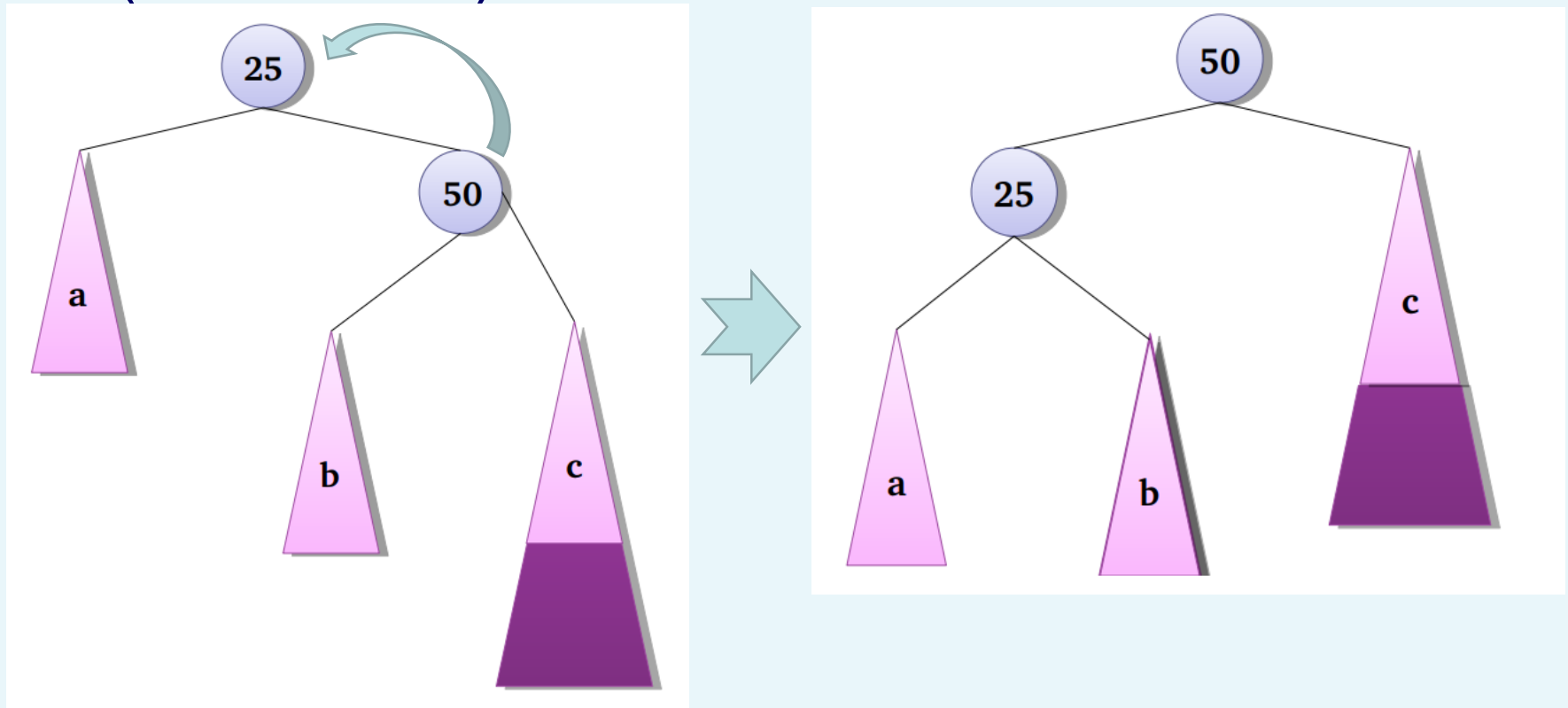
- Thêm nút 10





Các quy tắc xử lý khi cây mất cân bằng

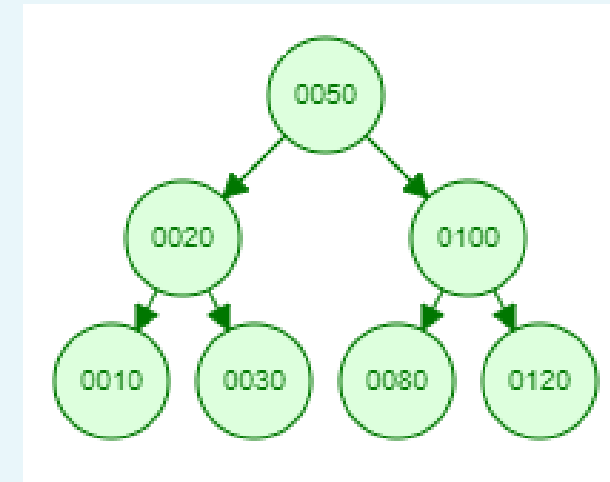
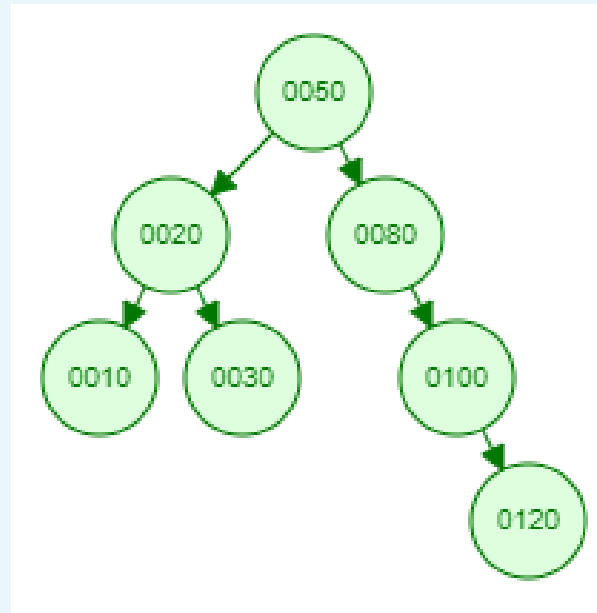
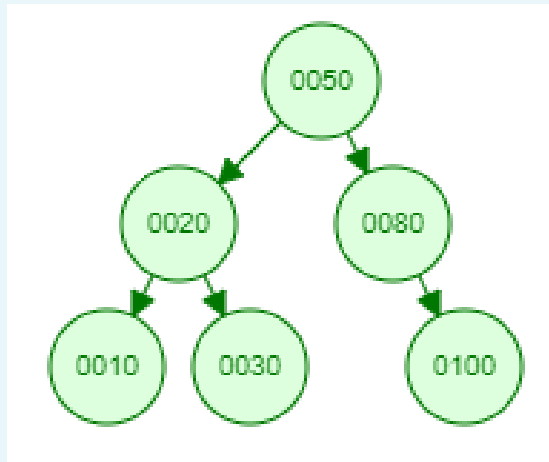
- Trường hợp 2: Ta thực hiện quay đơn sang trái (Left Rotate) như sau:





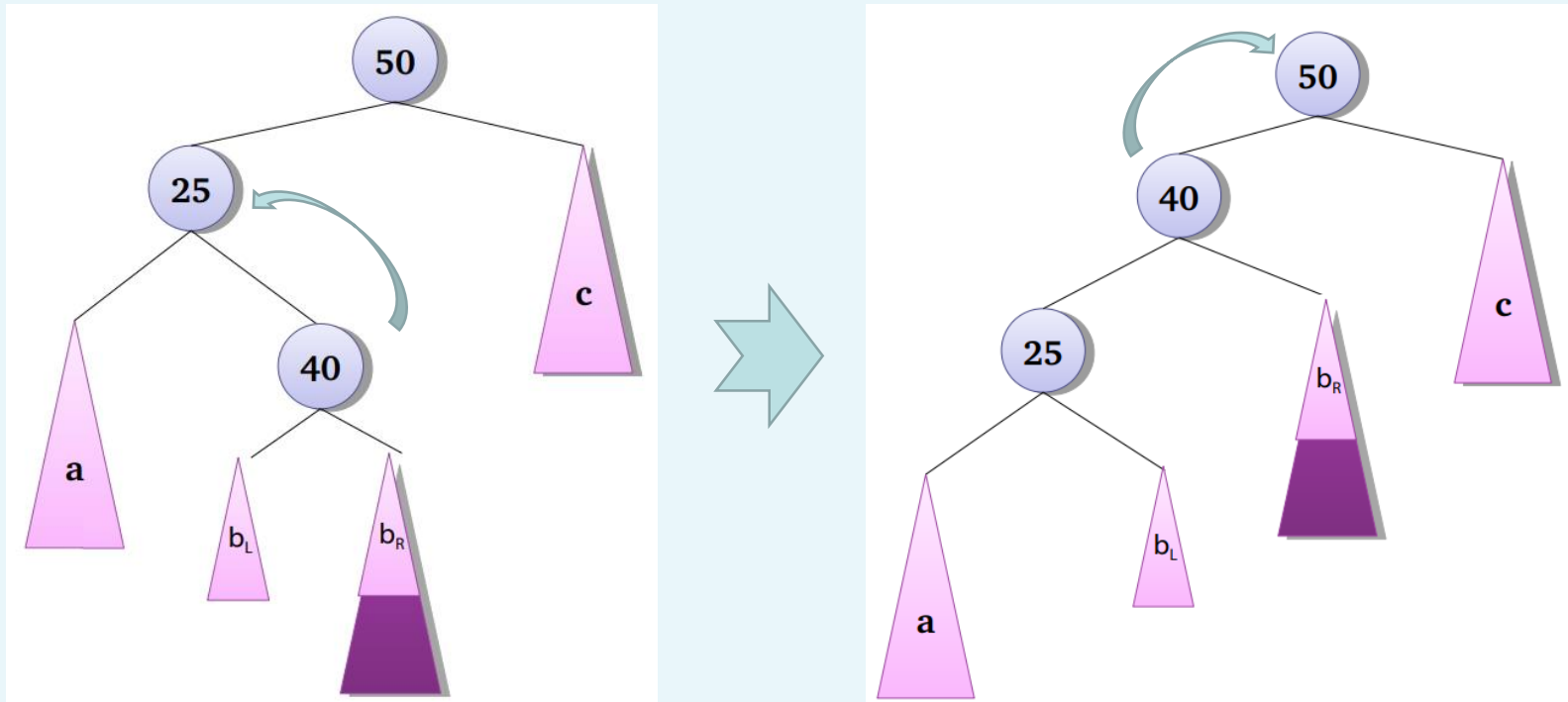
Ví dụ

- Thêm nút 120



Các quy tắc xử lý khi cây mất cân bằng

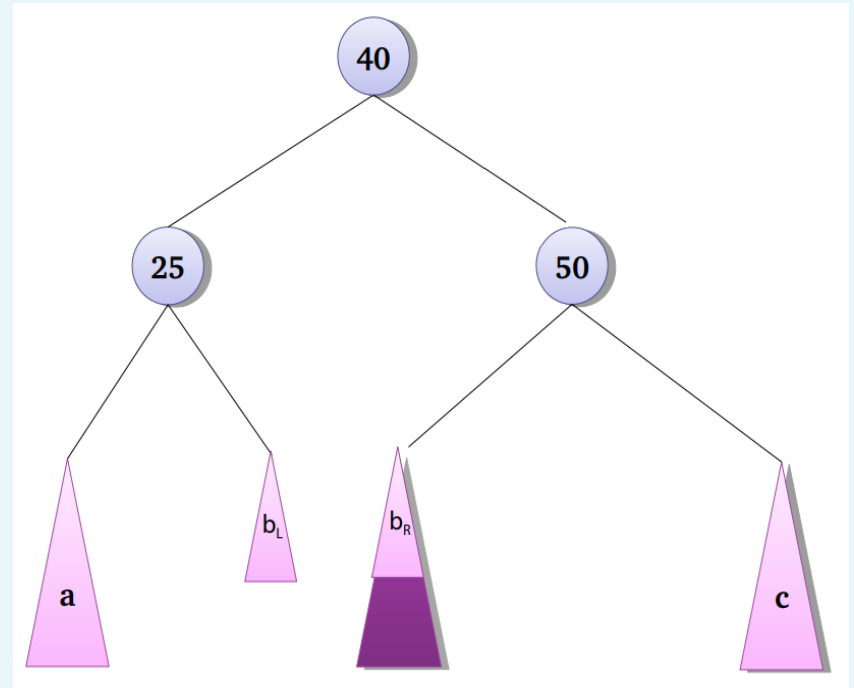
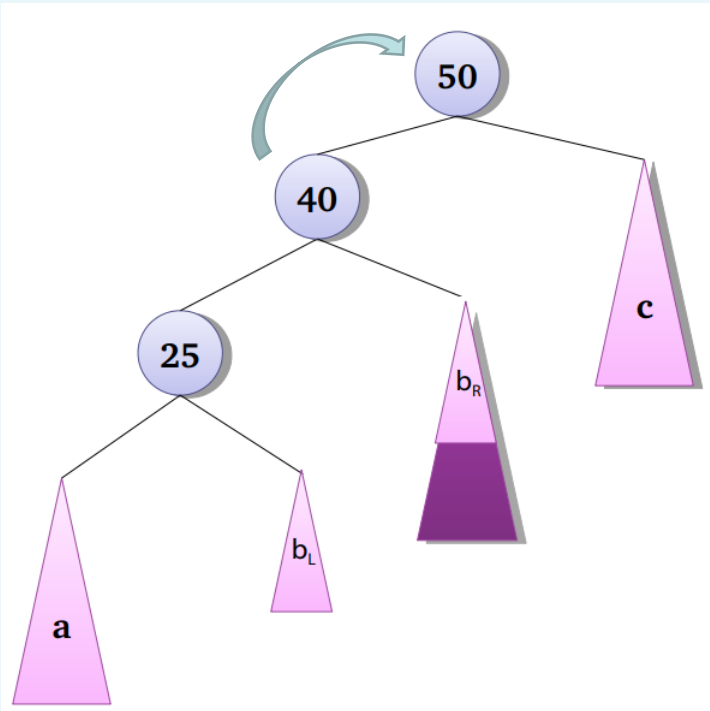
- Trường hợp 3: Cây mất cân bằng bên phải của con trái ta thực hiện xoay kép trái phải (left-right rotate)





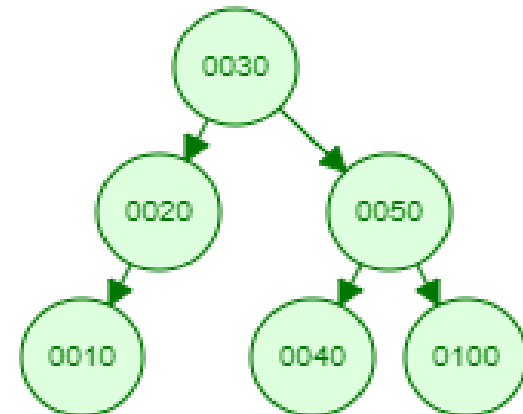
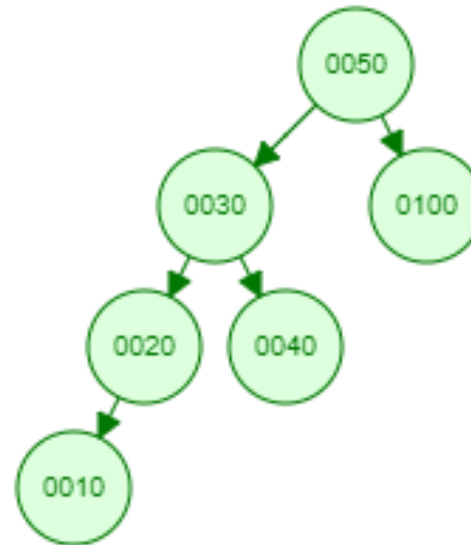
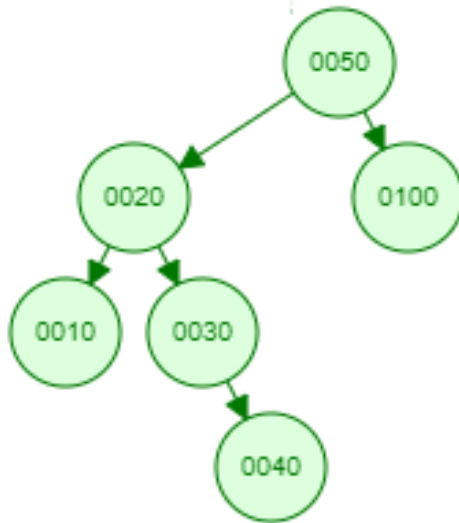
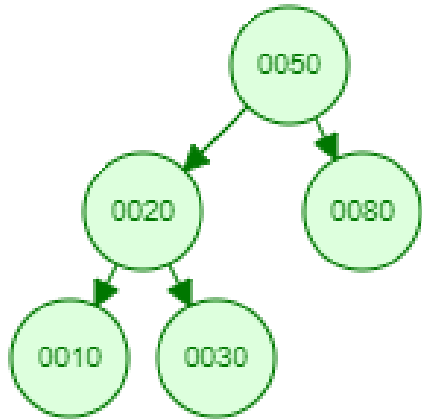
CANTHO UNIVERSITY

Các quy tắc xử lý khi cây mất cân bằng





Ví dụ: Thêm nút 40 vào cây

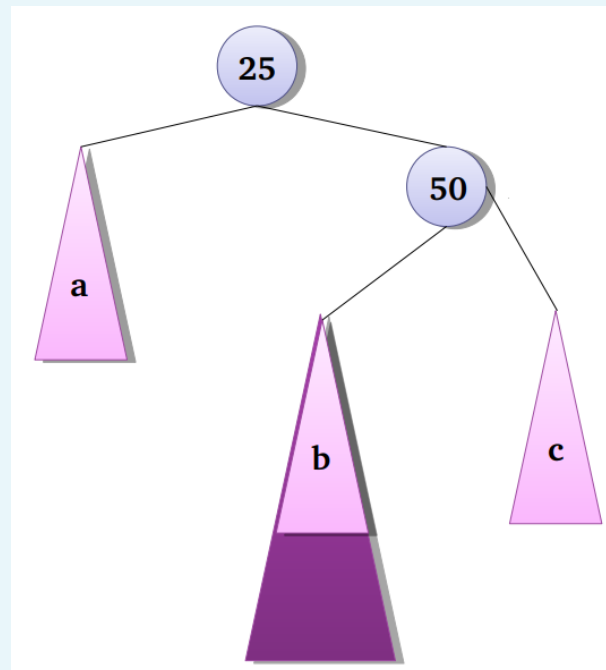




CANTHO UNIVERSITY

Các quy tắc xử lý khi cây mất cân bằng

- Trường hợp 4: Cây mất cân bằng bên trái của con phải. Trường hợp này ta thực hiện quay kép phải-trái như sau:

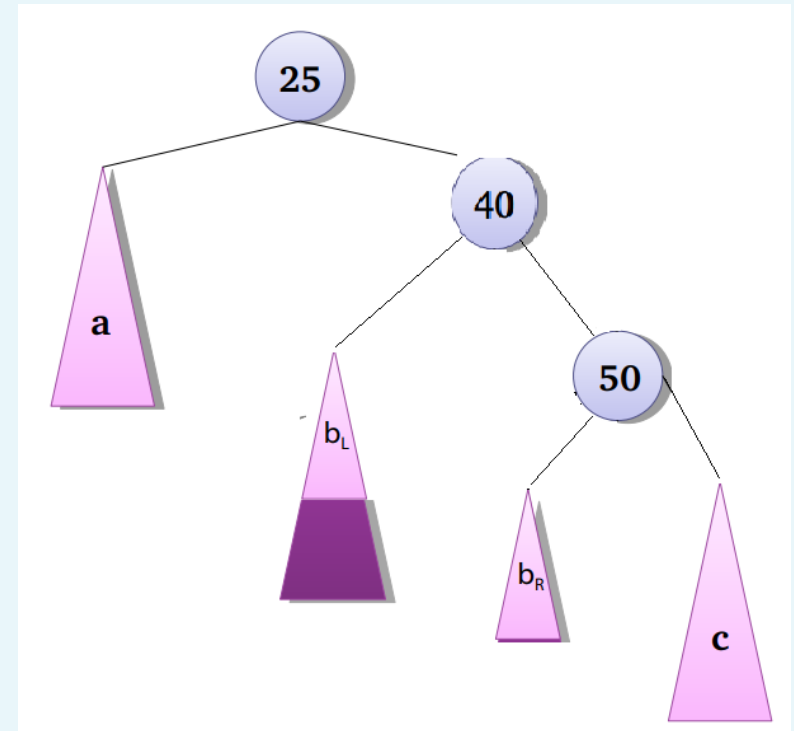
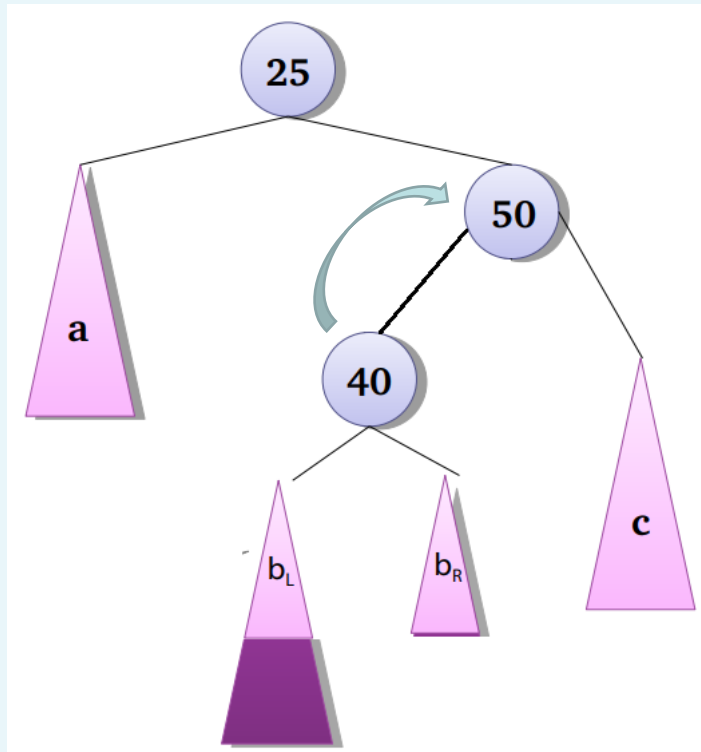




CANTHO UNIVERSITY

Các quy tắc xử lý khi cây mất cân bằng

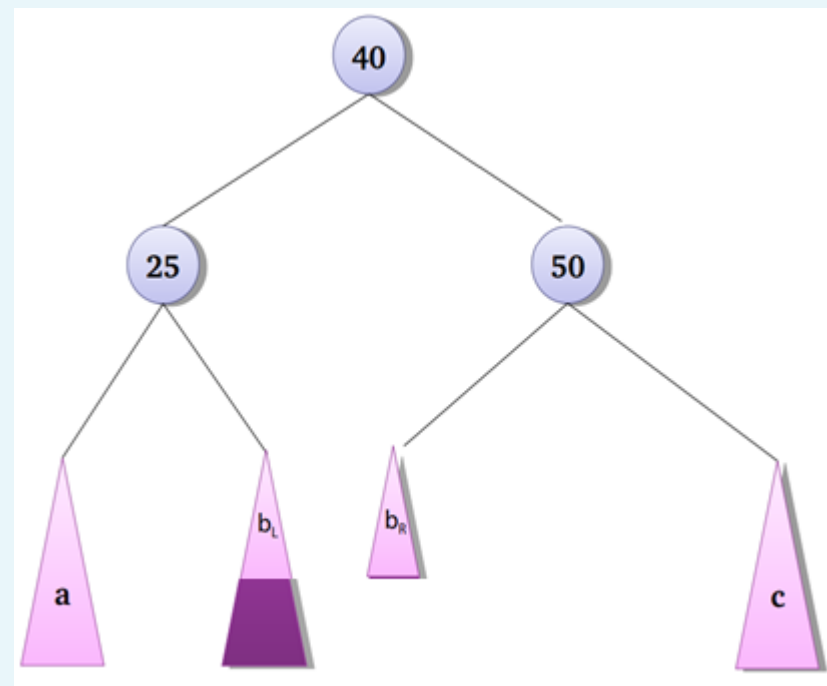
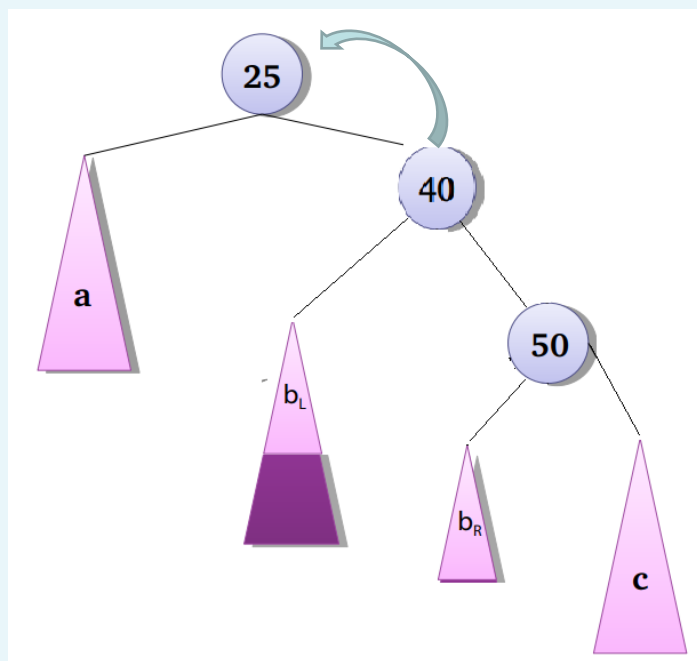
- Trường hợp 4 ta thực hiện quay kép Phải- Trái





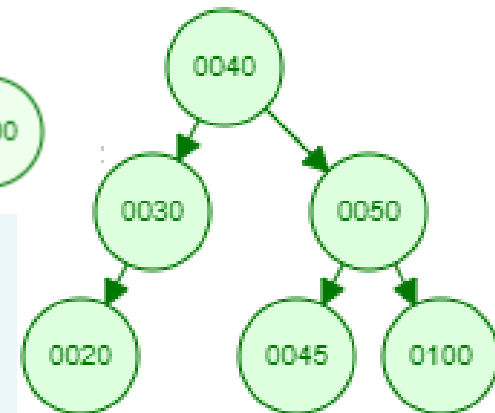
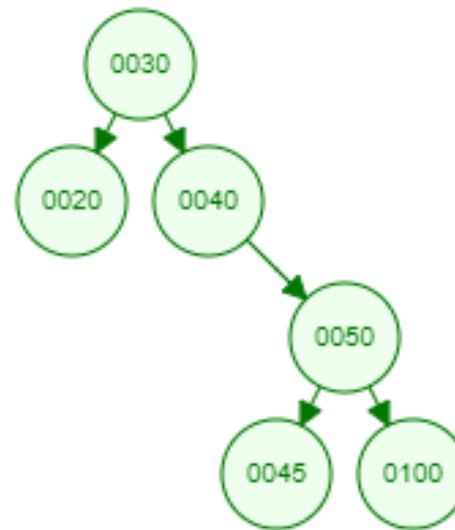
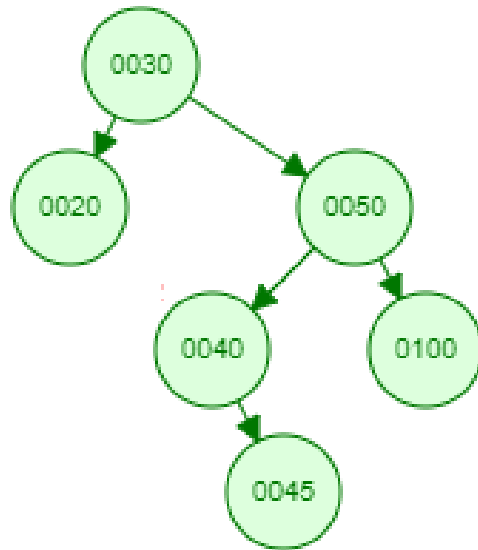
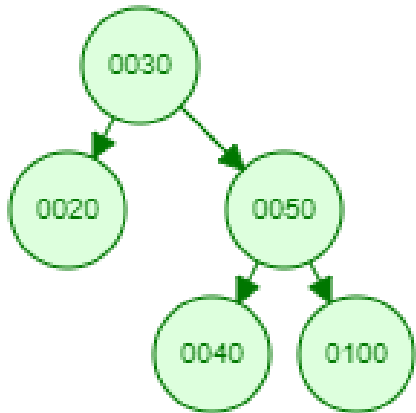
CANTHO UNIVERSITY

Các quy tắc xử lý khi cây mất cân bằng





Ví dụ: Thêm nút 45

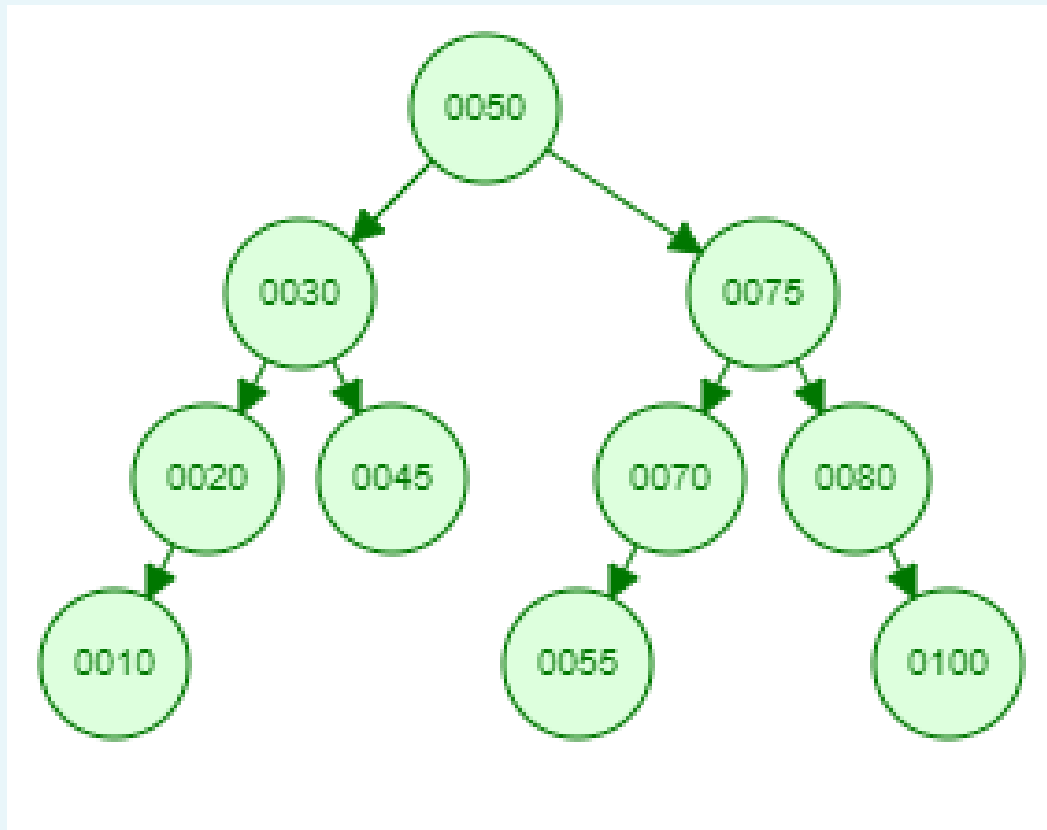




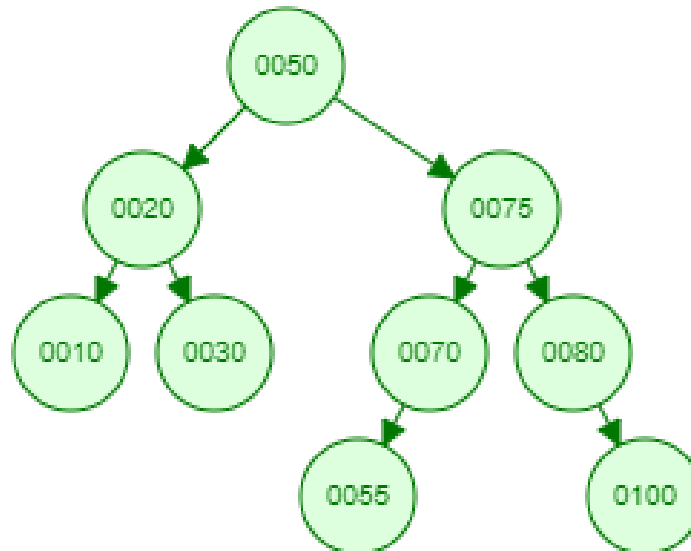
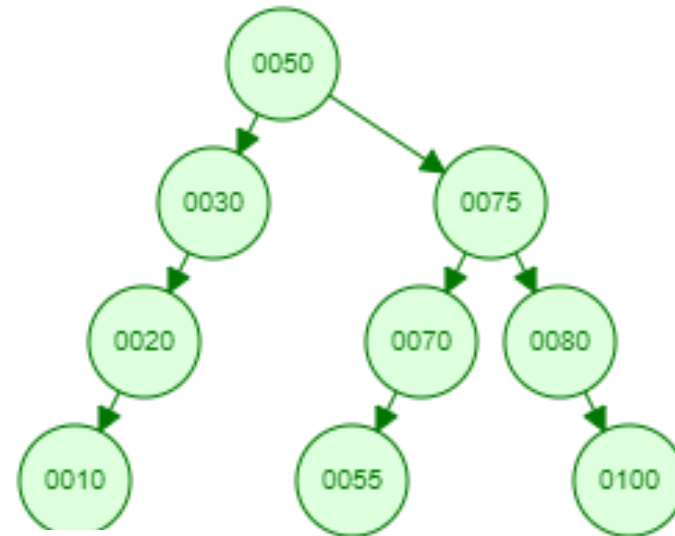
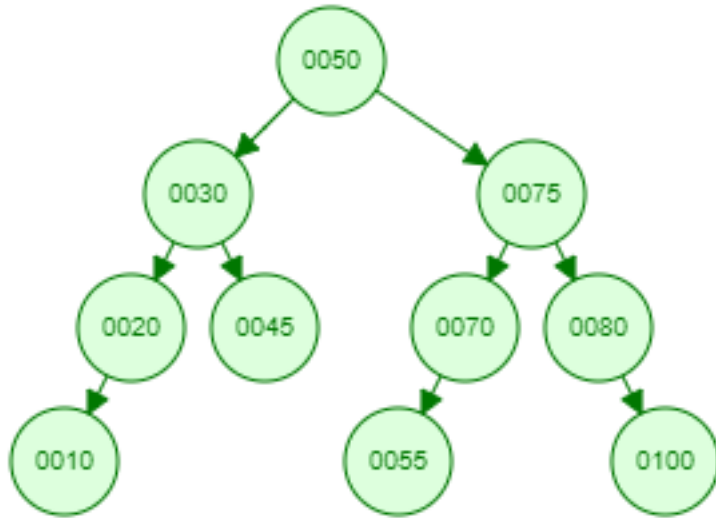
CANTHO UNIVERSITY

Vẽ cây AVL cho bởi danh sách

50, 70, 30, 10, 20, 45, 80, 75, 100, 55

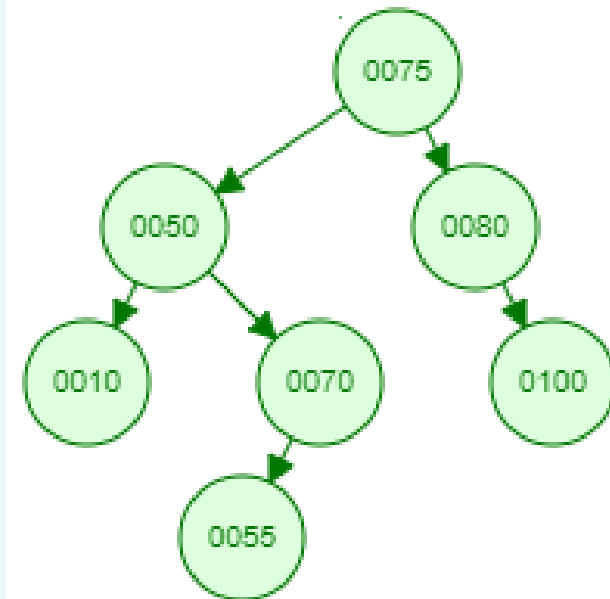
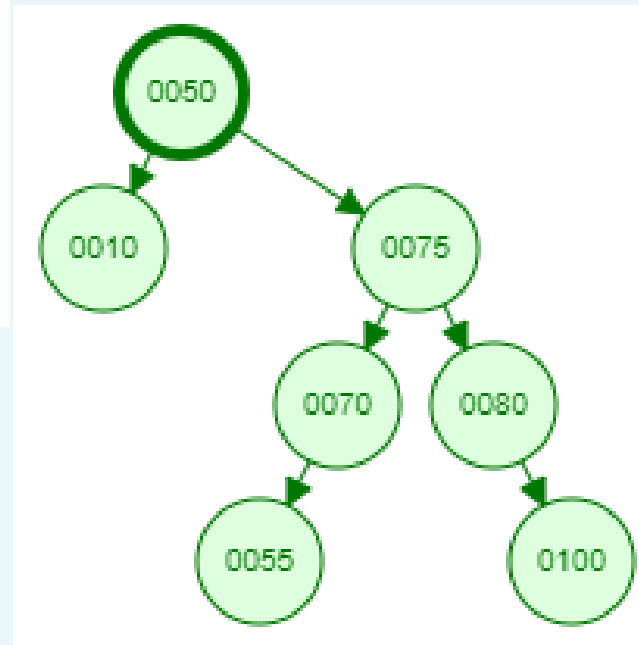
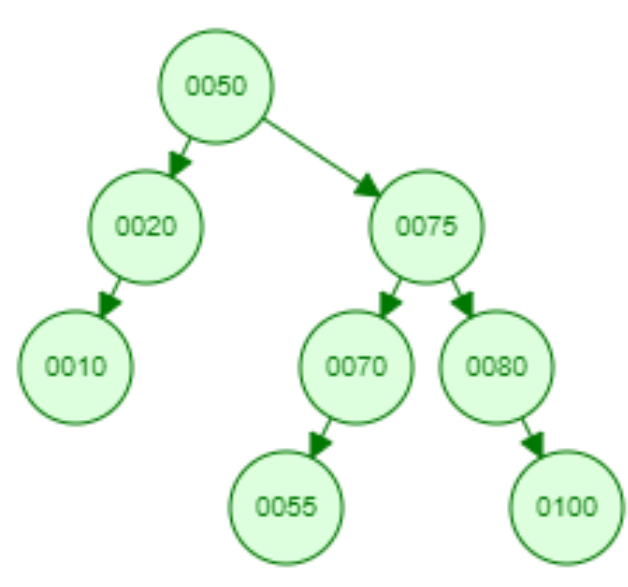


Xóa nút 45



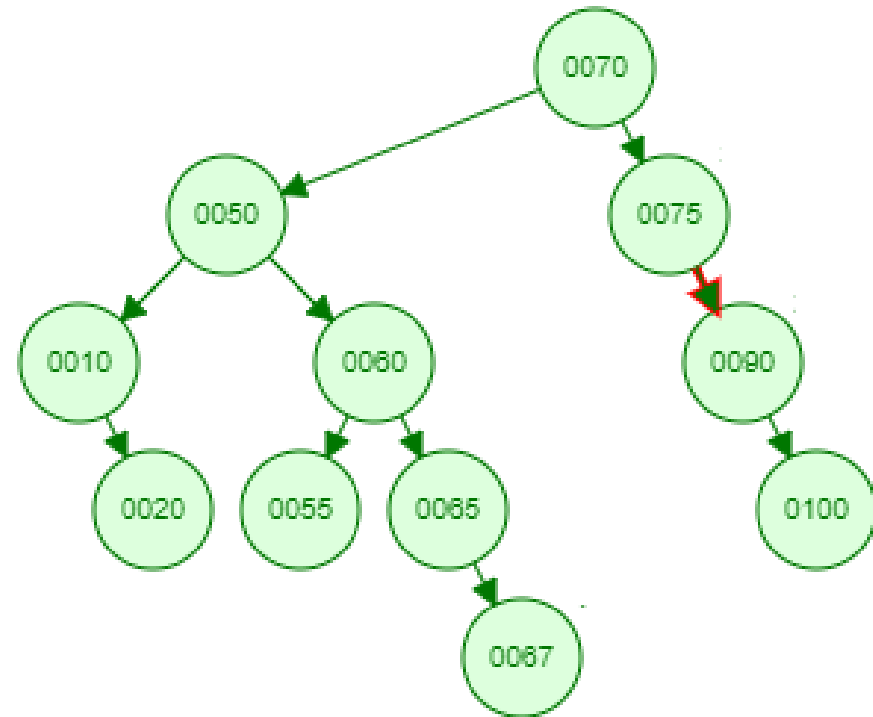
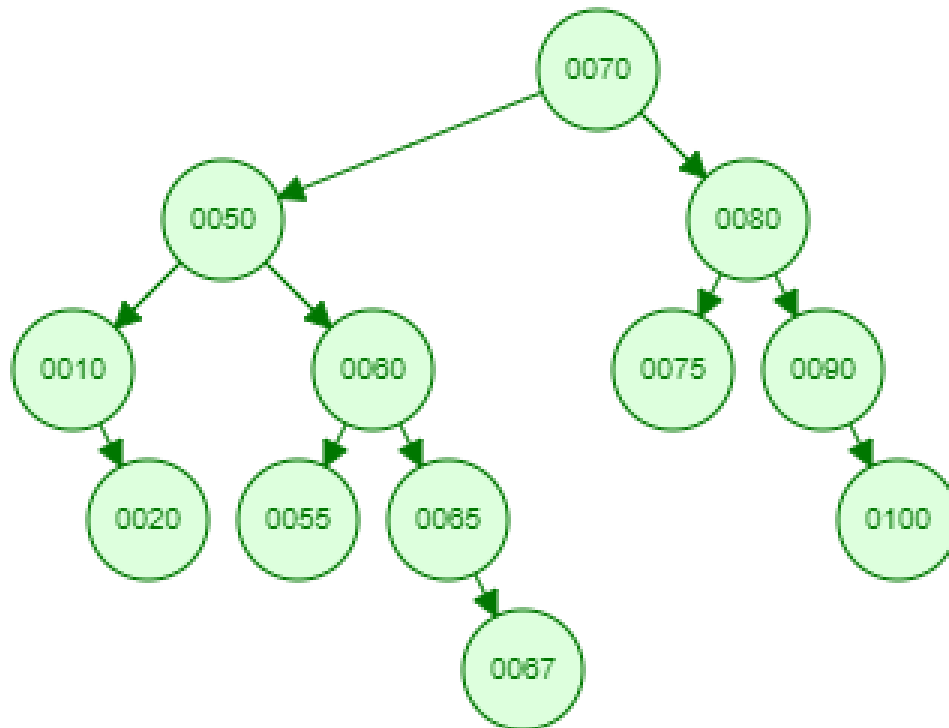


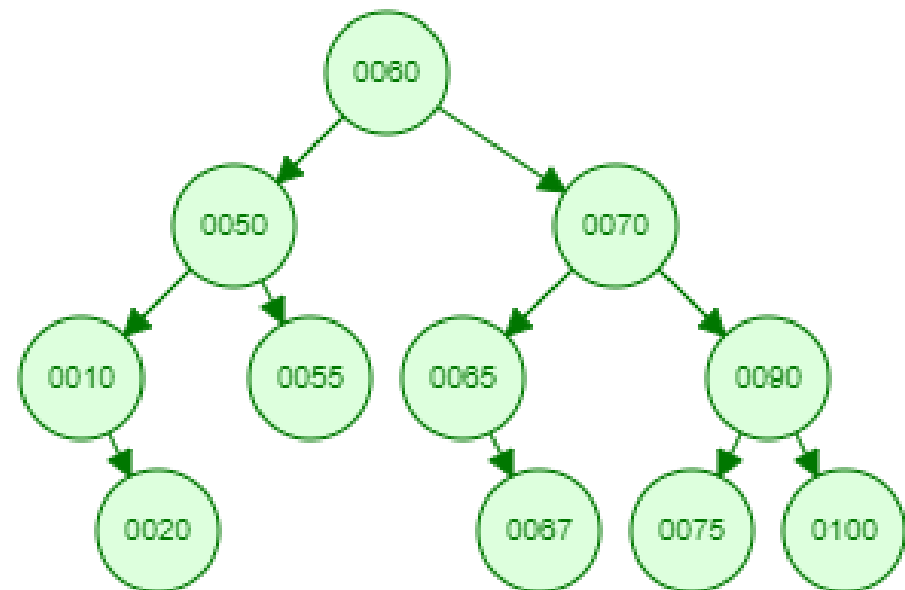
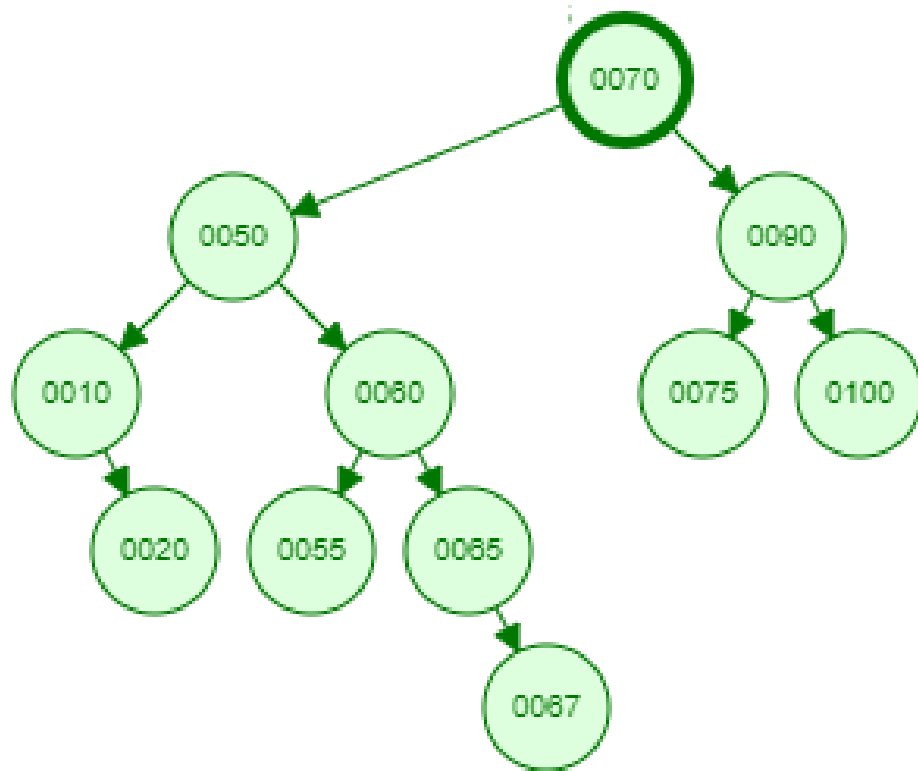
Xóa nút 20 trên cây





Xóa nút 80







Ý tưởng cài đặt cây

Mỗi nút trên cây cần có các thông tin:

- Khóa của nút
- Cây con trái
- Cây con phải
- Chỉ số cân bằng với quy ước như sau:

Chỉ số là EH ($=0$) nếu cây cân bằng

Chỉ số là LH ($=1$) nếu cây lệch trái

Chỉ số là RH ($=2$) nếu cây lệch phải



Thêm nút vào cây AVL

- Thêm nút vào AVL tương tự như thêm nút vào cây TKNP thông thường. Sau đó ta xét xem cây có bị mất cân bằng hay không để thực hiện cân bằng lại cây theo các quy tắc xoay và cập nhật lại chỉ số cân bằng của các nút.



Xóa nút ra khỏi cây AVL

- Xóa nút ra khỏi AVL tương tự như xóa nút ra khỏi cây TKNP thông thường. Sau đó ta xét xem cây có bị mất cân bằng hay không để thực hiện cân bằng lại cây theo các quy tắc xoay và cập nhật lại chỉ số cân bằng của các nút.
- Quá trình xóa nút có thể dẫn đến cây mất cân bằng dây chuyền liên tiếp đến các nút tiền bối của nó. Theo đó, nguyên tắc thực hiện là phải cân bằng từ nút con trước rồi đến lên nút cha.



Khai báo cây AVL

```
typedef ... KeyType;
typedef struct Node
{
    KeyType Key;
    int Bal;
    Node *Left;
    Node *Right;
};
typedef struct Node * AVLTree;
```



Tài liệu tham khảo

- [Adam Drozdek, Data structures and Algorithms Analysis in C++ 4th Edition, Cengage Learning, 2012](#)
- [Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest & Clifford Stein, MIT Press, 2012](#)
- <https://www.cs.usfca.edu/~galles/cs245S08/lecture/lecture23.pdf>
- <http://www.cse.chalmers.se/edu/year/2018/course/DAT037/slides/6c-avl-trees.pdf>
- <https://www.cs.usfca.edu/~galles/visualization/AVLtree.html>

Q&A?