CẤU TRÚC DỮ LIỆU NÂNG CAO

- Các kiến thức yêu cầu
- Tóm tắt nội dung môn học
- Phương pháp kiểm tra đánh giá
- Tài liệu tham khảo

TS Nguyễn Mạnh Hùng – BM CNPM

Các kiến thức yêu cầu

- Các thuật toán và cấu trúc dữ liệu cơ bản
- Ngôn ngữ lập trình: C++, nhưng không sử dụng class.
- Công cụ: VS C++, ứng dụng dạng console.

Tóm tắt nội dung môn học

- Cây cân bằng
- Cây đỏ đen (red black tree)
- Cây 2-3-4
- Interval Heap
- Priority Search Tree
- B-Tree
- Phương pháp phân tích khấu trừ
- Cấu trúc đóng nhị thức
- Cấu trúc đóng Fibonacci
- Cấu trúc các tập rời nhau

Ứng dụng

- Ứng dụng trong đánh chỉ mục các CSDL lớn.
- Ứng dụng xây dựng thuật toán tốc độ cao định tuyến gói tin trong router.
- Ứng dụng khai phá dữ liệu hiện năng cao trong CSDL lớn, nhiều chiều.
- Ứng dụng trong xử lý dữ liệu không gian
- Ứng dụng trong xử lý dữ liệu Multimedia

Phương pháp kiểm tra đánh giá

- Điểm chuyên cần: 10%
- Kiểm tra giữa học phần (bắt buộc): 20%
- Thi kết thúc học phần (bắt buộc): 70%

Tài liệu tham khảo

- Slides bài giảng môn học
- Advanced Data Structures. Cambridge University Press - 2008; PETER BRASS.
- Introduction to Algorithms. McGraw Hill - 1990; Thomas H. C., Charles E.L., and Ronald L.R.
- Giáo trình thuật toán. Thống kế 2002. Nhóm Ngọc Anh Thư dịch
- Data Structures, Algorithms, and Object-Oriented Programming; McGraw Hill; Gregory Heilleman - 1996.
- Algorithms and Data Structures in C++; Tác giả Alan Parker,1993

Lecture 1: Cây cân bằng

- Cây nhị phân tìm kiếm (NPTK) có thể bị suy biến thành danh sách tuyến tính.
- Khi đó thời gian tìm kiếm là O(N). Để tăng tốc độ tìm kiếm thì chiều cao của cây phải nhỏ.
- Để tăng tốc độ tìm kiếm thì chiều cao của cây NPTK phải là O(log N). Các cây NPTK như thế gọi là balanced binary search trees.
- Examples are AVL tree, red-black tree.

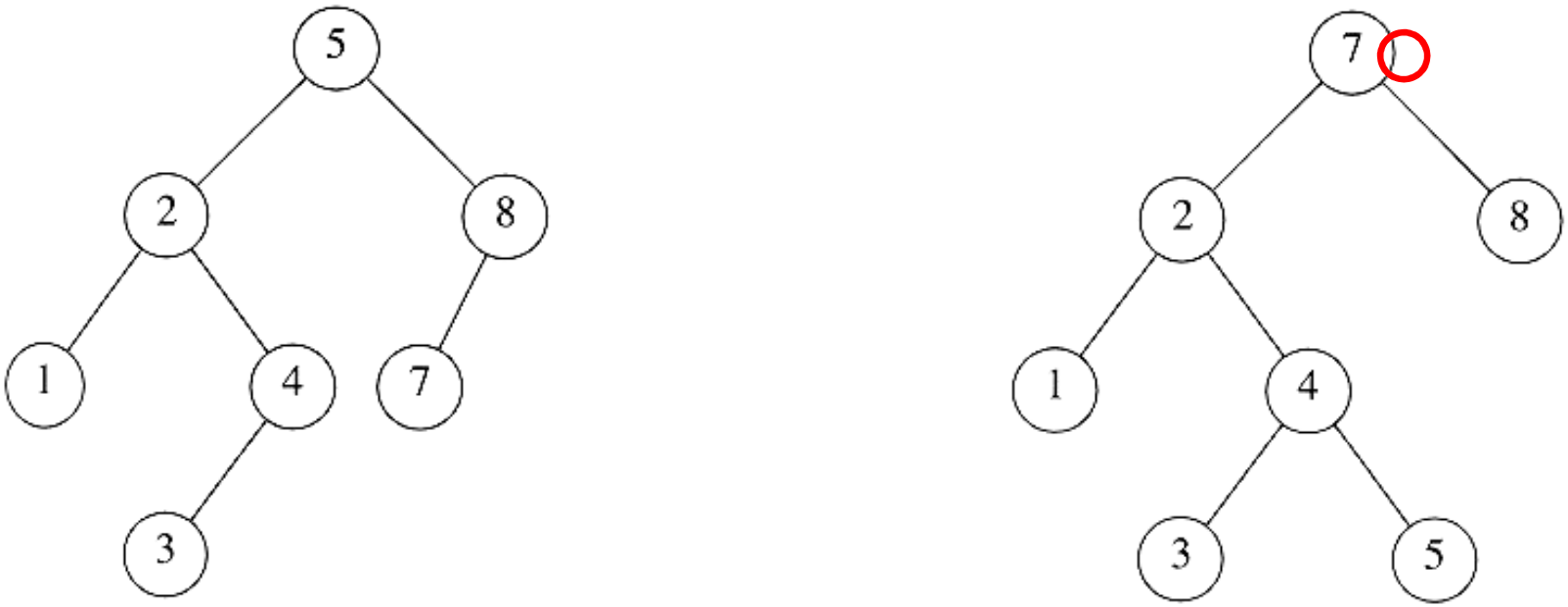
AVL Tree (Adelson-Velskii,Landis 1962)

Chiều cao của nút:

- Chiều cao của nút lá bằng 1
- The height of an internal node is the maximum height of its children plus 1

AVL Tree

- Cây AVL là cây NPTK trong đó :
  - Mọi nút trên cây thì chiều cao của cây con trái và chiều cao của cây con phải chênh lệch nhau không quá 1



Two binary search trees. Only the left tree is AVL.

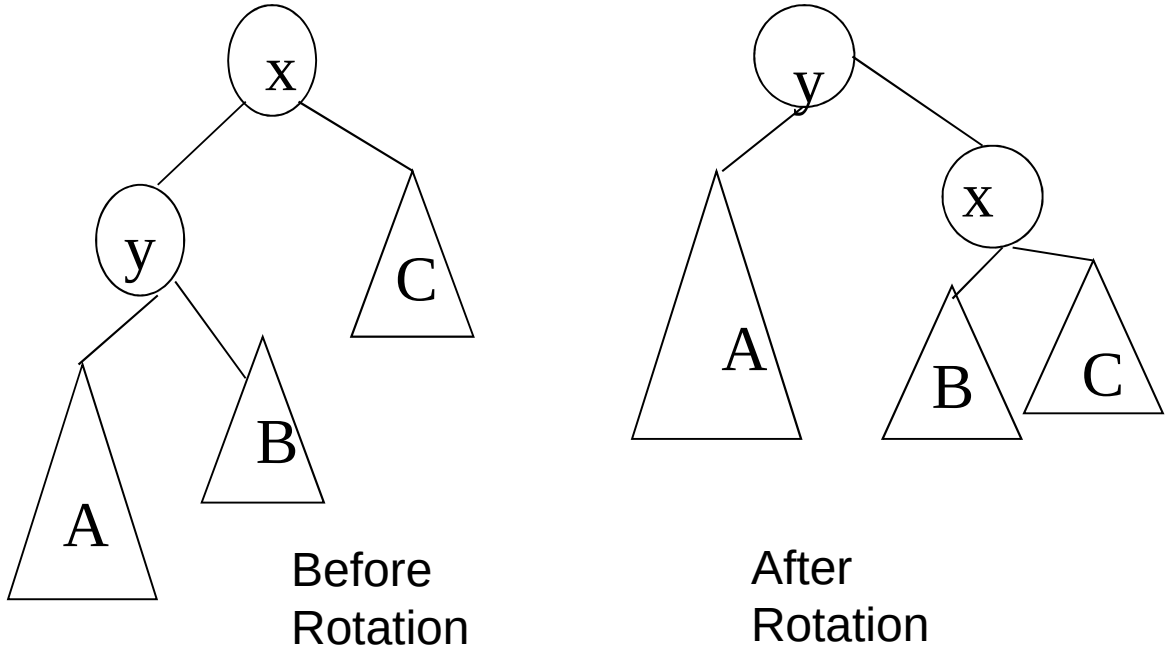
AVL Tree

- X gốc của cây AVL có chiều cao h.
- N_h số nút tối thiểu trên cây chiều cao h.
- Theo định nghĩa, ta có

$$N_h \geq N_{h-1} + N_{h-2} + 1$$
$$\geq 2N_{h-2} + 1$$
$$> 2N_{h-2}$$
$$N_h > 2^i N_{h-2*i} \quad \text{i=(h-1)/2;}$$
$$N > 2^{(h-1)/2}$$
$$\log N > (h-1)/2$$
$$(\log N + 1) * 2 > h$$

Các phép quay

- Khi thực hiện các phép toán (insertion or deletion), chúng ta cần phải biến đổi cây để đảm bảo tích chất cân bằng.
- Có hai dạng quay single rotations or double rotations.

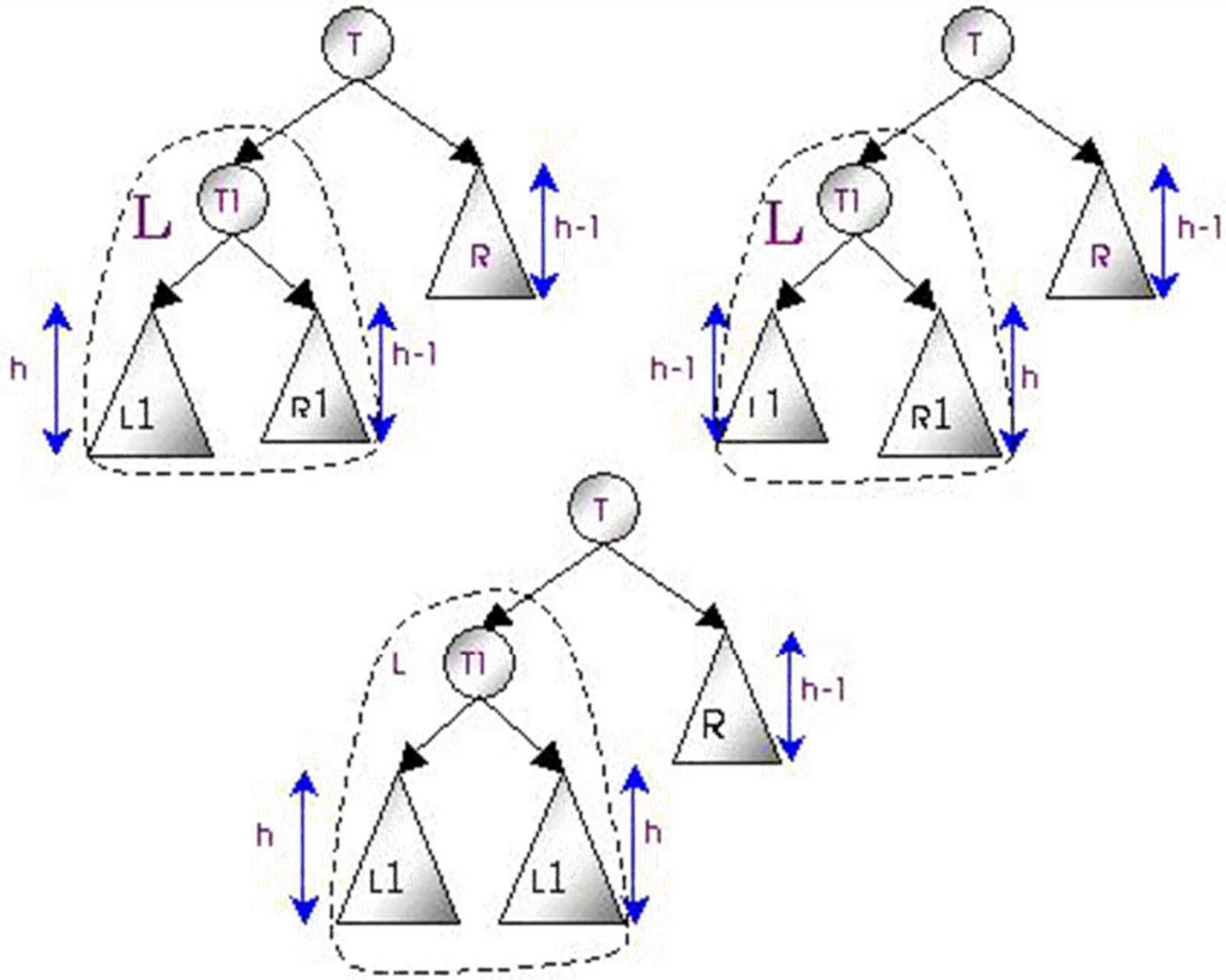


e.g. Single Rotation

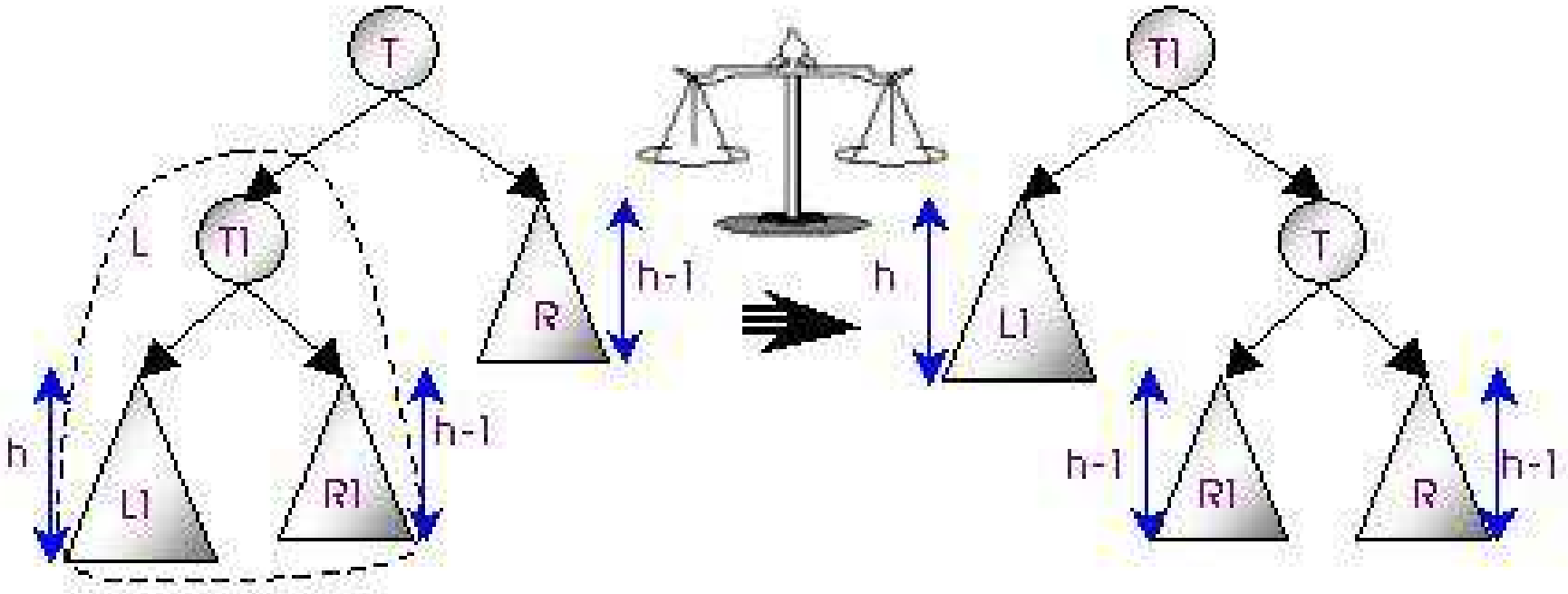
Các phép quay (cont.)

- Khi thêm/xóa có thể làm thay đổi chiều cao của cây.
- Khi đó điều kiện cân bằng có thể bị vi phạm. Chẳng hạn tại nút x sự khác nhau của left(x) và right(x) là 2.
- Áp dụng các phép quay tại nút x để hiệu chỉnh cây.

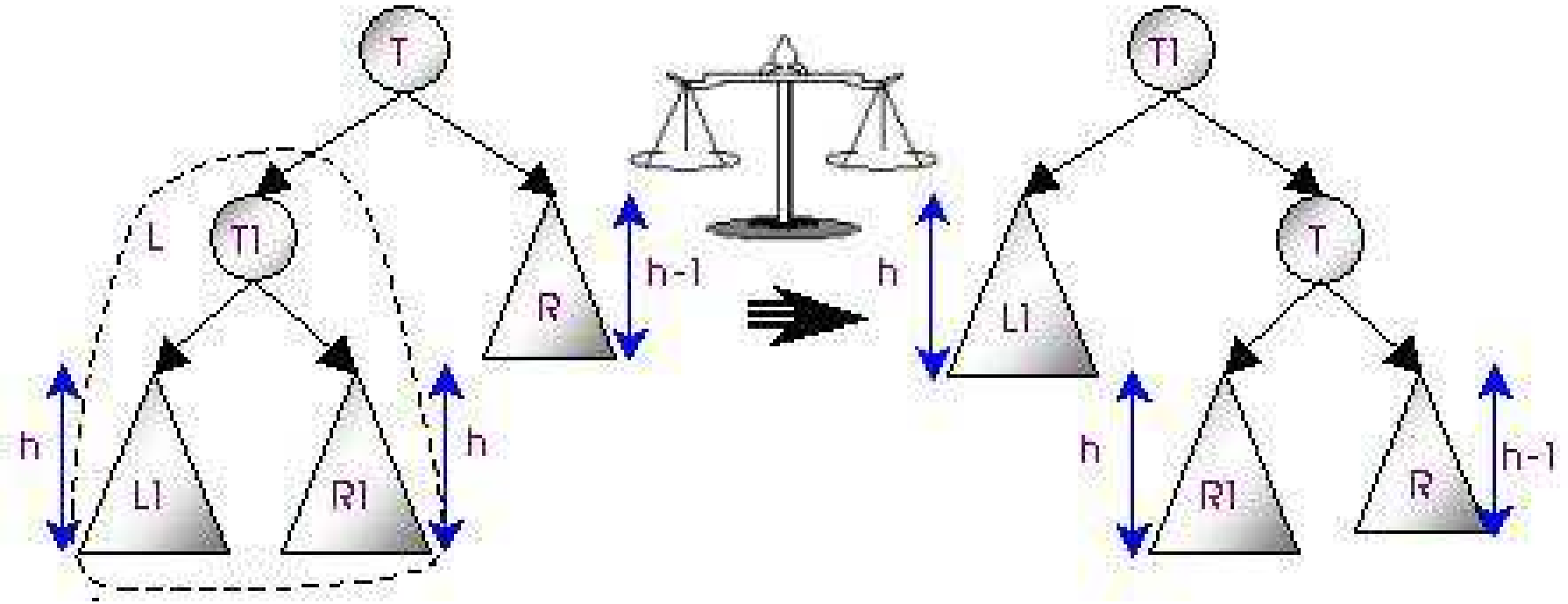
Case 1: Cây lệch trái



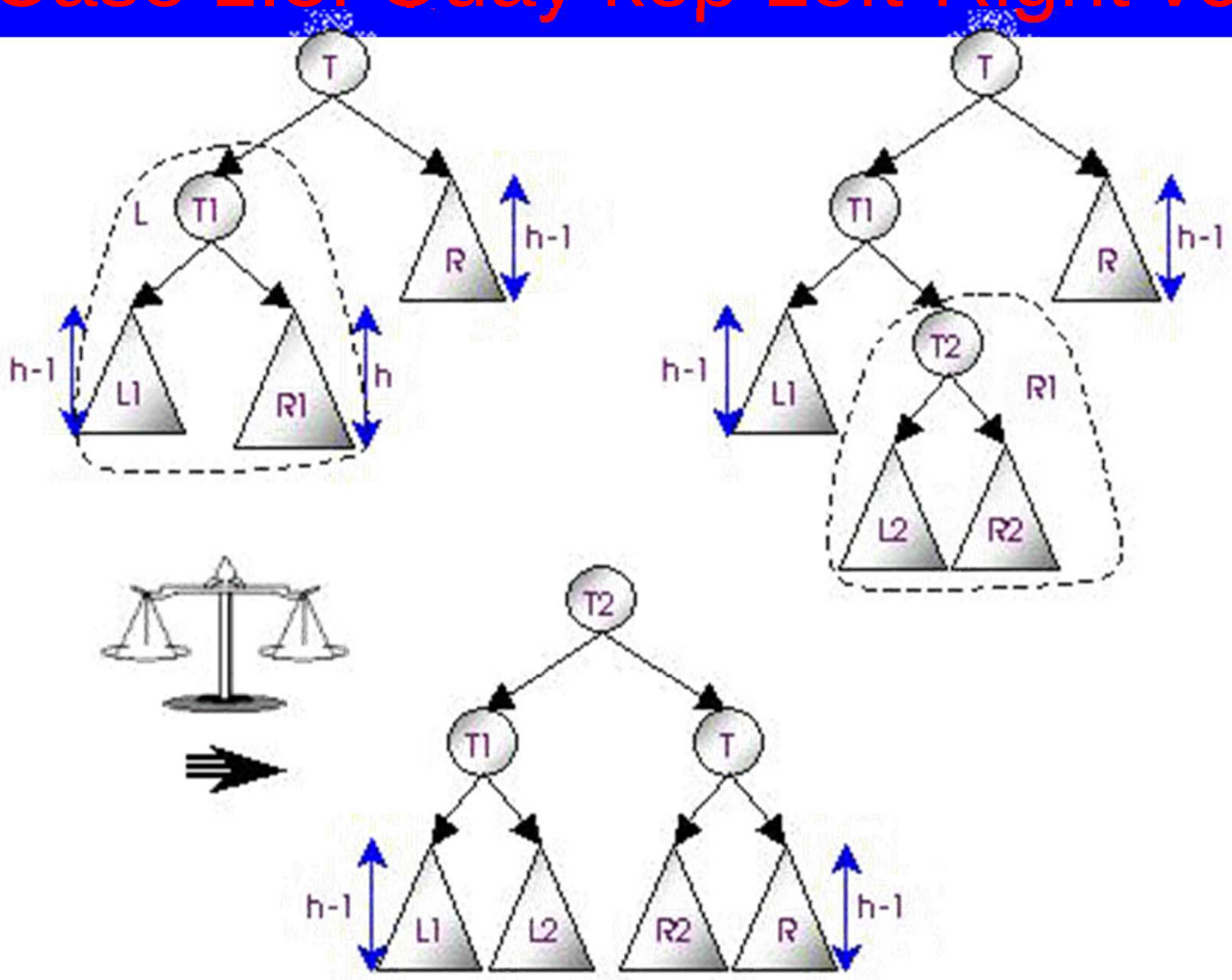
Case 1.1: Quay đơn phải T1 quanh T



Case 1.2: Quay đơn phải T1

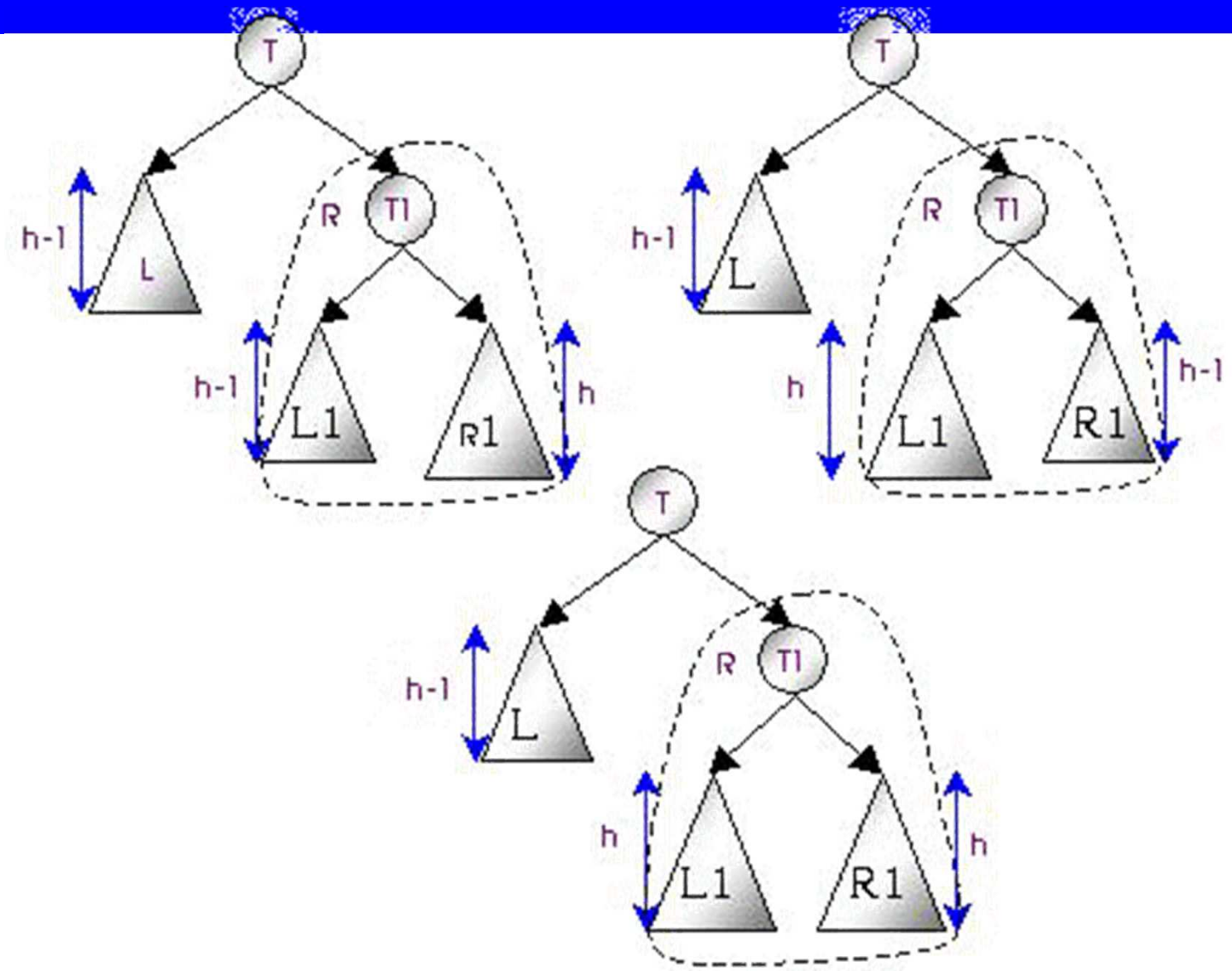


Case 1.3: Quay kép Left-Right với T2



Quay trái T2 quanh T1, Quay phải T2 quanh T

Case 2: Cây lệch phải



Case 2: Cây lệch phải (Cont.)

- Các trường hợp cây bị lệch phải ta thực hiện các phép quay đơn và quay kép tương tự như trương hợp cây lệch trái nhưng theo chiều ngược lại.

Insertion

- Tiến hành tương tự như thêm khóa trong CNPTK.
- Kiểm tra các nút trên đường dẫn từ nút mới thêm đến gốc. Với mỗi nút x bắt gặp trên path kiểm tra left(x) và right(x). Nếu đã thỏa mãn đk cân bằng thì bỏ qua và xét tiếp nút ở trên. Nếu không thỏa mãn thì phải tiến hành quay đơn hoặc kép.
- Với phép thêm nút, phải quay tối đa một lần (hoặc quay đơn hoặc quay kép)

Insertion

- Giả sử x là nút có sự chênh lệnh của left(x) và right(x) lớn hơn 1 (bằng 2).
- Giả sử chiều cao của x là h+3
- Có 4 TH sau:
  - Chiều cao của left(x) là h+2 (height of right(x) is h)
    - H of left(left(x)) is h+1 $\Rightarrow$ single rotate with left child
    - H of right(left(x)) is h+1 $\Rightarrow$ double rotate with left child
  - Chiều cao của right(x) là h+2 (height of left(x) is h)
    - H of right(right(x)) is h+1 $\Rightarrow$ single rotate with right child
    - H of left(right(x)) is h+1 $\Rightarrow$ double rotate with right child

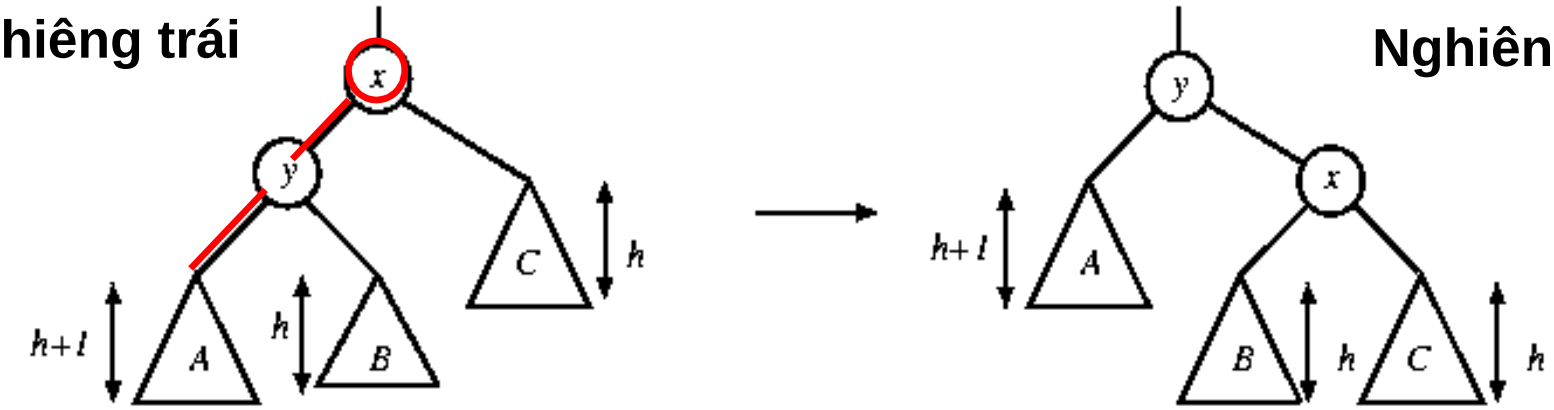
Single rotation

Thêm nút mới vào cây con A làm cho chiều cao của cây con A tăng lên 1 (bằng h+1).

ĐK cân bằng bị vi phạm tại nút x.

- chiều cao của left(x) is h+2
- chiều cao của right(x) is h.
- quay phải nút y quanh nút x.

Nghiêng trái



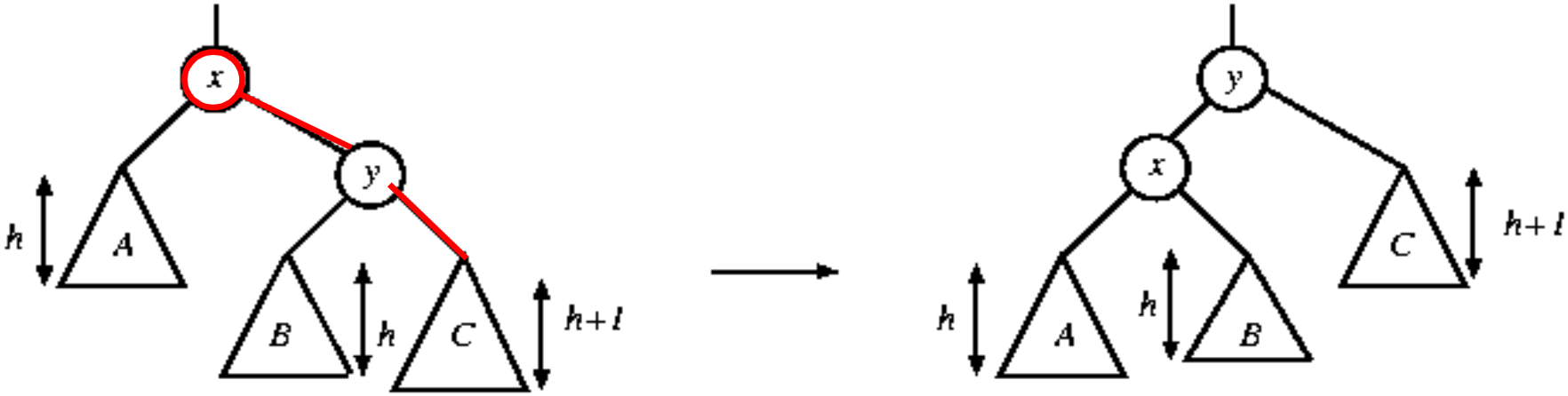
Nghiêng phải

Rotate with left child

Single rotation

Thêm nút vào cây con C.

ĐK cân bằng bị vi phạm tại nút x.

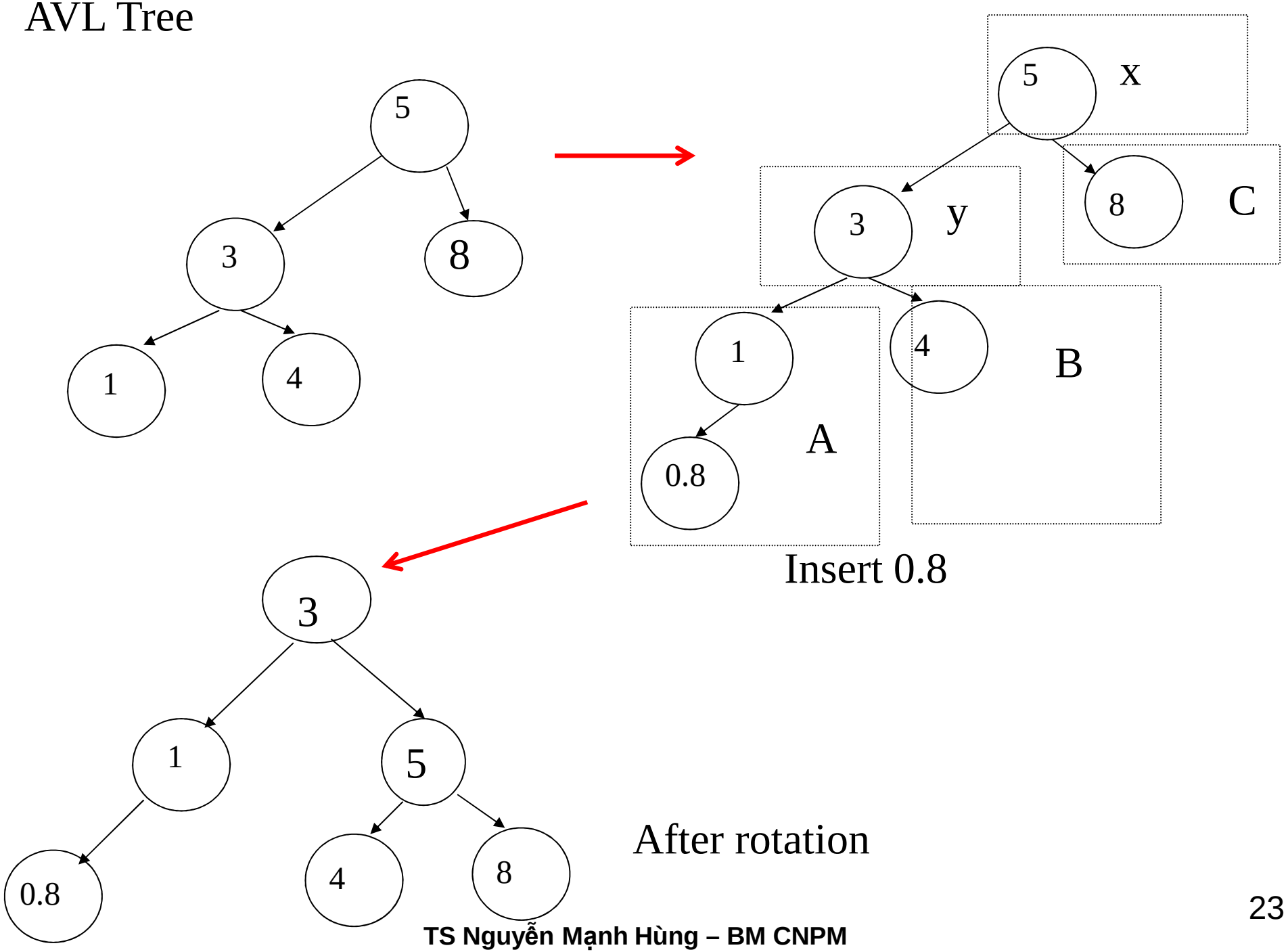


Rotate with right child

Single rotation takes O(1) time.

Insertion takes O(log N) time.

AVL Tree

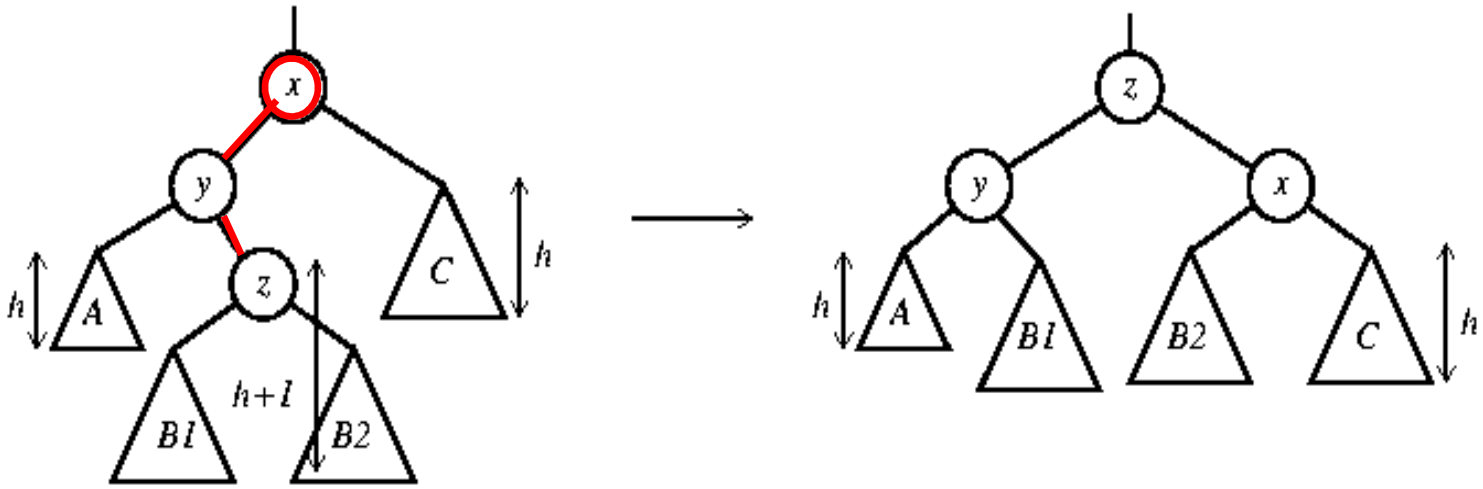


Double rotation

Thêm nút vào cây con B1 or B2.

ĐK cân bằng bị vi phạm tại nút x.

Quay trái z quanh y; quay phải z quanh x



Double rotate with left child

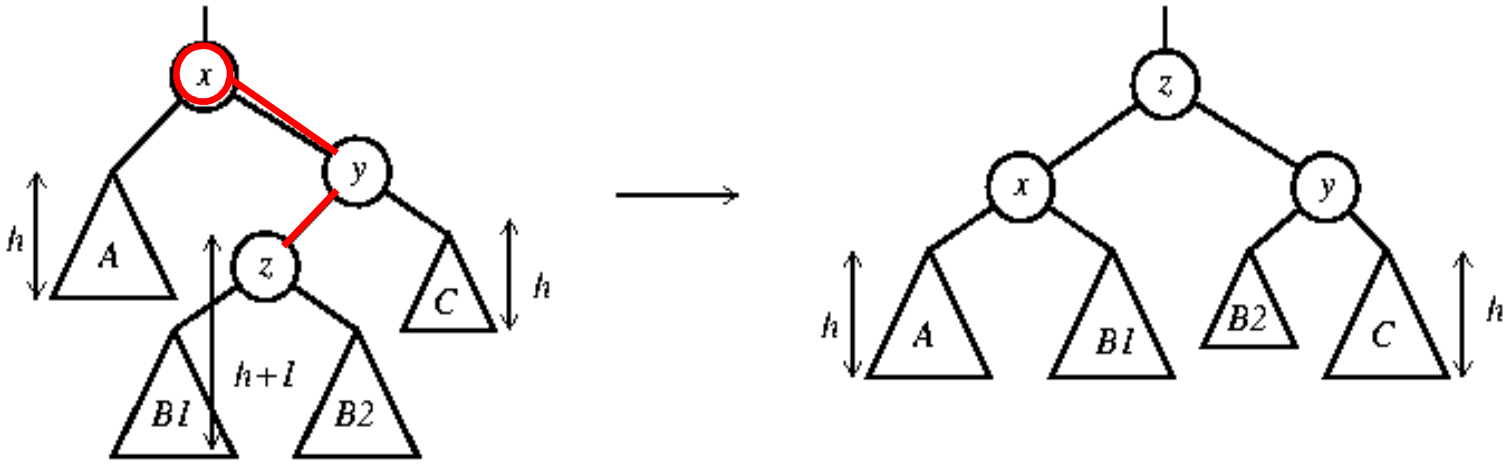
also called left-right rotate

Double rotation

Thêm nút vào cây con B1 or B2.

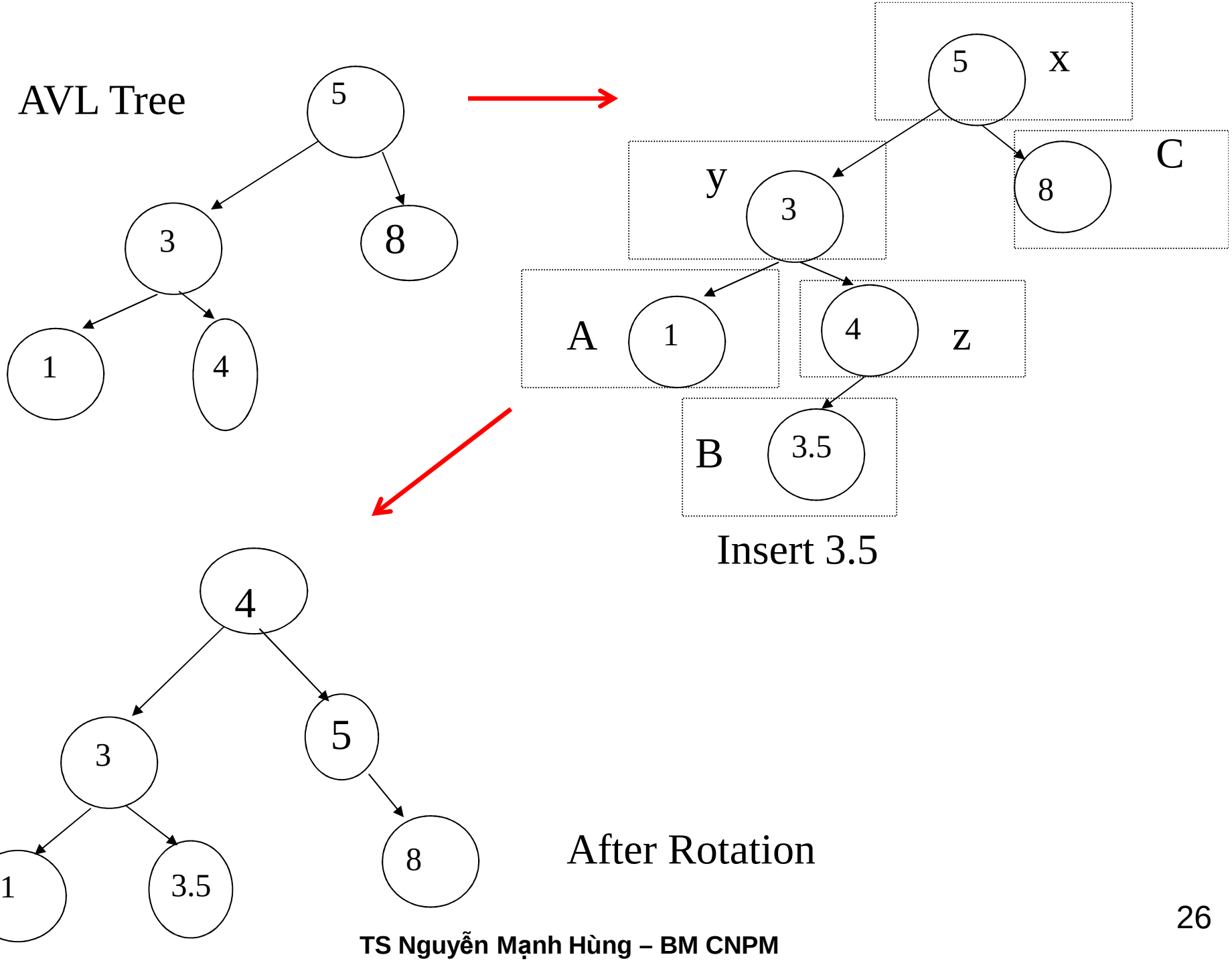
ĐK cân bằng bị vi phạm tại nút x.

Quay phải z quanh y; quay trái z quanh x



Double rotate with right child

also called right-left rotate


AVL Tree
5
3
8
1
4
5
x
y
3
8
C
A
1
4
z
B
3.5
Insert 3.5
4
3
5
1
3.5
8
After Rotation

An Extended Example

Insert 3,2,1,4,5,6,7, 16,15,14

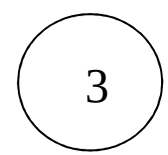


Fig 1

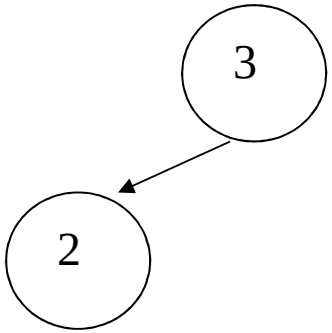


Fig 2

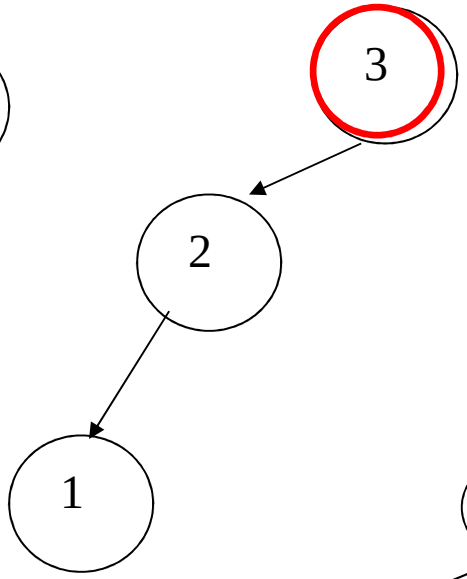


Fig 3

Single rotation

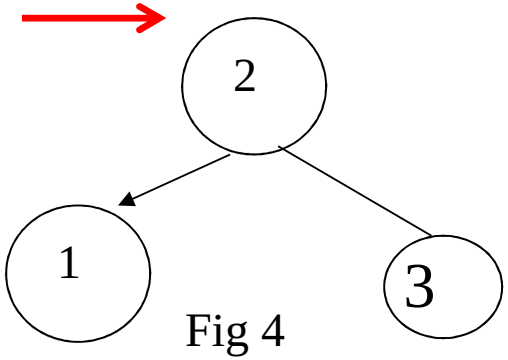


Fig 4

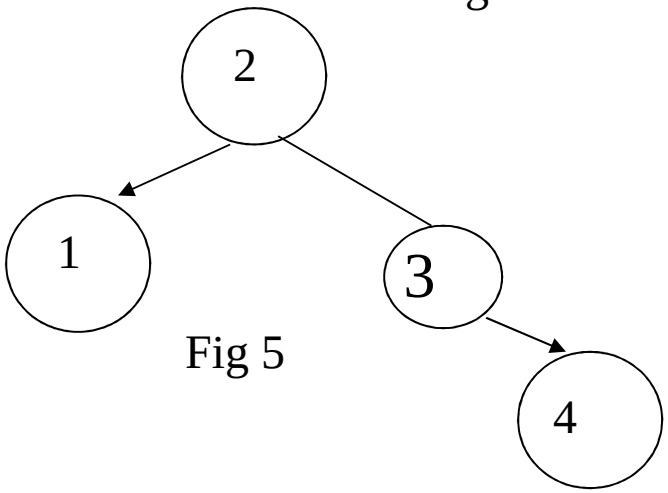


Fig 5

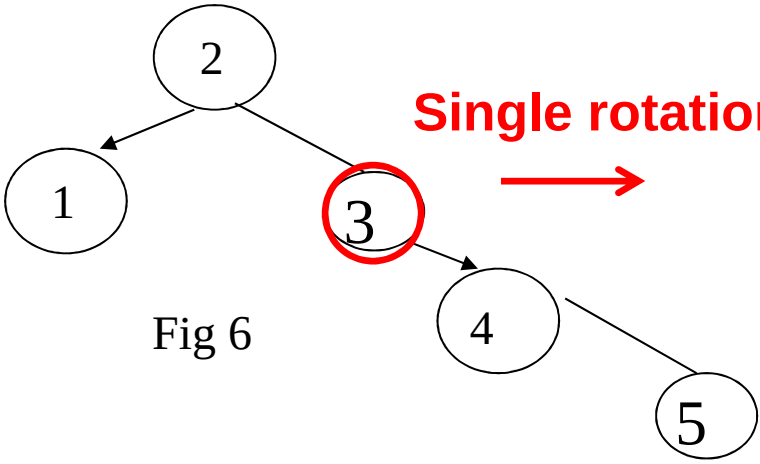


Fig 6

Single rotation

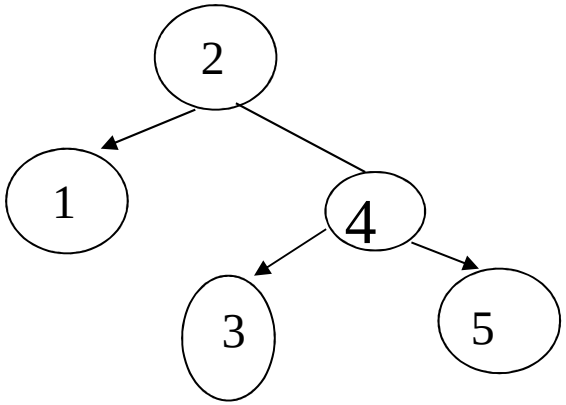
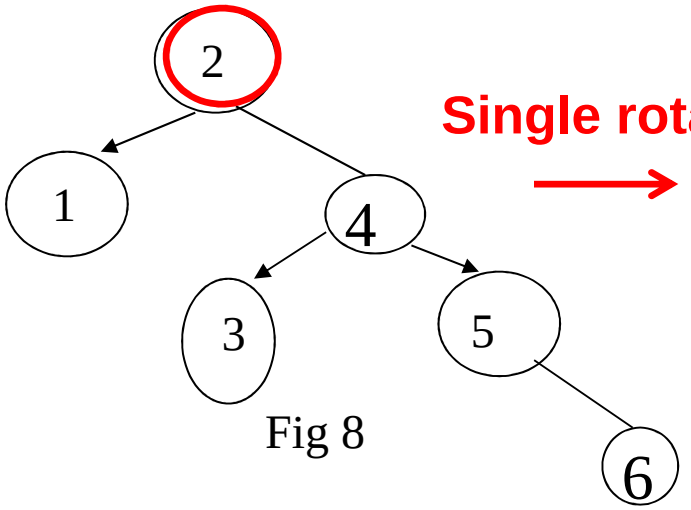


Fig 7



Single rotation

Fig 8

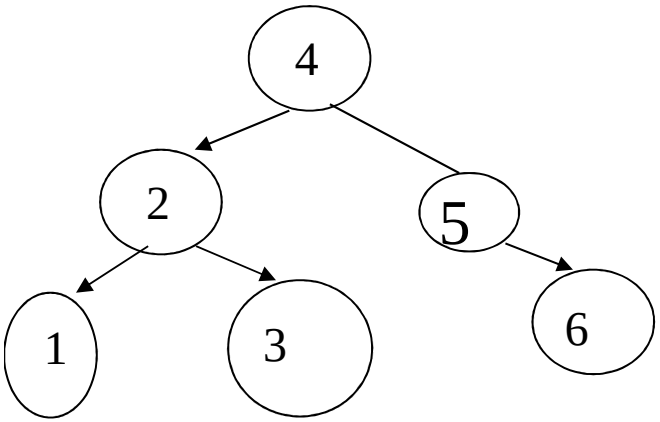
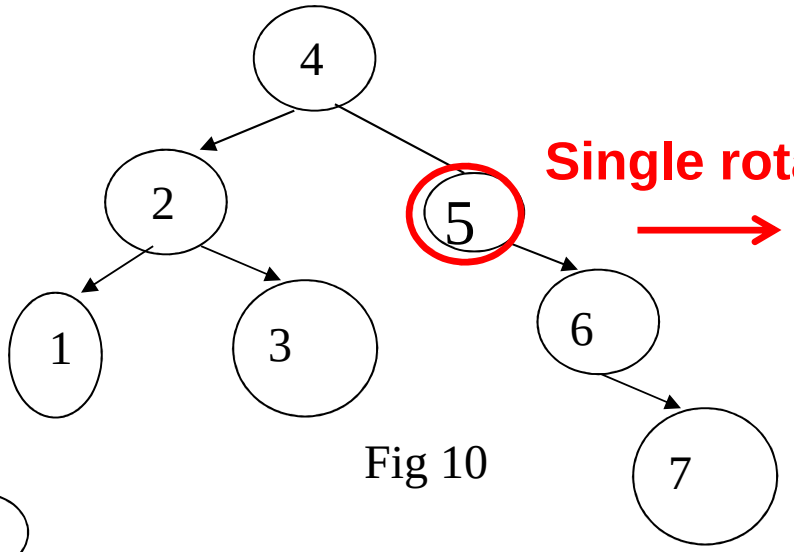


Fig 9



Single rotation

Fig 10

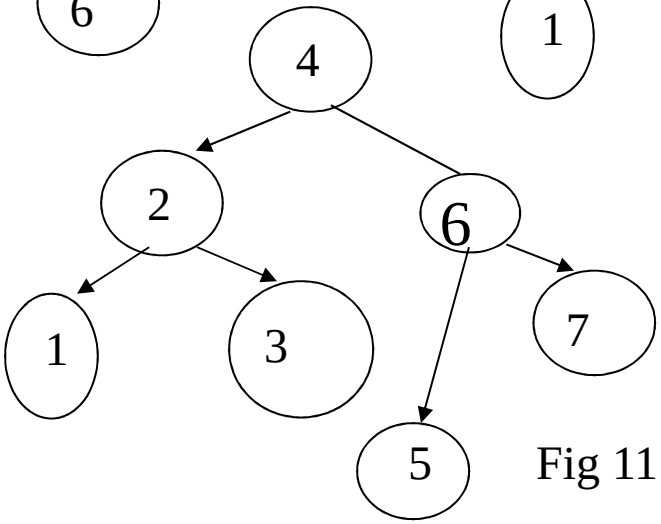


Fig 11

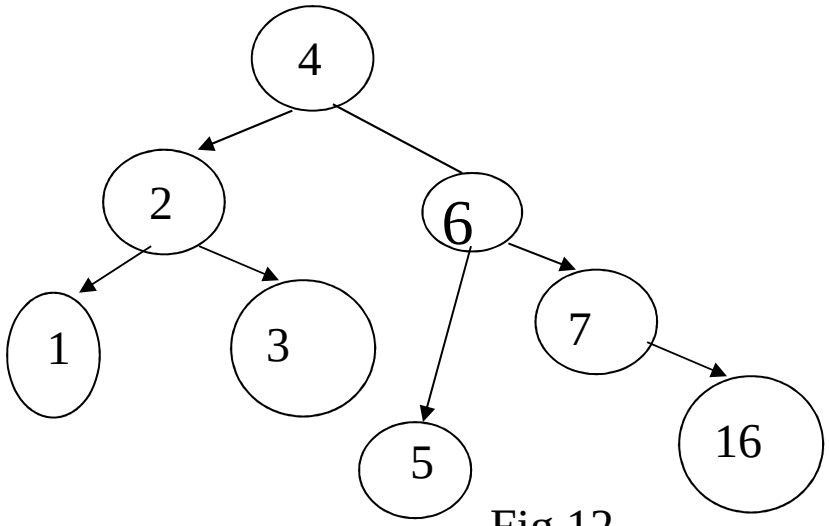


Fig 12

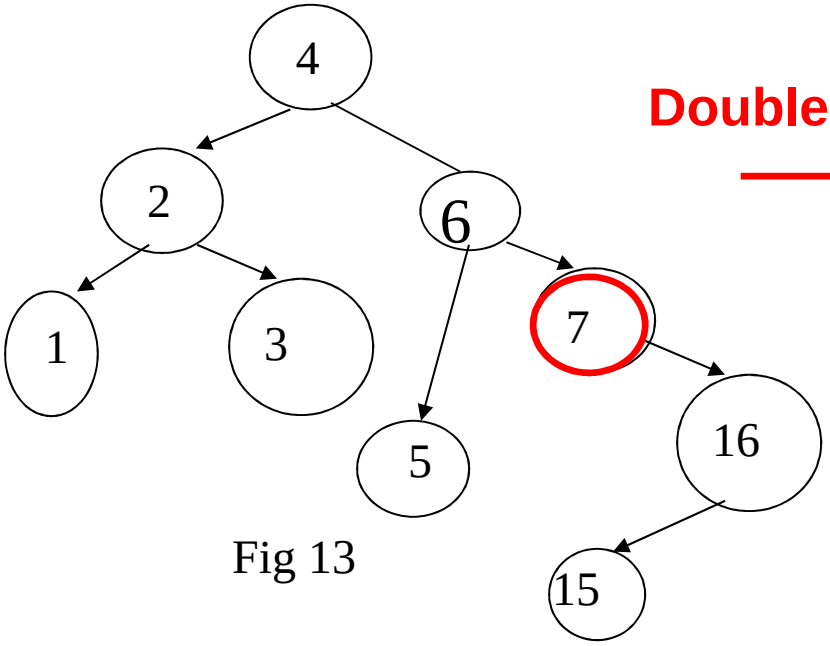


Fig 13

Double rotation

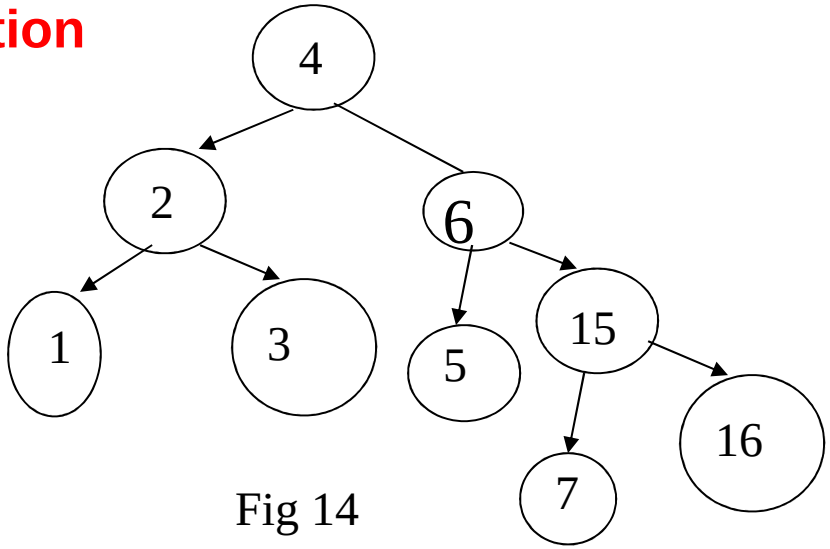


Fig 14

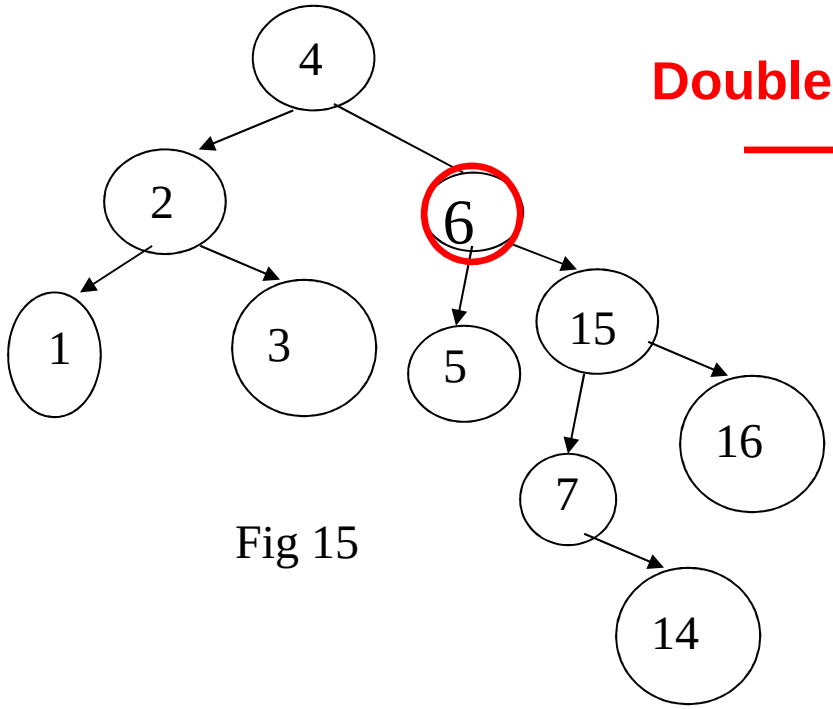


Fig 15

Double rotation

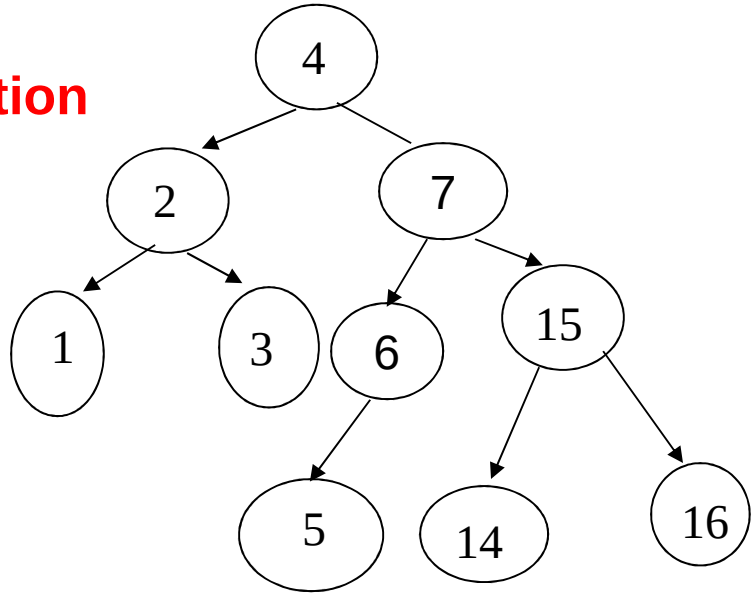


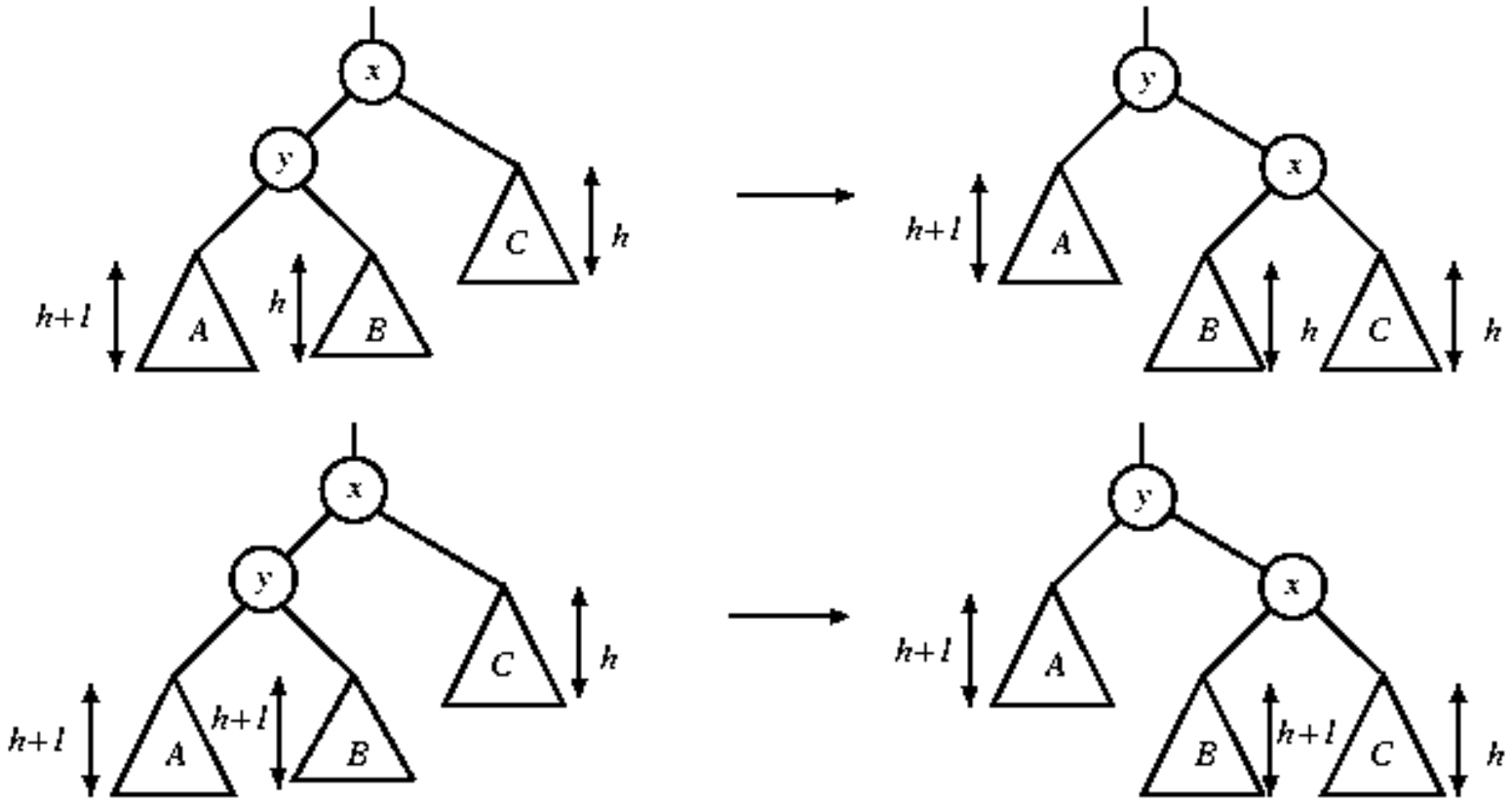
Fig 16

Deletion

- Xóa nút x như trong CNPTK.
- Kiểm tra các nút trên path từ nút thay thế đến nút gốc. Với mỗi nút x bắt gặp trên path, kiểm tra left(x) và right(x). Nếu thõa mãn ĐK cân bằng thì xét tiếp nút ở trên, nếu không thõa mãn thì thực hiện phép quay tương ứng.
- Với phép xóa nút thì sau khi thực hiện phép quay ở nút x, chúng ta vẫn có thể phải thực hiện phép quay ở các nút trên của x. Vì vậy sau khi quay chúng ta vẫn phải tiếp tục xét các nút ở phía trên

Single rotations in deletion

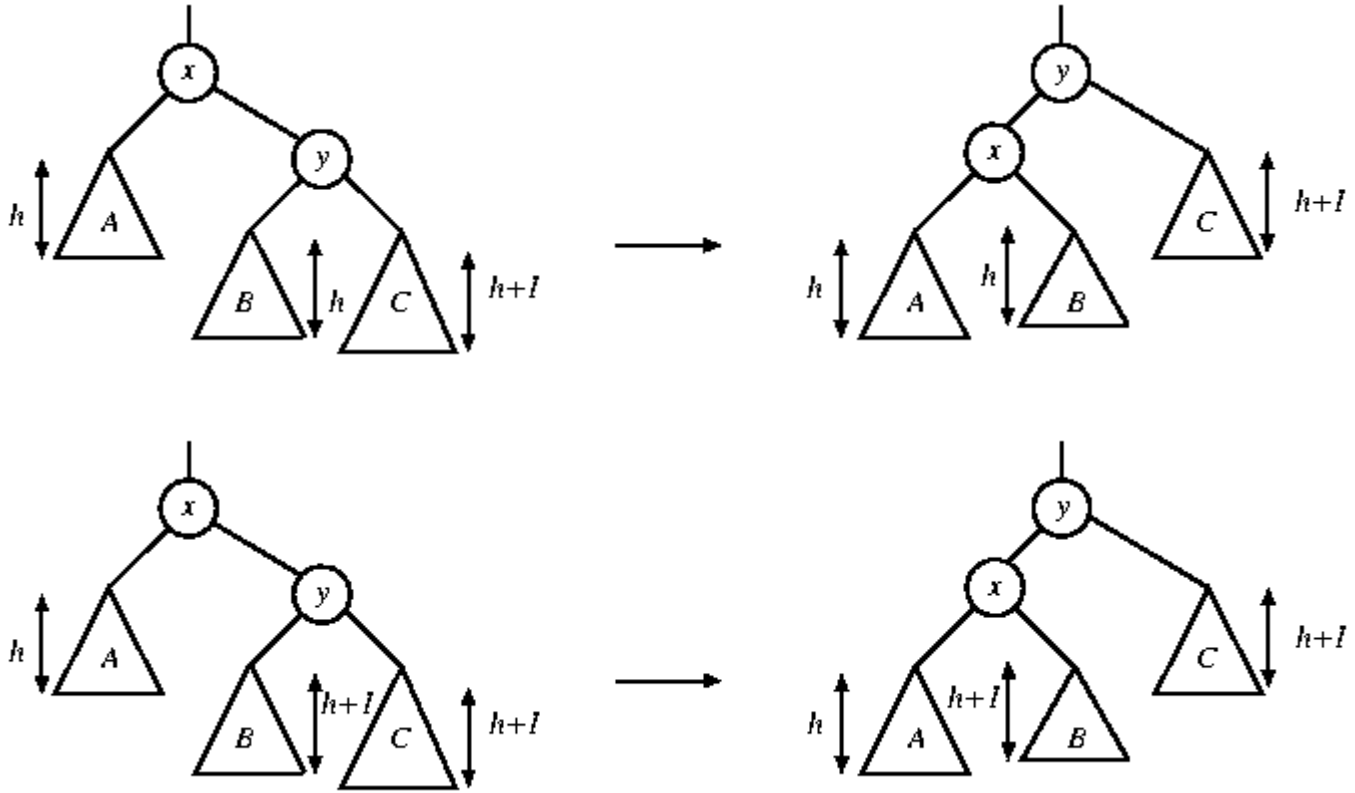
Xóa nút trên cây con C.



rotate with left child

Single rotations in deletion

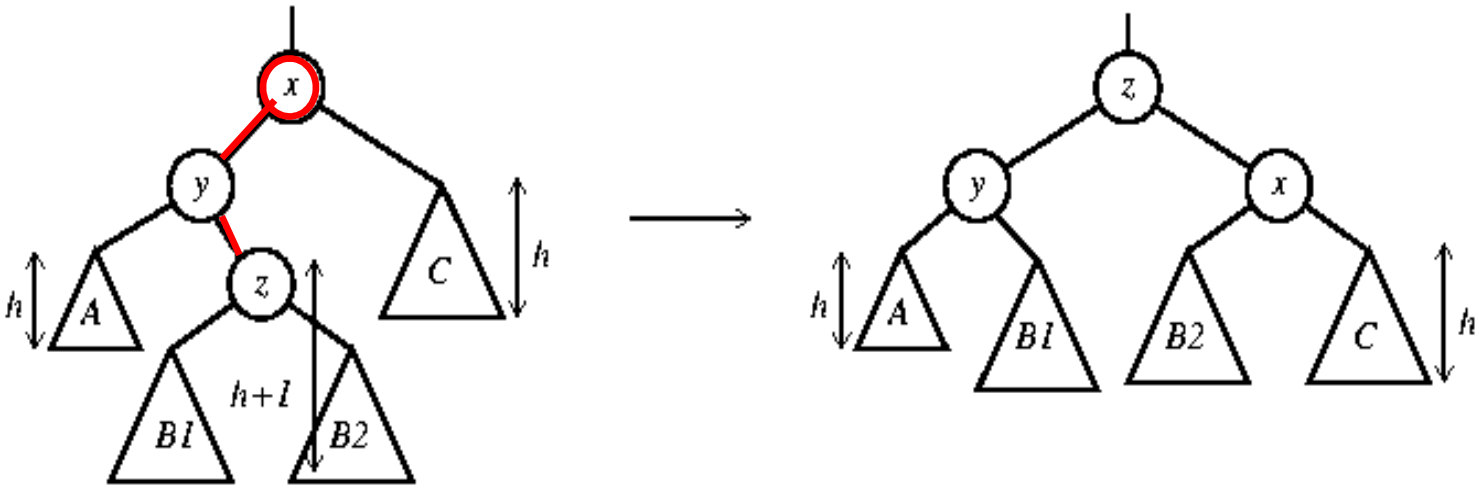
Xóa nút trên cây con A.



rotate with right child

Double rotation

Xóa nút trên cây con C.

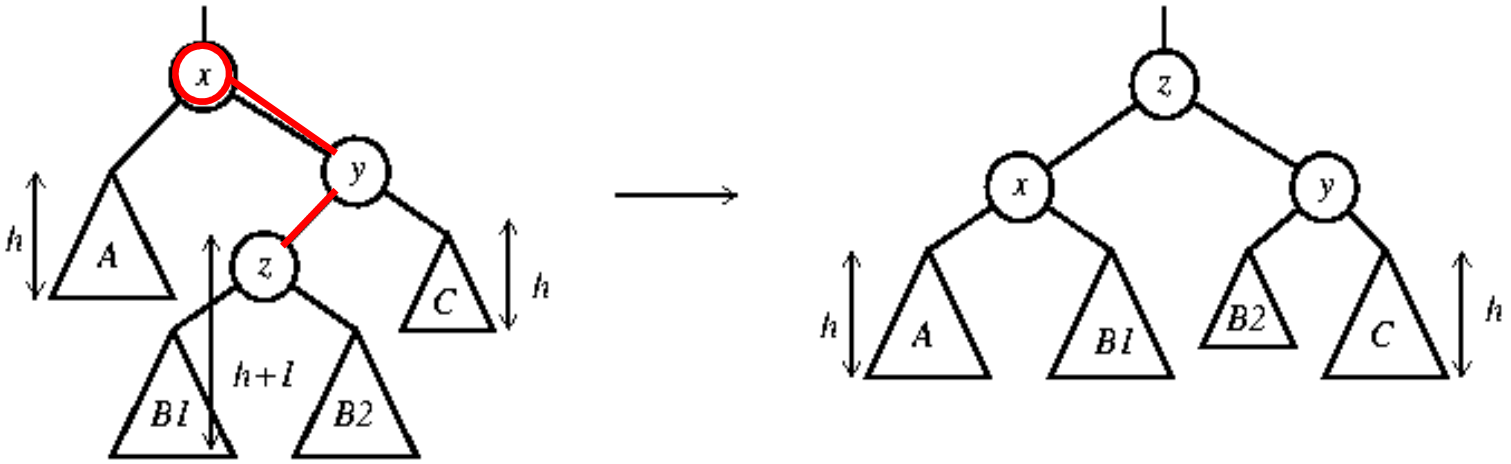


Double rotate with left child

also called left-right rotate

Double rotation

Xóa nút trên cây con A.



Double rotate with right child

also called right-left rotate

Ví dụ xóa nút X