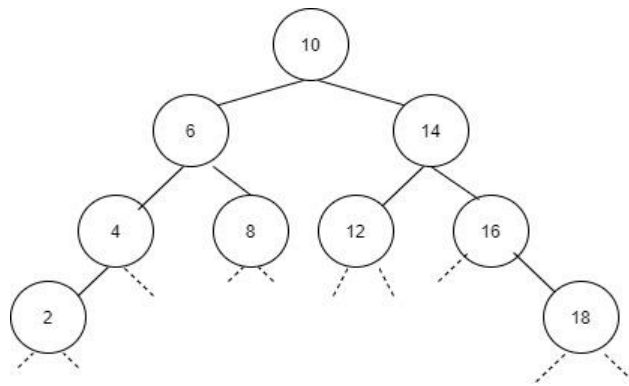


GIỚI THIỆU CÁC PHÉP XOAY

I. VẤN ĐỀ CÂN BẰNG

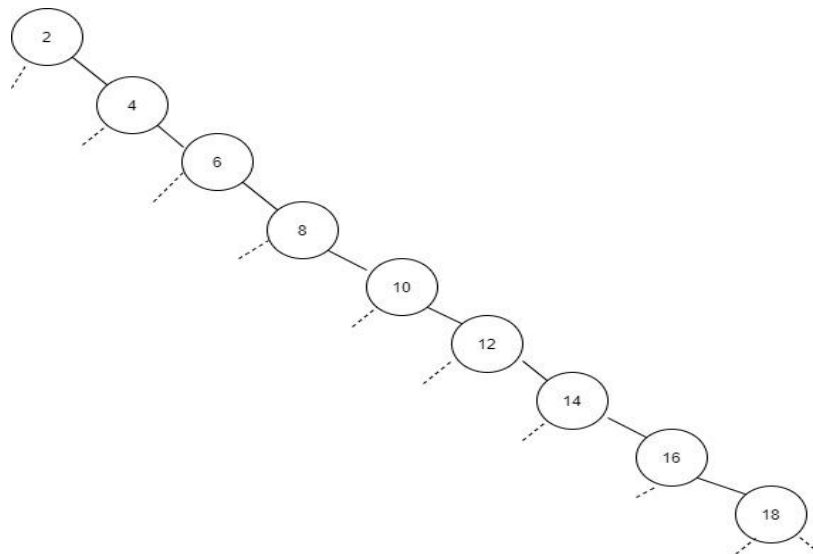
Cho dãy khóa 2 4 6 8 10 12 14 16 18

Một cây BST biểu diễn dãy khóa trên là:



Hình 5.1: Cây BST cân bằng với thời gian tìm kiếm một khóa $O(\log n)$.

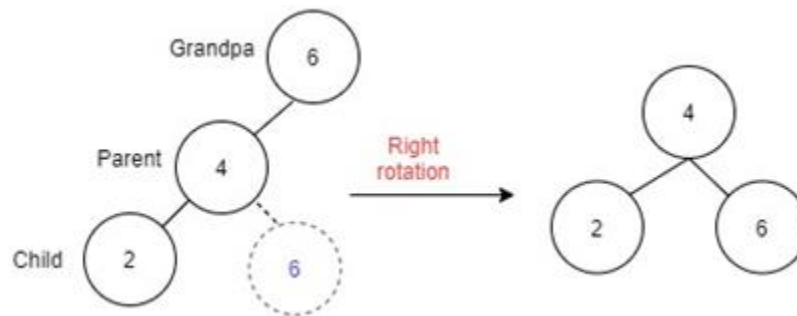
Nếu các khóa được chèn vào cây BST theo thứ tự xuất hiện của chúng:



Hình 5.2: Cây BST với thời gian tìm kiếm một khóa $O(n)$.

“Cân bằng” là sự khác biệt giữa cây BST ở hình 5.1 và hình 5.2. Độ phức tạp của việc tìm kiếm trên cây BST cân bằng được đảm bảo $O(\log n)$.

Việc cân bằng trên cây BST có thể được thực hiện mỗi khi thêm một khóa mới vào cây. Chẳng hạn cho dãy khóa: 6 4 2; khi lần lượt thêm vào cây BST ta được một cây tuyến tính; để cân bằng một phép xoay phải (right rotation) được thực hiện tại nút cha (parent) để xoay nút ông (grandpa) về là con phải của nút cha (hình 5.3), kết quả là một cây BST cân bằng (hình 5.3).

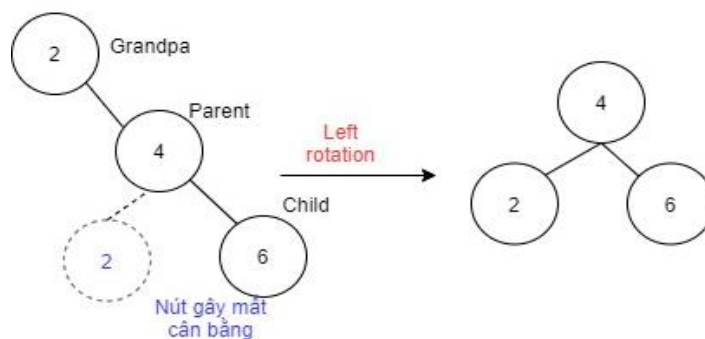


Hình 5.3: Xoay phải (right rotation) quanh nút cha (parent).

II. CÁC PHÉP XOAY

II.1 Xoay trái (Left rotation)

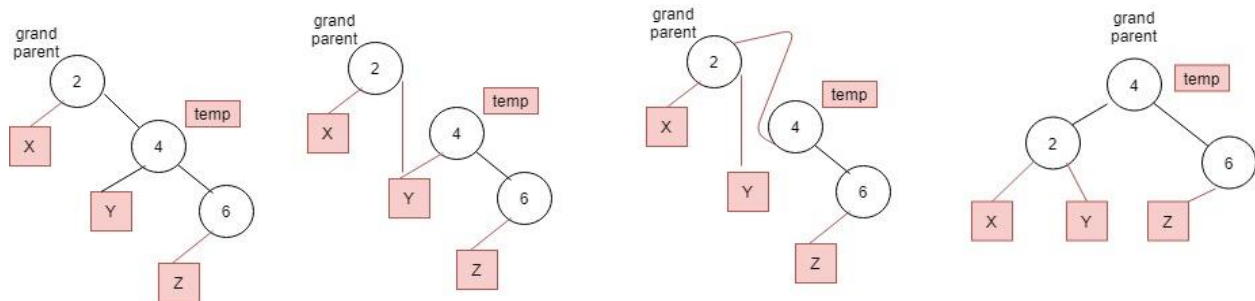
Cho dãy khóa 2 4 6; sau khi thêm lần lượt dãy khóa trên vào cây BST ta được cây BST mất cân bằng với việc mất cân bằng xảy ra sau khi thêm nút 6. Giải pháp để cân bằng là xoay trái nút cha (parent) để được cây cân bằng như hình 5.4.



Hình 5.4: Thực hiện xoay trái (left rotation) sau khi thêm khóa 6.

Quy luật: Từ nút ông (**grandpa**) nếu nút mất cân bằng ở con phải, cây cháu phải (**right child right subtree**) → thực hiện xoay trái (**left rotation**) nút ông (**grandpa**) quanh nút cha (**parent**).

Giải thuật:

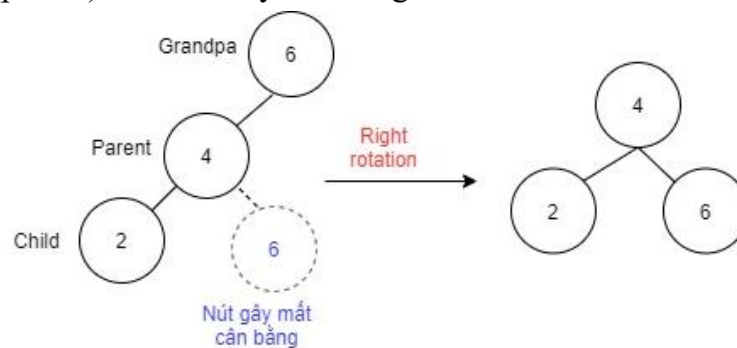


Hình 5.5: Các bước thực hiện xoay trái.

- temp = grandparent right child
- grandparent right child = temp left child
- temp left child = grandparent

II.2 Xoay phải (Right rotation)

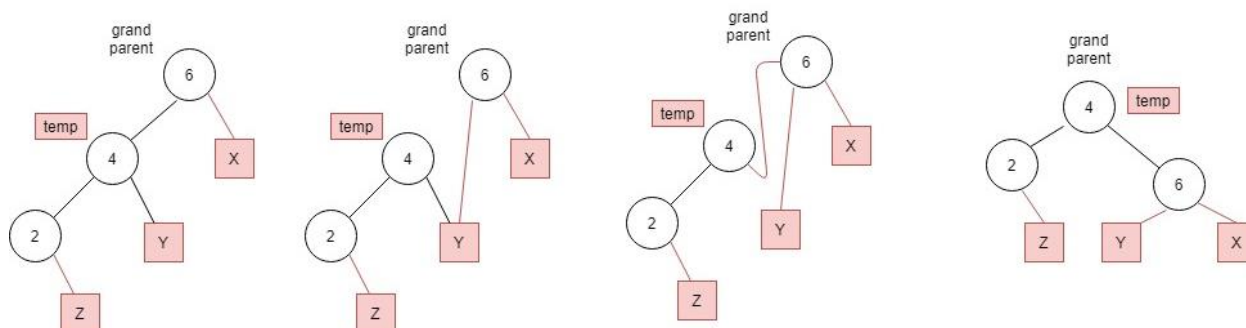
Cho dãy khóa 6 4 2; sau khi thêm lần lượt dãy khóa trên vào cây BST ta được cây BST mất cân bằng với việc mất cân bằng xảy ra sau khi thêm nút 2. Giải pháp để cân bằng là xoay phải nút cha (parent) để được cây cân bằng như hình 5.6.



Hình 5.6: Thực hiện xoay phải (right rotation) sau khi thêm khóa 2.

Quy luật: Từ nút ông (**grandpa**) nếu nút mất cân bằng ở con trái, cây cháu trái (**left child left subtree**) → thực hiện xoay phải (**right rotation**) nút ông (**grandpa**) quanh nút cha (**parent**).

Giải thuật

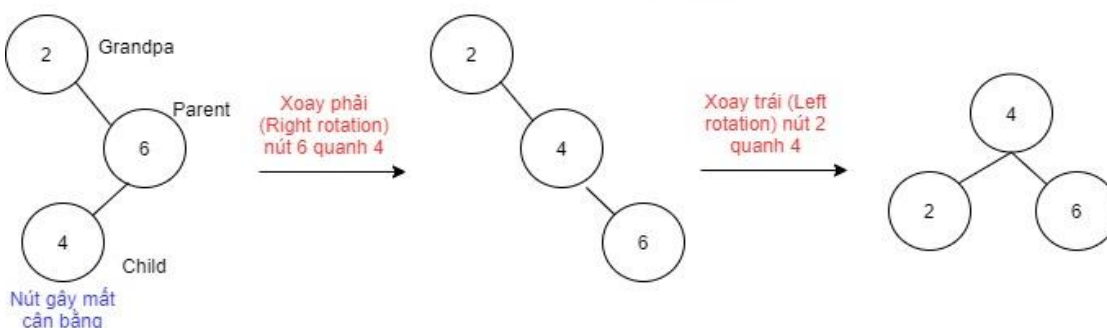


Hình 5.7: Các bước thực hiện xoay phải.

- temp = grandparent left child
- grandparent left child = temp right child
- temp right child = grandparent

II.3 Xoay phải trái (Right Left rotation)

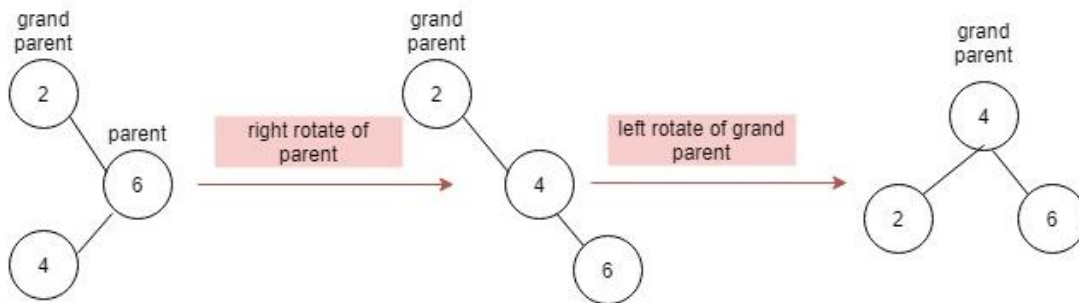
Cho dãy khóa 2 6 4; sau khi thêm lần lượt dãy khóa trên vào cây BST ta được cây BST mất cân bằng với việc mất cân bằng xảy ra sau khi thêm nút 4. Giải pháp để cân bằng là thực hiện 2 phép xoay: xoay phải, rồi xoay trái để được cây cân bằng như hình 5.8.



Hình 5.8: Thực hiện xoay phải trái (right left rotation) sau khi thêm khóa 4.

Quy luật: Từ nút ông (**grandpa**) nếu nút mất cân bằng ở con phải, cây cháu trái (**right child left subtree**) → thực hiện tổ hợp phép xoay phải trái (**right left rotation**).

Giải thuật

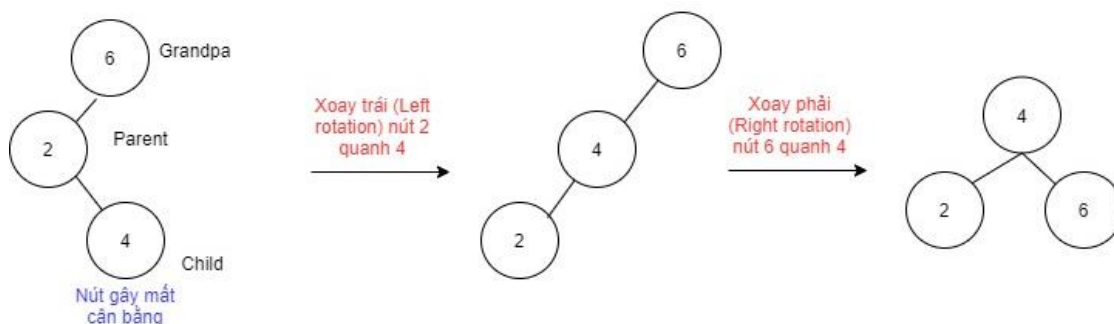


Hình 5.9: Các bước thực hiện xoay phải trái.

- grandparent right child = **rightRotate**(grandparent right child)
- grandparent = **leftRotate**(grandparent)

II.4 Xoay trái phải (Left Right rotation)

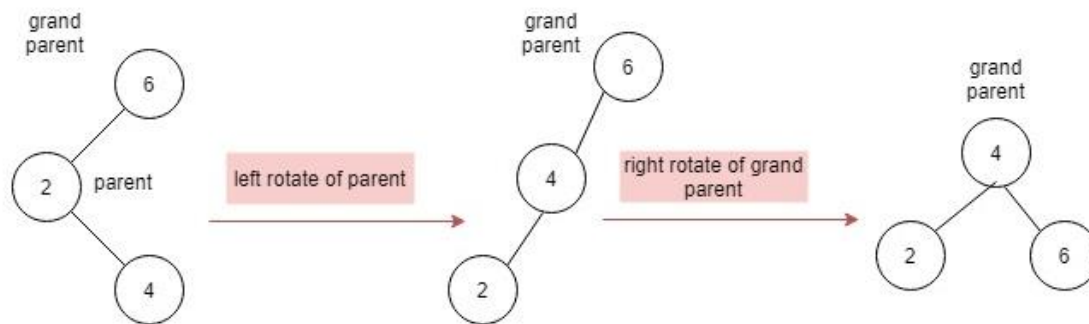
Cho dãy khóa 6 2 4; sau khi thêm lần lượt dãy khóa trên vào cây BST ta được cây BST mất cân bằng với việc mất cân bằng xảy ra sau khi thêm nút 4. Giải pháp để cân bằng là thực hiện 2 phép xoay: xoay trái, rồi xoay phải để được cây cân bằng như hình 5.10.



Hình 5.10: Thực hiện xoay trái phải (left right rotation) sau khi thêm khóa 4.

Quy luật: Từ nút ông (**grandpa**) nếu nút mất cân bằng ở con trái, cây cháu phải (**left child right subtree**) → thực hiện tổ hợp phép xoay trái phải (**left right rotation**).

Giải thuật



Hình 5.11: Các bước thực hiện xoay trái phải.

- grandparent left child = **leftRotate**(grandparent left child)
- grandparent = **rightRotate**(grandparent)