TH 03 - CÂY CÂN BẰNG

01/2018

1 Thực hành

AVL là tên viết tắt của các tác giả người Nga đã đưa ra định nghĩa của cây cân bằng Adelson-Velskii và Landis (1962). Vì lý do này, người ta gọi cây nhị phân cân bằng là cây AVL.

Cây AVL là cây nhị phân tìm kiến có độ cân bằng cao. Cây AVL kiểm tra độ cao của các cây con bên trái và cây con bên phải và bảo đảm rằng hiệu số giữa chúng là không lớn hơn 1. Hiệu số này được gọi là *Balance Factor (Nhân tố cân bằng)*.

Nếu hiệu số giữa độ cao của các cây con bên trái và cây con bên phải là lớn hơn 1 thì cây sẽ được cân bằng lại.

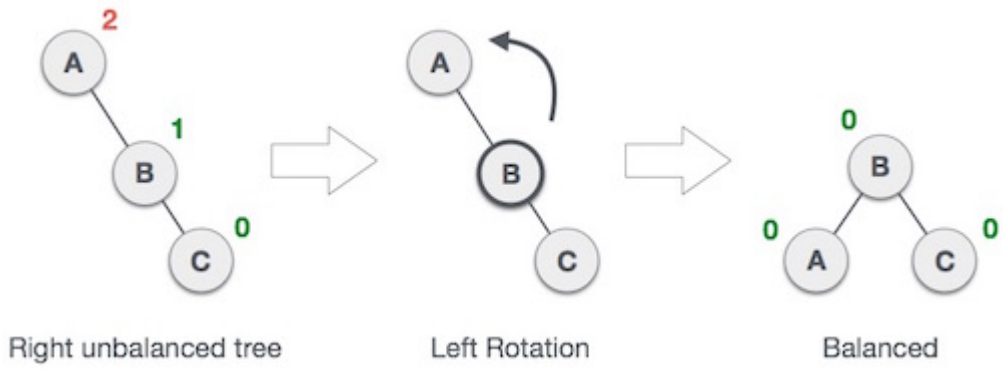
1.1 Kiểm tra cân bằng

Nhân tố cân bằng của một node trong cây cân bằng được xác định bởi hiệu số giữa chiều cao của cây con trái và cây con phải tại node đó.

```
private int checkBalance(Node x) {
    return height(x.left) - height(x.right);
}
```

1.2 Kỹ thuật quay trái

Khi một node được chèn vào cây con bên phải của nhánh bên phải và gây ra sự mất cân bằng của cây thì ta có thể thực hiện kỹ thuật quay trái đơn tại node bất cân bằng như sau.

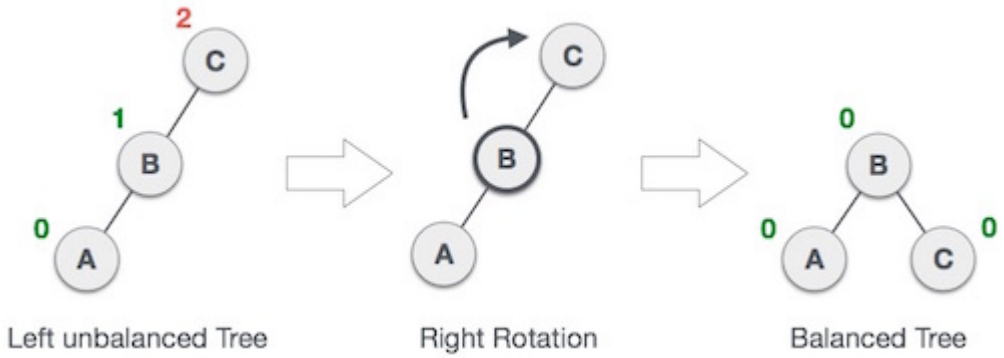


Hình 1: Kỹ thuật quay trái

```
private Node rotateLeft(Node x) {
    Node y = x.right;
    x.right = y.left;
    y.left = x;
    y.size = x.size;
    x.size = 1 + size(x.left) + size(x.right);
    x.height = 1 + Math.max(height(x.left), height(x.right));
    y.height = 1 + Math.max(height(y.left), height(y.right));
    return y;
}
```

1.3 Kỹ thuật quay phải

Khi một node được chèn vào cây con bên trái của nhánh bên trái và gây ra sự mất cân bằng của cây thì ta có thể thực hiện kỹ thuật quay phải đơn tại node bất cân bằng như sau.

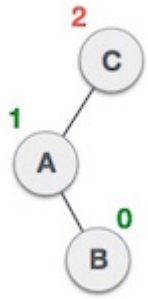


Hình 2: Kỹ thuật quay phải

```
private Node rotateRight(Node x) {
    Node y = x.left;
    x.left = y.right;
    y.right = x;
    y.size = x.size;
    x.size = 1 + size(x.left) + size(x.right);
    x.height = 1 + Math.max(height(x.left), height(x.right));
    y.height = 1 + Math.max(height(y.left), height(y.right));
    return y;
}
```

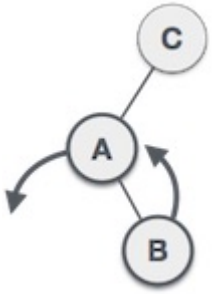
1.4 Kỹ thuật quay trái-phải

Khi một node được chèn vào trong cây con bên phải của nhánh bên trái và gây mất cân bằng tại node C. Với tình huống này, cây AVL có thể được cân bằng lại bằng cách thực hiện kỹ thuật quay trái-phải.



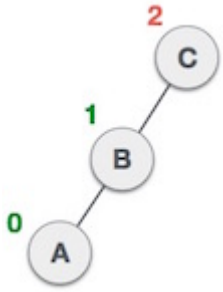
Hình 3: Mất cân bằng trái-phải

Đầu tiên, thực hiện phép quay trái trên node A (cây con bên trái của C). Điều này làm cho A trở thành cây con bên trái của B. Và B thay thế A để trở thành cây con bên trái của C.



Hình 4: Mất cân bằng trái-phải

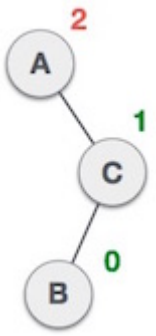
Bây giờ node C tuy vẫn mất cân bằng nhưng đã trở thành trường hợp 1.2 và có thể áp dụng kỹ thuật quay phải để cân bằng cây.



Hình 5: Mất cân bằng trái-phải

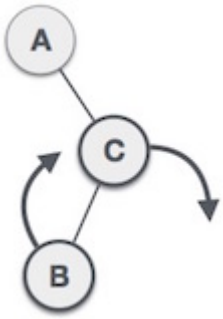
1.5 Kỹ thuật quay phải-trái

Khi một nút được chèn vào trong cây con bên trái của nhánh bên phải và gây mất cân bằng tại node A. Với tình huống này, cây AVL có thể được cân bằng lại bằng cách thực hiện kỹ thuật quay phải-trái.



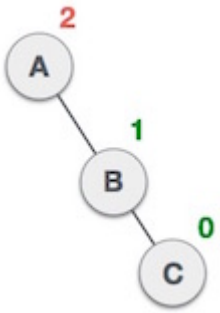
Hình 6: Mất cân bằng phải-trái

Đầu tiên, thực hiện phép quay phải trên phải node C. Điều này làm cho C trở thành cây con bên phải của B. Và B thay thế C để trở thành cây con bên phải của A.



Hình 7: Mất cân bằng phải-trái

Bây giờ node A tuy vẫn mất cân bằng nhưng đã trở thành trường hợp 1.3 và có thể áp dụng kỹ thuật quay phải để cân bằng cây.



Hình 8: Mất cân bằng phải-trái

1.6 Mã nguồn kỹ thuật cân bằng node trong cây

```
private Node balance(Node x) {
    if (checkBalance(x) < -1) {
        if (checkBalance(x.right) > 0) {
            x.right = rotateRight(x.right);
        }
        x = rotateLeft(x);
    }
    else if (checkBalance(x) > 1) {
        if (checkBalance(x.left) < 0) {
            x.left = rotateLeft(x.left);
        }
        x = rotateRight(x);
    }
    return x;
}
```