

   **Cây tìm kiếm nhị phân**

## Contents

- Cây, cây nhị phân
- Cây nhị phân tìm kiếm
  - Định nghĩa
  - Thao tác tìm kiếm
  - Thao tác thêm

Nhắc lại định nghĩa về cây:

- Cây là một đồ thị có hướng.
- Nếu có cạnh từ  $u$  đến  $v$  thì ta nói  $u$  là cha của  $v$ , và  $v$  là con của  $u$ .
- Cây có một đỉnh duy nhất không có cha, gọi là gốc của cây.
- Mỗi đỉnh khác gốc có duy nhất một cha
- Mỗi đỉnh có thể có hoặc không có con.
- Các đỉnh không có con gọi là nút lá.

Đối với cây nhị phân, số con của mỗi đỉnh không quá 2. Để cho tiện, ta gọi 2 con đó lần lượt là con bên trái và con bên phải. Cây con với gốc là đỉnh con bên trái gọi là cây con bên trái, tương tự với cây con bên phải.

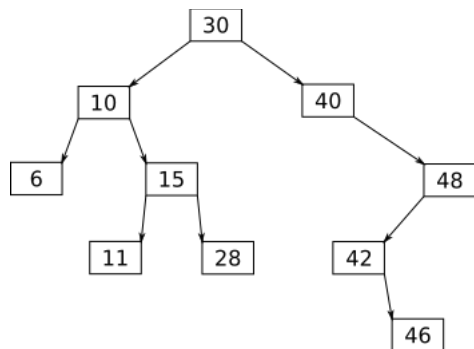
## Định nghĩa

Với cây nhị phân tìm kiếm, mỗi đỉnh có thêm một thuộc tính là khóa. Khóa của các đỉnh thỏa tính chất sau:

- Khóa của các đỉnh không trùng lặp, mỗi đỉnh có một khóa khác với tất cả các đỉnh còn lại.
- Với mỗi đỉnh, khóa của tất cả các đỉnh trong cây con bên trái nhỏ hơn khóa của đỉnh đó.
- Với mỗi đỉnh, khóa của tất cả các đỉnh trong cây con bên phải lớn hơn khóa của đỉnh đó.

Trong thực tế khóa của đỉnh thường là một số nguyên. Tuy nhiên nó có thể là bất cứ kiểu dữ liệu nào mà có thể so sánh được.

Ví dụ một cây nhị phân tìm kiếm:



Cho sẵn một cây nhị phân, thao tác này sẽ duyệt cây để tìm một nút có khóa là K. Cài đặt mã giả như sau:

```
function FindKey(root, K)
    if root == NIL
        return NIL
    if K > root.key
        return FindKey(root.rightChild, K)
    if K < root.key
        return FindKey(root.leftChild, K)
    return root
```

Giải thích:

- Nếu nút đang xét là nút rỗng thì trả về nút rỗng, ý là không tìm được nút nào có khóa K
- Nếu khóa của nút đang xét bằng K thì trả về nút đang xét, quá trình tìm kiếm thành công
- Nếu K bé hơn khóa của nút đang xét, ta thấy nếu tồn tại một nút có khóa bằng K thì nó phải nằm ở cây con bên trái, vì vậy ta gọi đệ quy để tìm ở nút con trái
- Tương tự với trường hợp K lớn hơn khóa của nút đang xét

Ta thấy, có thể dễ dàng khử đệ quy quá trình trên:

```
function FindKey(root, K)
    while root != NIL
        if K < root.key
            root = root.leftChild
        else if K > root.key
            root = root.rightChild
        else
            break
    return root
```

## Thao tác thêm

Hàm sau thêm một nút có khóa là K:

```
function InsertKey(root, K)
    if root == NIL
        return NewNode(K)
    if K > root.key
        root.rightChild = InsertKey(root.rightChild, K)
    if K < root.key
        root.leftChild = InsertKey(root.leftChild, K)
    return root

// Một lời gọi mẫu
root = InsertKey(root, 0)
```

Giải thích cũng tương tự như ở thao tác tìm kiếm.

**Còn tiếp...**

Loading...