

Cây tìm kiếm nhị phân (Binary Search Tree)

⌂ Trang trước (../cau-truc-du-lieu-va-giai-thuat/duyet-cay.jsp)

Trang sau ➔ (../cau-truc-du-lieu-va-giai-thuat/giai-thuat-cay-avl.jsp)

Cây tìm kiếm nhị phân là gì ?

Một cây tìm kiếm nhị phân (Binary Search Tree – viết tắt là BST) là một cây mà trong đó tất cả các nút đều có các đặc điểm sau:

Cây con bên trái của một nút có khóa (key) nhỏ hơn hoặc bằng giá trị khóa của nút cha (của cây con này).

Cây con bên phải của một nút có khóa lớn hơn hoặc bằng giá trị khóa của nút cha (của cây con này).

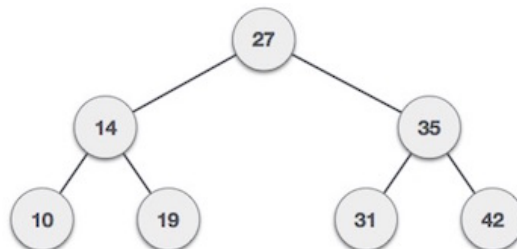
Vì thế có thể nói rằng, một cây tìm kiếm nhị phân (BST) phân chia tất cả các cây con của nó thành hai phần: *cây con bên trái* và *cây con bên phải* và có thể được định nghĩa như sau:

$$\text{left_subtree (keys)} \leq \text{node (key)} \leq \text{right_subtree (keys)}$$

Biểu diễn cây tìm kiếm nhị phân (BST)

Cây tìm kiếm nhị phân (BST) là một tập hợp bao gồm các nút được sắp xếp theo cách để chúng có thể duy trì hoặc tuân theo các đặc điểm của cây tìm kiếm nhị phân. Mỗi một nút thì đều có một khóa và giá trị liên kết với nó. Trong khi tìm kiếm, khóa cần tìm được so sánh với các khóa trong cây tìm kiếm nhị phân (BST) và nếu tìm thấy, giá trị liên kết sẽ được thu nhận.

Ví dụ một cây tìm kiếm nhị phân (BST):



Từ hình ví dụ minh họa trên ta thấy rằng, khóa của nút gốc có giá trị 27 và tất cả khóa bên trái của cây con bên trái đều có giá trị nhỏ hơn 27 này và tất cả các khóa bên phải của cây con bên phải đều có giá trị lớn hơn 27.



Hoạt động cơ bản trên cây tìm kiếm nhị phân

Dưới đây là một số hoạt động cơ bản có thể được thực hiện trên cây tìm kiếm nhị phân:

Hoạt động tìm kiếm: tìm kiếm một phần tử trong một cây.

Hoạt động chèn: chèn một phần tử vào trong một cây.

Hoạt động duyệt tiền thứ tự: duyệt một cây theo cách thức duyệt tiền thứ tự.

Hoạt động duyệt trung thứ tự: duyệt một cây theo cách thức duyệt trung thứ tự.

Hoạt động duyệt hậu thứ tự: duyệt một cây theo cách thức duyệt hậu thứ tự.

Nút (Node) trong cây tìm kiếm nhị phân

Một nút có một vài dữ liệu, tham chiếu tới các nút con bên trái và nút con bên phải của nó.

```
struct node {
    int data;
    struct node *leftChild;
    struct node *rightChild;
};
```

Hoạt động tìm kiếm trong cây tìm kiếm nhị phân

Mỗi khi một phần tử được tìm kiếm: bắt đầu tìm kiếm từ nút gốc, sau đó nếu dữ liệu là nhỏ hơn giá trị khóa (key), thì sẽ tìm phần tử ở cây con bên trái; nếu lớn hơn thì sẽ tìm phần tử ở cây con bên phải. Dưới đây là giải thuật cho mỗi nút:

```
struct node* search(int data){
    struct node *current = root;
    printf("Truy cap cac phan tu: ");

    while(current->data != data){

        if(current != NULL) {
            printf("%d ", current->data);

            //tới cây con bên trái
            if(current->data > data){
                current = current->leftChild;
            } //else tới cây con bên phải
            else {
                current = current->rightChild;
            }

            //không tìm thấy
            if(current == NULL){
                return NULL;
            }
        }
    }
    return current;
}
```

Hoạt động chèn trong cây tìm kiếm nhị phân

Mỗi khi một phần tử được chèn: đầu tiên chúng ta cần xác định vị trí chính xác của phần tử này. Bắt đầu tìm kiếm từ nút gốc, sau đó nếu dữ liệu là nhỏ hơn giá trị khóa (key), thì tìm kiếm vị trí còn trống ở cây con bên trái và chèn dữ liệu vào đó; nếu dữ liệu là nhỏ hơn thì tìm kiếm vị trí còn trống ở cây con bên phải và chèn dữ liệu vào đó.

```

void insert(int data){
    struct node *tempNode = (struct node*) malloc(sizeof(struct node));
    struct node *current;
    struct node *parent;

    tempNode->data = data;
    tempNode->leftChild = NULL;
    tempNode->rightChild = NULL;

    //Nếu cây là trống
    if(root == NULL){
        root = tempNode;
    }else {
        current = root;
        parent = NULL;

        while(1){
            parent = current;

            //tới cây con bên trái
            if(data < parent->data){
                current = current->leftChild;
                //chèn dữ liệu vào cây con bên trái

                if(current == NULL){
                    parent->leftChild = tempNode;
                    return;
                }
            }//tới cây con bên phải
            else{
                current = current->rightChild;
                //chèn dữ liệu vào cây con bên phải
                if(current == NULL){
                    parent->rightChild = tempNode;
                    return;
                }
            }
        }
    }
}

```

Loạt bài hướng dẫn **Cấu trúc dữ liệu và giải thuật** của chúng tôi dựa trên nguồn tài liệu của trang: Tutorialspoint

Follow fanpage của team <https://www.facebook.com/vietjackteam/> (<https://www.facebook.com/vietjackteam/>) hoặc facebook cá nhân Nguyễn Thanh Tuyền <https://www.facebook.com/tuyen.vietjack> (<https://www.facebook.com/tuyen.vietjack>) để tiếp tục theo dõi các loạt bài mới nhất về Java,C,C++,Javascript,HTML,Python,Database,Mobile.... mới nhất của chúng tôi.

Bài học Cấu trúc dữ liệu và giải thuật phổ biến tại vietjack.com:

Giải thuật tiệm cận - Asymptotic Algorithms (<http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/phan-tich-tiem-can.jsp>)

Cấu trúc dữ liệu mảng (Array) (<http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/cau-truc-du-lieu-mang.jsp>)

Danh sách liên kết - Linked List (<http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/cau-truc-du-lieu-danh-sach-lien-ket.jsp>)

Cấu trúc dữ liệu ngăn xếp - Stack (<http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/cau-truc-du-lieu-ngan-xep.jsp>)

Cấu trúc dữ liệu hàng đợi - Queue (<http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/cau-truc-du-lieu-hang-doi.jsp>)

Tìm kiếm tuyến tính - Linear Search (<http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/giai-thuat-tim-kiem-tuyen-tinh.jsp>)

Tìm kiếm nhị phân - Binary Search (<http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/giai-thuat-tim-kiem-nhi-phan.jsp>)

Sắp xếp nổi bọt - Bubble Sort (<http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/giai-thuat-sap-xep-noi-bot.jsp>)

Sắp xếp chèn - Insertion Sort (<http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/giai-thuat-sap-xep-chen.jsp>)

⏪ Trang trước ([../cau-truc-du-lieu-va-giai-thuat/duyet-cay.jsp](http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/duyet-cay.jsp))

Trang sau ⏩ ([../cau-truc-du-lieu-va-giai-thuat/giai-thuat-cay-avl.jsp](http://vietjack.com/cau-truc-du-lieu-va-giai-thuat/giai-thuat-cay-avl.jsp))

0 bình luận

Sắp xếp theo

Cũ nhất



Thêm bình luận...

Plugin bình luận của Facebook

Bài viết liên quan

Các loạt bài khác:

160 bài học ngữ pháp tiếng Anh hay nhất (<http://vietjack.com/ngu-phap-tieng-anh/index.jsp>)

155 bài học Java tiếng Việt hay nhất (<http://vietjack.com/java/index.jsp>)

100 bài học Android tiếng Việt hay nhất (<http://vietjack.com/android/index.jsp>)

247 bài học CSS tiếng Việt hay nhất (<http://vietjack.com/css/index.jsp>)

197 thẻ HTML cơ bản (<http://vietjack.com/html/index.jsp>)

297 bài học PHP (<http://vietjack.com/php/index.jsp>)

85 bài học C# hay nhất (<http://vietjack.com/csharp/index.jsp>)

101 bài học C++ hay nhất (<http://vietjack.com/cplusplus/index.jsp>)

97 bài tập C++ có giải hay nhất (http://vietjack.com/bai_tap_cplusplus_co_giai/index.jsp)

208 bài học Javascript có giải hay nhất (<http://vietjack.com/javascript/index.jsp>)

Và còn rất nhiều loạt bài học khác....



Học tiếng Anh tại vietjack.com:

160 bài học ngữ pháp tiếng Anh hay nhất (<http://vietjack.com/ngu-phap-tieng-anh/index.jsp>)

160 bài tập ngữ pháp tiếng Anh hay nhất (<http://vietjack.com/bai-tap-ngu-phap-tieng-anh/index.jsp>)

72 bài ngữ pháp thực hành (<http://vietjack.com/ngu-phap-tieng-anh-co-ban/index.jsp>)

50 tình huống tiếng Anh thông dụng (<http://vietjack.com/tinh-huong-tieng-anh-thong-dung/index.jsp>)

120 bí kíp luyện phần V TOEIC (<http://vietjack.com/part-5-toeic/index.jsp>)



VietJack.com - Chia sẻ...
95.744 lượt thích

Đã thích Đăng kí ngay Chia sẻ

Bạn và 12 người bạn khác thích nội dung này




Boosts your productivity
with PhpStorm —
the most powerful PHP IDE

PhpStorm

Try FREE now

J .com (index.jsp)

Trang web chia sẻ nội dung miễn phí dành cho người Việt.

Lớp (../series/lop-3.jsp)	3	Lớp (../series/lop-4.jsp)	4	Lớp (../series/lop-5.jsp)	5	Lớp (../series/lop-6.jsp)	6	Lớp (../series/lop-7.jsp)	7	Lớp (../series/lop-8.jsp)	8
Lớp (../series/lop-9.jsp)	9	Lớp (../series/lop-10.jsp)	10	Lớp (../series/lop-11.jsp)	11	Lớp (../series/lop-12.jsp)	12	IT - Lập trình (../series/it-lap-trinh.jsp)		Tiếng Anh (../series/mon-tieng-anh.jsp)	

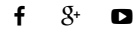
Liên hệ với chúng tôi

📍 66 Kim Hoa, Đống Đa, Hà Nội

📞 Phone: 01689933602

✉ Email: vietjackteam@gmail.com
(mailto:vietjackteam@gmail.com)

Tuyển dụng (<http://vietjack.com/lien-he/tuyen-dung.jsp>)



Về chúng tôi (<http://vietjack.com/lien-he/>)

