BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP CƠ SỞ**

**ĐỀ TÀI: Viết chương trình mô phỏng thuật toán duyệt cây nhị phân trên môi trường đồ họa.**

**Giảng viên hướng dẫn : Nguyễn Đình Hưng**

**Sinh viên thực hiện : Nguyễn Việt Hưng**

**Lớp : 61-CNTT-1**

**Cơ sở lý thuyết**

1. **Cây là gì:**

* Cây là một tập hữu hạn các node có cùng kiểu dữ liệu, trong đó có 1 node đặc biệt gọi là node gốc (root).
* Giữa các node có một quan hệ phân cấp gọi là “quan hệ cha con”.

Định nghĩa cây dưới dạng đệ quy:

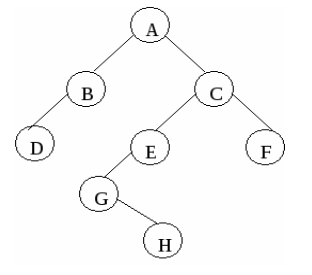
* Một node là 1 cây, node đó cũng là gốc (root) của cây đó.
* Cây gồm 1 node kết hợp với một số cây con bên dưới.
* Cây rỗng: Cây không có bất kỳ một node nào.
* Các node được gọi là cùng 1 cây khi có đường đi giữa các node này.
* Mỗi node trong cây (trừ node gốc) có duy nhất 1 node nằm trên nó, gọi là node cha.
* Node con: Các node nằm ngay dưới 1 node.
* Node lá: Các node không có con.
* Mức của node: node gốc có mức là 0. Nếu node cha có mức là i thì node con có mức là i+1.
* Chiều cao (sâu) của cây: Số mức lớn nhất của node có trên cây

1. **Mô tả và cách duyệt cây:**

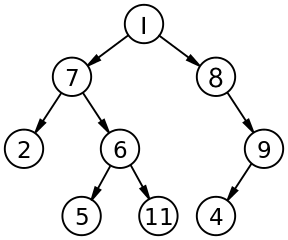
Duyệt cây là một tiến trình để truy cập tất cả các nút của một cây và cũng có thể in các giá trị của các nút này. Bởi vì tất cả các nút được kết nối thông qua các cạnh (hoặc các link), nên chúng ta luôn luôn bắt đầu truy cập từ nút gốc. Do đó, chúng ta không thể truy cập ngẫu nhiên bất kỳ nút nào trong cây. Có ba phương thức mà chúng ta có thể sử dụng để duyệt một cây:

Các tên gọi quy ước trong một node của cây nhị phân:

* **Root:** là node hiện tại đang xét.
* **Left:** là node con bên trái của node đang xét.
* **Right:** là node con bên phải của node đang xét.

**Cấu trúc của một cây nhị phân:**

 Có 2 cách để duyệt một cây nhị phân đó là **duyệt sâu** (depth first traversal) và **duyệt rộng** (breadth first traversal)



Đối với cách **duyệt sâu**, ta có 3 phương pháp khác nhau, phân loại dựa theo thứ tự thăm (visit) các node con của cây:

* **In-order:** Duyệt theo thứ tự **Left -> Root -> Right**. Ví dụ cây ở hình trên, thứ tự duyệt sẽ là: 2, 7, 5, 6, 11, 1, 8, 4, 9.
* **Pre-order:** Duyệt theo thứ tự **Root -> Left -> Right**. Ví dụ ở cây trên, thứ tự duyệt là: 1, 7, 2, 6, 5, 11, 8, 9, 4.
* **Post-order:** Duyệt theo thứ tự **Left -> Right -> Root**. Ví dụ ở cây trên, thứ tự duyệt là: 2, 5, 11, 6, 7, 4, 9, 8, 1.

**Duyệt rộng** thì ta sẽ đi từng level của cây, và duyệt hết tất cả các node ở từng level. Ví dụ cây trên thứ tự duyệt sẽ là: 1, 7, 8, 2, 6, 9, 5, 11, 4.

1. **Phân tích cây nhị phân gồm:**

**Trường hợp đối với cây nhị phân tìm kiếm:**

* Cây con bên trái của một Node luôn luôn có giá trị nhỏ hơn hoặc bằng giá trị của Node cha phía trên nó.
* Cây con bên phải của một Node luôn luôn có giá trị lớn hơn giá trị của Node cha phía trên nó
* Tất cả các cây con(bao gồm bên trái và phải) cũng đều phải đảm bảo 2 tính chất trên.

**Tìm kiếm 1 node trên cây NPTK**

* Để tìm kiếm 1 node có giá trị x trên cây NPTK, so sánh giá trị của node gốc với x:
* Nếu node gốc = NULL ⇒ không có x trên cây.
* Ngược lại, nếu x bằng giá trị node gốc ⇒ dừng, tìm được node chứa x.
* Nếu x > giá trị node gốc ⇒ tìm x trên cây con bên phải.
* Nếu x < giá trị node gốc ⇒ tìm x trên cây con bên trái.

**Thêm 1 node vào cây NPTK**

* Thuật toán thêm 1 node có giá trị x vào cây NPTK:
* Nếu node gốc = NULL: x chưa có trên cây ⇒ thêm node mới chứa x.
* Nếu x trùng với node gốc ⇒ không thể thêm.
* Nếu x < node gốc ⇒ thêm x ở cây con bên trái.
* Nếu x > node gốc ⇒ thêm x ở cây con bên phải.

**Xóa 1 phần tử x trên cây NPTK**

* Thuật toán xóa 1 node:
* Nếu không tìm thấy node chứa giá trị x ⇒ dừng.
* Nếu gặp node p chứa x:
* TH1: Nếu p là node lá ⇒ hủy node p.
* TH2: Nếu p chỉ có 1 node con ⇒ cho tree chỉ đến node con, hủy p.
* TH3: Nếu p có đủ 2 node con, tìm node nhỏ nhất ở cây con bên phải (hoặc node lớn nhất ở cây con bên trái), thay thế giá trị tại node p bằng giá trị của node nhỏ nhất ở cây con bên phải, sau đó xóa node nhỏ nhất.