Câu 3:

a.Cây nhị phân tìm kiếm

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 :

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

b.Thao tác thực hiện cây nhị phân tìm kiếm

\* Thao tác thực hiện tốt nhất là thao tác tìm kiếm  
− Tìm kiếm một nút trên cây là một trong các phép toán quan trọng nhất đối với cây nhị phân tìm kiếm.

\* Bài toán:

+Đầu vào: Cây nhị phân tìm kiếm có gốc được trỏ bởi con trỏ Root, khóa của nút cần tìm là K.

+ Đầu ra: Con trỏ trỏ vào nút tìm thấy, hoặc con trỏ NULL nếu không thấy.

• Nếu cây rỗng -> return NULL.

• Ngược lại so sánh khóa của nút gốc với khóa K.

− Nếu bằng -> return Root.

− Nếu lớn hơn -> tìm kiếm nút K ở cây con trái.

− Nếu nhỏ hơn -> tìm kiếm nút K ở cây con phải.

* Đệ Quy

TRO Search(TRO Root, KeyType K)

{

if (Root ==NULL)   
 return NULL;   
 else if (Root->infor==K)  
 return Root;  
 else if (Root->infor>K)  
 return Search(Root->left,K);  
 else return Search(Root->right,K);

}

* Lặp

TRO Search(TRO Root, KeyType K) {   
 TRO Q;  
 Q = Root;   
 while (Q != NULL && Q->infor != K)

{  
 if (Q->infor > K)  
 Q = Q->left;  
 else Q = Q->right;

}

return Q;   
}

c.Hạn chế  
 - Hạn chế của kiểu này là:

+ Có độ phức tạp trung bình

+ Phụ thuộc hoàn toàn vào dãy khóa đưa vào(trong quá trình xử lí động ta không biết trước được cây sẽ phát triền ra sao ,hình dạng của nó sẽ như thế nào.

Bài 12

a.Cấu trúc dữ liệu bằng cây nhị phân tìm kiếm để lưu trữ nhân viên

struct NhanVien

{

int MaNV;

char TenNV[20];

};

typedef struct NhanVien NV;

struct Node

{

NV infor;

struct Node \*L;

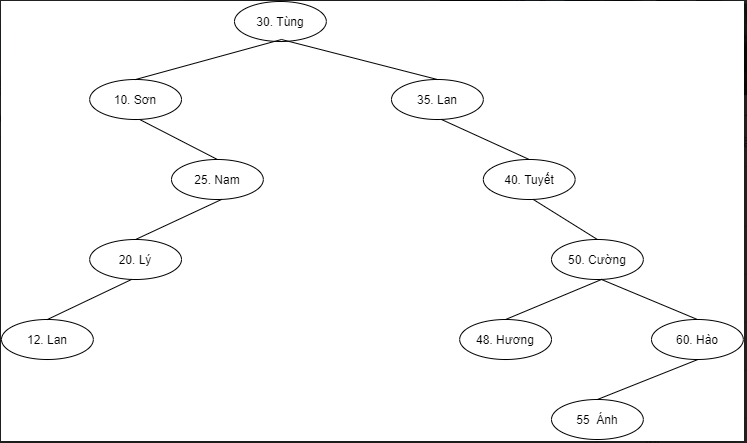
struct Node \*R;

};

typedef struct Node \*TRO;

TRO Root;

b.Cây nhị phân tìm kiếm với mỗi nút chứa thông tin của một nhân viên và mã nhân viên



c. Hàm hiển thị thông tin của các nhân viên có tên Lan

TRO Search(TRO Root )

{

if (Root ==NULL)

return NULL;

else

{

if(stricmp(Root->infor.TenNV,"Lan")==0)

cout<<"\nma Nv la:"<<Root->infor.MaNV<<endl;

Search(Root->L);

Search(Root->R);

}

}