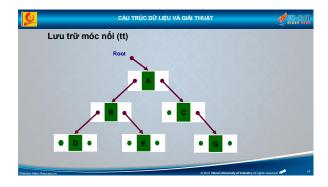
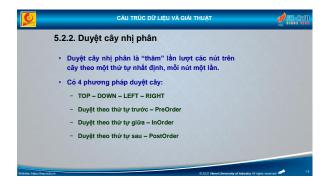


```
CAU TRÜC DỮ LIỆU VÀ GIAI THUẬT

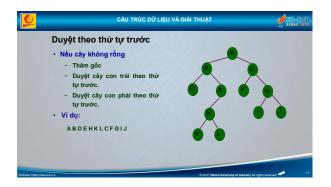
Lưu trữ móc nối (tt)

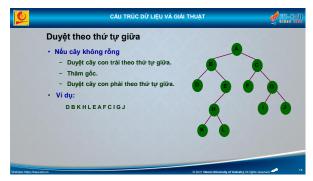
//Khai báo kiểu dữ liệu DataType
struct Node
{
    DataType infor;
    Node *left, *right;
    };
    typedef Node *TRO;
    TRO Root;
    //Root = NULL -> cây rỗng
```

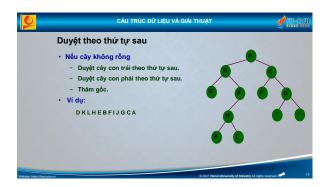


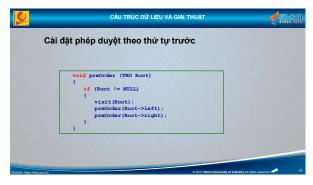


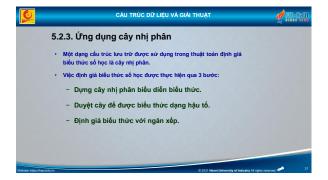


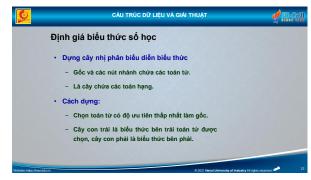


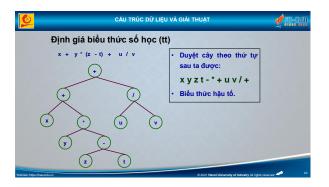






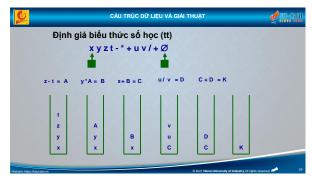


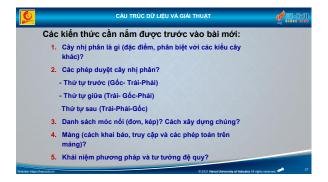




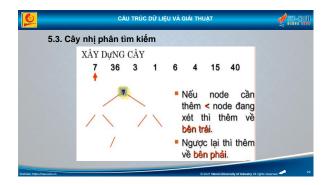


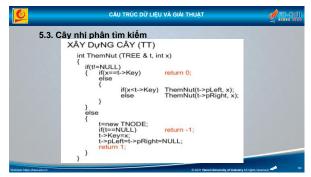












```
5.3.2. Cài đặt cấu trúc dữ liệu

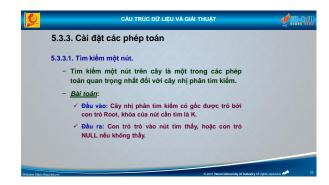
• Mỗi nút trên cây nhị phân tim kiếm có dạng.

[eft infor right]

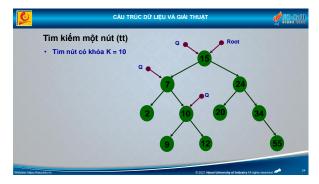
//Khai báo kiểu dữ liệu DataType

struct Node{
   DataType infor;
   Node *left, *right;
   };

typedef Node *TRO; //Định nghĩa kiểu con tró
   TRO Root; //Khai báo con tró trò vào gốc cây
   Root = NULL -> cây rỗng
```







```
CAU TRÚC DÛ LIỆU VÀ GIÁI THUẬT

TÎM KIỂM MỘT NÚT (tt)

Hầm đệ quy

TRO search (TRO Root, KeyType K)

{
    if (Root == NULL)
        return NULL;
    else if (Root->infor)
        return Root->infor)
        return search (Root->left, K);
    else return search (Root->right, K);
}
```

```
CAU TRÜC DÜ LIEU VÀ GIAI THUAT

TÎM KIẾM MỘT NÚT (tt)

• Hàm lập

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;

}
```

