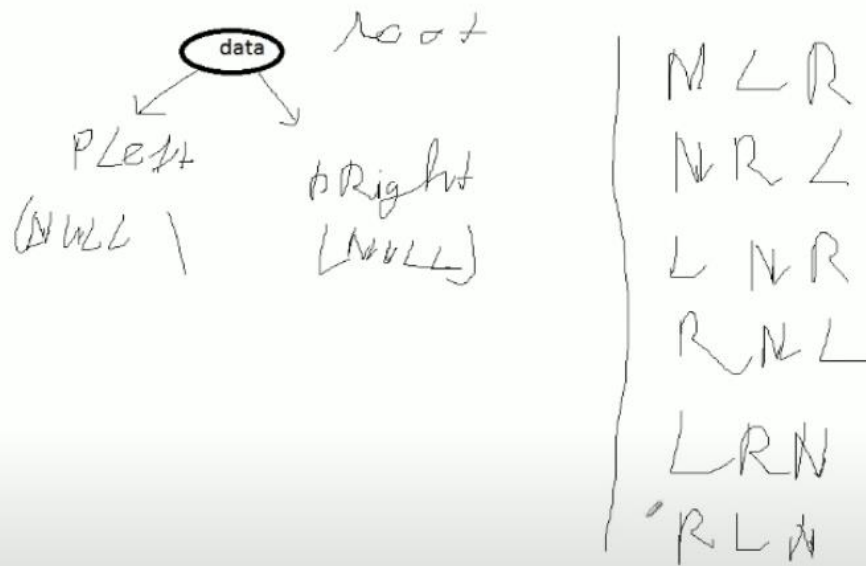


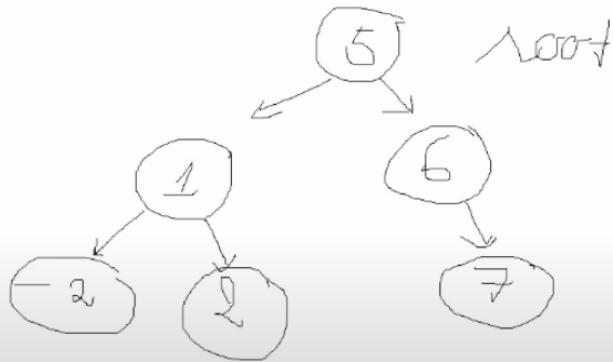
Cây nhị phân tìm kiếm



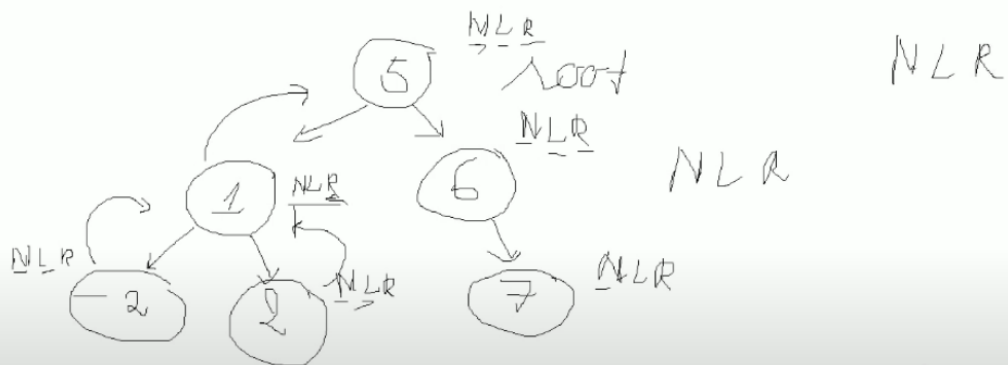
Duyệt NLR

```
// hàm xuất cây nhị phân theo NLR
void Duyệt_NLR(TREE t)
{
    // nếu cây còn phần tử thì tiếp tục duyệt
    if (t != NULL)
    {
        cout << t->data << " "; // xuất dữ liệu trong node
        Duyệt_NLR(t->pLeft); // duyệt qua trái
        Duyệt_NLR(t->pRight); // duyệt qua phải
    }
}
```

5 1 2 -2 6 7



5 1 2 -2 6 7



5 1 -2 2 6 7

```
C:\Windows\system32\cmd.exe

          ===== MENU =====
1. Nhập dữ liệu
2. Xuất dữ liệu cây NLR
0. Kết thúc

          =====
Nhập lựa chọn: 2

          DUYỆT CÂY THEO NLR
5 1 -2 2 6 7 Press any key to continue . . .
```

6 các duyệt cây nhị phân

- NLR
- NRL
- LNR
- RNL
- RLN
- LRN

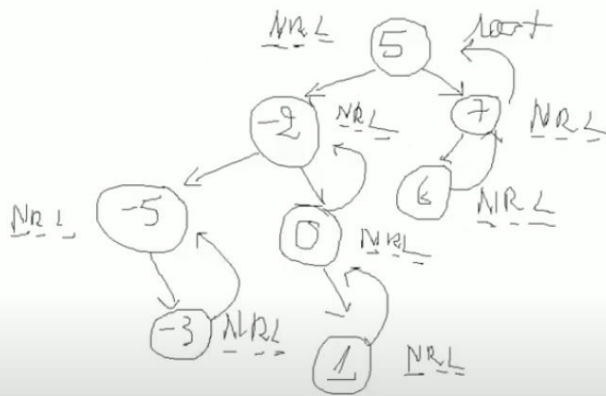
Duyệt NRL

```

// hàm xuất cây nhị phân theo NRL
void Duyet_NRL(TREE t)
{
    // nếu cây còn phần tử thì tiếp tục duyệt
    if (t != NULL)
    {
        cout << t->data << " "; // xuất dữ liệu trong node
        Duyet_NRL(t->pRight); // duyệt qua phải
        Duyet_NRL(t->pLeft); // duyệt qua trái
    }
}

```

5 7 -2 0 -5 1 6 -3



NRL

NRL: 5 7 6 -2 0 1 -5 -3

```
C:\Windows\system32\cmd.exe

***** MENU *****
1. Nhap du lieu
2. Duyet cay NLR
3. Duyet cay NRL
0. Ket thuc

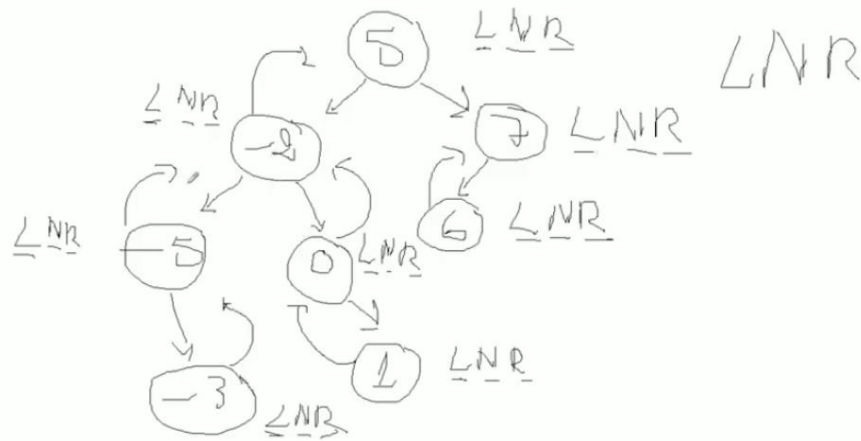
Nhap lua chon: 3

*****
                DUYET CAY THEO NRL
S 7 6 -2 0 1 -5 -3 Press any key to continue . . .
```

Duyệt LNR

```
// hàm xuất cây nhị phân theo LNR <=> xuất ra các phần tử từ bé đến lớn
void Duyet_LNR(TREE t)
{
    // nếu cây còn phần tử thì tiếp tục duyệt
    if (t != NULL)
    {
        Duyet_LNR(t->pLeft); // duyệt qua trái
        cout << t->data << " "; // xuất giá trị của node
        Duyet_LNR(t->pRight); // duyệt qua phải
    }
}
```

5 7 -2 0 -5 1 6 -3



LNR: -5 -3 -2 0 1 5 6 7

Duyệt RNL

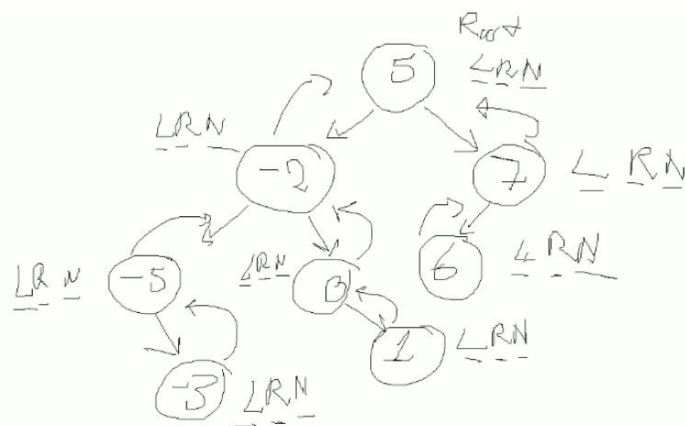
```
// hàm xuất cây nhị phân theo RNL <=> xuất ra các phần tử từ lớn đến bé
void Duyệt_RNL(TREE t)
{
    // nếu cây còn phần tử thì tiếp tục duyệt
    if (t != NULL)
    {
        Duyệt_RNL(t->pRight); // duyệt qua phải
        cout << t->data << " "; // xuất giá trị của node
        Duyệt_RNL(t->pLeft); // duyệt qua trái
    }
}
```



```

// hàm xuất cây nhị phân theo LRN
void Duyet_LRN(TREE t)
{
    // nếu cây còn phần tử thì tiếp tục duyệt
    if (t != NULL)
    {
        Duyet_LRN(t->pLeft); // duyệt qua trái
        Duyet_LRN(t->pRight); // duyệt qua phải
        cout << t->data << " "; // xuất giá trị của node
    }
}

```



LRN

LRN: -3 -5 1 0 -2 6 7 5

```

C:\Windows\system32\cmd.exe
***** MENU *****
1. Nhập dữ liệu
2. Duyệt cây NLR
3. Duyệt cây NRL
4. Duyệt cây LNR
5. Duyệt cây RNL
6. Duyệt cây LRN
0. Kết thúc

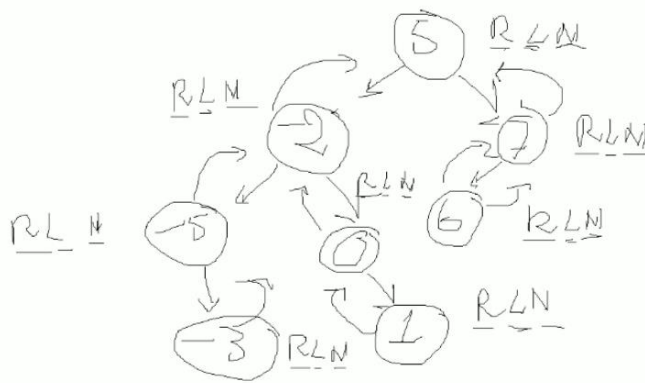
Nhập lựa chọn: 6

DUYET CAY THEO LRN
-3 -5 1 0 -2 6 7 5 Press any key to continue . . .

```


Duyệt RLN

```
// hàm xuất cây nhị phân theo RLN
void Duyệt_RLN(TREE t)
{
    // nếu cây còn phần tử thì tiếp tục duyệt
    if (t != NULL)
    {
        Duyệt_RLN(t->pRight); // duyệt qua phải
        Duyệt_RLN(t->pLeft); // duyệt qua trái
        cout << t->data << " "; // xuất giá trị của node
    }
}
```



RLN

RLN: 6 7 1 0 -3 -5 -2 5

```
C:\Windows\system32\cmd.exe

===== MENU =====
1. Nhap du lieu
2. Duyet cay NLR
3. Duyet cay NRL
4. Duyet cay LNR
5. Duyet cay RNL
6. Duyet cay LRN
7. Duyet cay RLN
0. Ket thuc

Nhap lua chon: 7

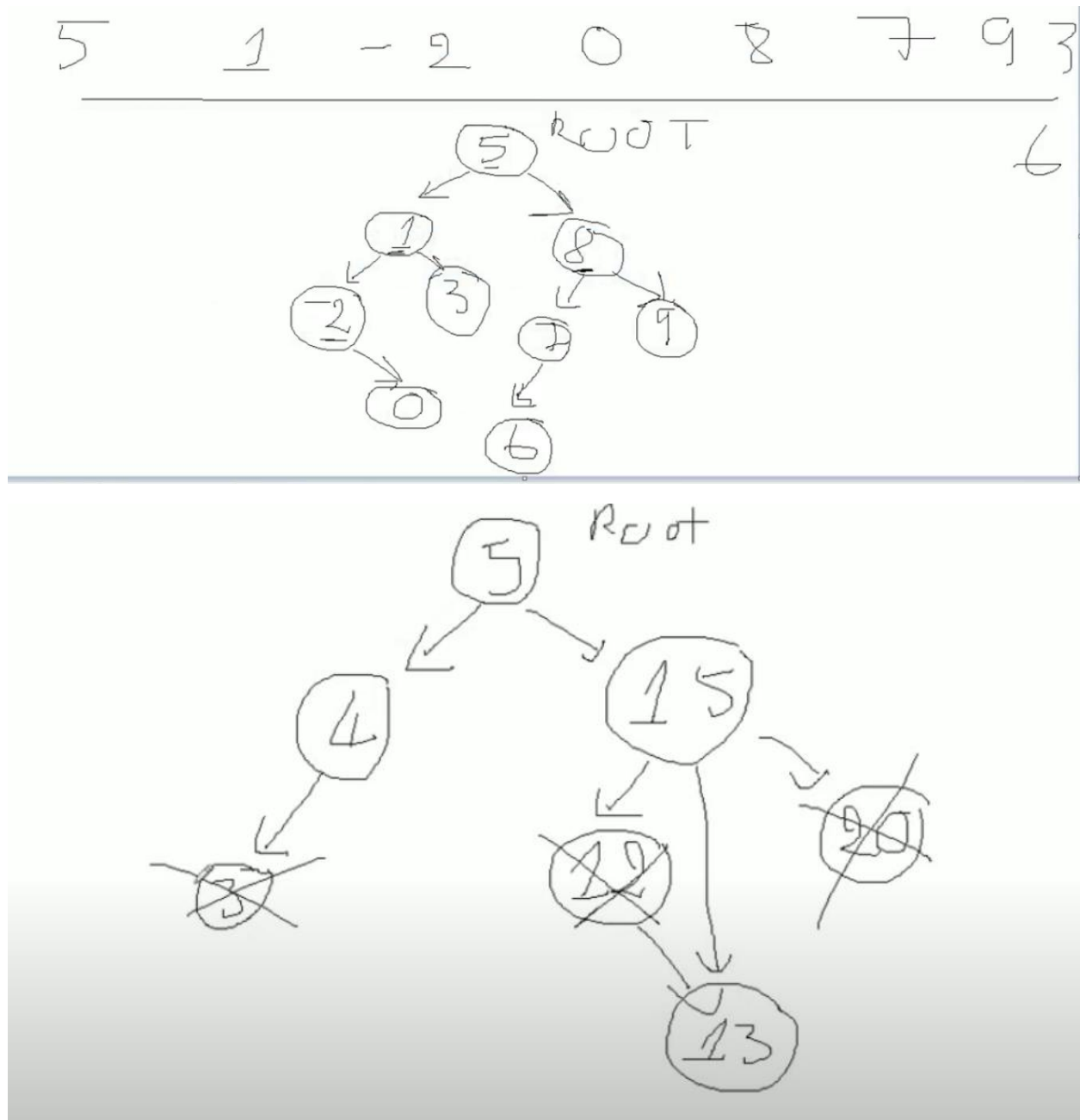
DUYET CAY THEO RLN
6 7 1 0 -3 -5 -2 5 Press any key to continue . . .
```

- Tìm kiếm trên cây nhị phân
 - Xuất các Node có 2 con
 - Xuất các Node có 1 con
 - Xuất các Node lá
 - Tìm MAX- Tìm MIN
 - Xóa
-

kiểm trên cây nhị phân
: các Node có 2 con(có tồn tại cây con trái và cây con phải)
: các Node có 1 con(có tồn tại cây con trái và không tồn tại cây con phải)
: các Node lá(node không tồn tại cây con trái và cây con phải)
MAX- Tìm MIN

....
CÂY

Xóa node con



Xóa node có 2 cây con trái và phải

