20220614-数据结构

1.学习内容

1.1 数据结构

二叉树

C++ 复制代码

```cpp
#define BinNodePosi(T) BinNode<T>* //节点位置
#define stature(p)((p)?(p)->height:-1)//节点高度
#define IsRoot(x)(!((x).parent))
#define IsLChild(x)(!IsRoot(x)&&(&(x)==(x).parent->lc))
#define IsRChild(x)(!IsRoot(x)&&(&(x)==(x).parent->rc))
#define HasParent(x)(!IsRoot(x))
#define HasLChild(x)((x).lc)
#define HasRChild(x)((x).rc)
#define HasChild(x)(HasLChild(x)||HasRChild(x))
#define HasBothChild(x)(HasLChild(x)&&HasRChild(x))
#define IsLeaf(x)(!HasChild(x))
#define sibling(p)(IsLChild(*(p))?(p)->parent->rc:(p)->parent->lc)
#define uncle(x)(IsChild(*((x)->parent))?(x)->parent->parent->rc:(x)->parent->parent->lc)
#define FromParentTo(x)(IsRoot(x)?_root:(IsLChild(x)?(x).parent->lc:(x).parent->rc))



typedef enum {RB_RED,RB_BLACK} RBColor;//节点颜色

template <typename T> struct BinNode
{
    T data;
    BinNodePosi(T) parent;//父节点
    BinNodePosi(T) lc;//左孩子
    BinNodePosi(T) rc;//右孩子
    int height;
    int npl;//null path length, 左式堆
    RBColor color;
    //构造函数
    BinNode():
        parent(NULL),lc(NULL),rc(NULL),height(0),npl(1),color(RB_RED){}
    BinNode(T e,BinNodePosi(T) p=NULL,BinNodePosi(T) lc=NULL,BinNodePosi(T) rc=NULL,int h=0,
        int l=1,RBColor c=RB_RED):
        data(e),parent(p),lc(lc),rc(rc),height(h),npl(l),color(c){}
    //操作接口
    int size();//统计当前节点后代总数
    BinNodePosi(T) insertAsLC(T const&);
    BinNodePosi(T) insertAsRC(T const&);
    BinNodePosi(T) succ();
    template<typename VST> void travLevel(VST&);
    template<typename VST> void travPre(VST&);
    template<typename VST> void travIn(VST&);
```

```
43        template<typename VST> void travPost(VST&);
44        bool operator<(BinNode const& bn) { return data < bn.data; }
45        bool operator==(BinNode const& bn) { return data == bn.data; }
46    };
47
48    template<typename T> BinNodePosi(T) BinNode<T>::insertAsLC(T const& e)
49    {
50        return lc = new BinNode(e, this);
51    }
52    template<typename T> BinNodePosi(T) BinNode<T>::insertAsRC(T const& e)
53    {
54        return rc = new BinNode(e, this);
55    }
56
57    template<typename T> template<typename VST>
58    void BinNode<T>::travIn(VST& visit)
59    {
60        switch (rand() % 5)
61        {
62        case 1:travIn_I1(this, visit); break;
63        case 2:travIn_I2(this, visit); break;
64        case 3:travIn_I3(this, visit); break;
65        case 4:travIn_I4(this, visit); break;
66        default:travIn_R(this, visit); break;
67        }
68    }
```

2.结果描述

今天依旧在学习二叉树。