

● 代码实践

Leetcode

106. 从中序与后序遍历序列构造二叉树

```
class Solution {
public:
    TreeNode* helper(vector<int>& inorder,int l1,int r1, vector<int>& postorder,int r2)
    {
        if(l1>r1) return NULL;
        TreeNode* root=new TreeNode(postorder[r2]);
        for(int i=l1;i<=r1;i++)
        {
            if(postorder[r2]==inorder[i])
            {
                root->left=helper(inorder,l1,i-1,postorder,r2-(r1-i)-1);
                root->right=helper(inorder,i+1,r1,postorder,r2-1);
                break;
            }
        }
        return root;
    }
    TreeNode* buildTree(vector<int>& inorder, vector<int>& postorder) {
        return helper(inorder,0,inorder.size()-1,postorder,postorder.size()-1);
    }
};
```

105. 从前序与中序遍历序列构造二叉树

```
class Solution {
public:
    TreeNode* helper(vector<int>& preorder, int l1,vector<int>& inorder,int l2,int r2)
    {
        if(l2>r2) return NULL;
        TreeNode* root=new TreeNode(preorder[l1]);
        for(int i=l2;i<=r2;i++)
        {
            if(preorder[l1]==inorder[i])
            {
                root->left=helper(preorder,l1+1,inorder,l2,i-1);
                root->right=helper(preorder,l1+i+1-l2,inorder,i+1,r2);
                break;
            }
        }
        return root;
    }
};
```

```

    }
    TreeNode* buildTree(vector<int>& preorder, vector<int>& inorder) {

        if(preorder.size()==0) return NULL;
        return helper(preorder,0,inorder,0,inorder.size()-1);
    }
};

```

● 计算机基础知识整理

死锁的定义：一个进程组的所有进程都在等待一个由该进程组的其他进程才能引发的事件，那么该组进程是死锁的。

产生死锁的必要条件：

- (1) 互斥
- (2) 请求和保持
- (3) 不可抢占
- (4) 循环等待

银行家算法：

我们可以把操作系统看作是银行家，操作系统管理的资源相当于银行家管理的资金，进程向操作系统请求分配资源相当于用户向银行家贷款。

为保证资金的安全，银行家规定：

- (1) 当一个顾客对资金的最大需求量不超过银行家现有的资金时就可接纳该顾客；
- (2) 顾客可以分期贷款，但贷款的总数不能超过最大需求量；
- (3) 当银行家现有的资金不能满足顾客尚需的贷款数额时，对顾客的贷款可推迟支付，但总能使顾客在有限的时间里得到贷款；
- (4) 当顾客得到所需的全部资金后，一定能在有限的时间里归还所有的资金。

银行家算法的基本思想是分配资源之前，判断系统是否是安全的；若是，才分配。它是最具有代表性的避免死锁的算法。

● 开源软件特训营总结

通过 Git 将 9 月 24 日作业文件添加到了 GitHub 仓库中，熟悉了相关 git 操作。

```
MINGW32:/c/Users/zy/Desktop/OSproject-miniprogram2020
zy@LAPTOP-6KBDVIUD MINGW32 ~/Desktop/osproject-miniprogram2020 (master)
$ git add .

zy@LAPTOP-6KBDVIUD MINGW32 ~/Desktop/osproject-miniprogram2020 (master)
$ git commit -m "0925"
[master 8b6a809] 0925
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "Notes/ZhaoYuan/\346\257\217\346\227\245\344\275\234\344\270\232\0924\344\275\234\344\270\232_\350\265\265\345\252\233.pdf"

zy@LAPTOP-6KBDVIUD MINGW32 ~/Desktop/osproject-miniprogram2020 (master)
$ git push origin master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 184.02 KiB | 23.00 MiB/s, done.
Total 6 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/zyy1225/osproject-miniprogram2020
fde3a58..8b6a809 master -> master

zy@LAPTOP-6KBDVIUD MINGW32 ~/Desktop/osproject-miniprogram2020 (master)
$
```