1.代码实践

递归的一些例题：

（1）二叉树的最大深度

Class Solution｛

public:

int maxepth(TreeNode\* root){

if(root==NULL) return 0;

if (maxDepth(root->left)>=maxDepth(root->right))

return maxDepth(root->left) + 1;

else

return maxDepth(root->right) + 1;

}

｝

（2）翻转二叉树

Class Solution {

public:

TreeNode\* invertTree(TreeNode\* root) {

if (root == nullptr) {

return nullptr;

}

TreeNode\* left = invertTree(root->left);

TreeNode\* right = invertTree(root->right);

root->left = right;

root->right = left;

return root;

}

};

（3）对称二叉树

class Solution {

public:

bool check(TreeNode \*p, TreeNode \*q) {

if (!p && !q) return true;

if (!p || !q) return false;

return p->val == q->val && check(p->left, q->right) && check(p->right, q->left);

}

bool isSymmetric(TreeNode\* root) {

return check(root, root);

}

};

（4）平衡二叉树

class Solution {

public:

int maxzhijing=0;

int depth(TreeNode \*root){

if(root==NULL) return 0;

return max(depth(root->left),depth(root->right))+1;

}

Void maxheight(TreeNode \*root){

If(root==NULL) return;

Int h= depth(root->left)+depth(root->right);

maxzhijing=maxzhijing>h?maxzhijing:h;

maxheight(root->left);

maxheight(root->right);

}

Int diameterofBinaryTree(TreeNode \*root){

Maxheight(root);

Return maxzhijing;

}

};

2. 知识点

（1）输入完网址按下回车，到看到网页这个过程中发生了什么

a. 域名解析

b. 发起TCP的3次握手

c. 建立TCP连接后发起http请求

d. 服务器端响应http请求，浏览器得到html代码

e. 浏览器解析html代码，并请求html代码中的资源

f. 浏览器对页面进行渲染呈现给用户

（2）进程和线程

定义：进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动,进程是系统进行资源分配和调度的一个独立单位.

线程是进程的一个实体,是CPU调度和分派的基本单位,它是比进程更小的能独立运行的基本单位.线程自己基本上不拥有系统资源,只拥有一点在运行中必不可少的资源(如程序计数器,一组寄存器和栈),但是它可与同属一个进程的其他的线程共享进程所拥有的全部资源.

（1）进程是对运行时程序的封装,是系统进行资源调度和分配的基本单位,实现操作系统的并发。

（2）线程是进程的子任务,是CPU调度和分派的基本单位,用于保证程序的实时性,实现进程内部的并发。

（3）一个程序至少有一个进程,一个进程至少有一个线程,线程依赖进程的存在。

（4）进程执行过程中拥有独立的内存单元,而多个线程共享进程的内存。

3. 开源项目

根据目前的情况重新调整了项目内容，并且重新规划了未来项目方向。

关于issue的学习：

# 一、 Issue的构成

提问、交流

提bug

提需求

# 二、异步沟通和信息沉淀

优点：跨时区合作、个人开发者时间成本低

​ 信息沉淀、可检

开发者：

（1）开发者提出issue之前先个人检索有无类似历史issue，为项目维护人员减轻工作量，降低沟通成本，保持项目正常运营。

（2）提交issue，内容尽量详实专业，换位思考，为社区维护者提供方便。

维护者：

（1）    维护良好的沟通环境 CODE\_OF\_CONDUCT.md

（2）    使用模板指引开发者提供更全面的信息

（3）    优化流程或采用一些自动化工具实现issue的更快响应

（4）    必要时可以采取措施屏蔽开源社区的恶意参与者。

挑战：沟通总体时间成本高、减少沟通的回合数、需要尽量在一次沟通中表达尽量多的内容

如何降低沟通成本？

- 提issue前搜索有没有相关历史

- 提交的内容尽量详实充分，bug说明运行环境、参数配置；最好提供日志和最小可复现仓库