

## ● 代码实践

Leetcode

117. 填充每个节点的下一个右侧节点指针 II

```
class Solution {
public:
    Node* get(Node* root)
    {
        if(!root) return NULL;
        if(root->left) return root->left;
        if(root->right) return root->right;
        if(root->next) return get(root->next);
        return NULL;
    }
    Node* connect(Node* root) {
        if(!root) return NULL;
        if(root->right&&root->left) root->left->next=root->right;
        if(root->left&&!root->right) root->left->next=get(root->next);
        if(root->right) root->right->next=get(root->next);
        connect(root->right);
        connect(root->left);
        return root;
    }
};
```

剑指 Offer 13. 机器人的运动范围

```
class Solution {
public:
    int count=0;
    bool record[110][110];
    int dir[4][2]={{0,1},{1,0},{-1,0},{0,-1}};
    bool isok(int x,int y,int m,int n,int k)
    {
        int tmp=0;
        while(x)
        {
            tmp+=x%10;
            x/=10;
        }
        while(y)
        {
            tmp+=y%10;
            y/=10;
        }
        if(tmp<=k)
```

```

        return true;
    return false;
}
void dfs(int x,int y,int m,int n,int k)
{
    count++;
    for(int i=0;i<4;i++)
    {
        int newx=x+dir[i][0];
        int newy=y+dir[i][1];
        if(newx>=0&&newx<m&&newy>=0&&newy<n&&!record[newx][newy]&&isok(
newx,newy,m,n,k))
        {
            record[newx][newy]=1;
            dfs(newx,newy,m,n,k);
        }
    }
}
int movingCount(int m, int n, int k) {
    record[0][0]=1;
    dfs(0,0,m,n,k);
    return count;
}
};

```

## ● 计算机基础知识整理

### 七层模型

**应用层：**网络服务与最终用户的一个接口。

协议有：HTTP FTP TFTP SMTP SNMP DNS TELNET HTTPS POP3 DHCP

**表示层：**数据的表示、安全、压缩。（在五层模型里面已经合并到了应用层）

格式有，JPEG、ASCII、EBCDIC、加密格式等 [2]

**会话层：**建立、管理、终止会话。（在五层模型里面已经合并到了应用层）

对应主机进程，指本地主机与远程主机正在进行的会话

**传输层：**定义传输数据的协议端口号，以及流控和差错校验。

协议有：TCP UDP，数据包一旦离开网卡即进入网络传输层

**网络层：**进行逻辑地址寻址，实现不同网络之间的路径选择。

协议有：ICMP IGMP IP（IPV4 IPV6）

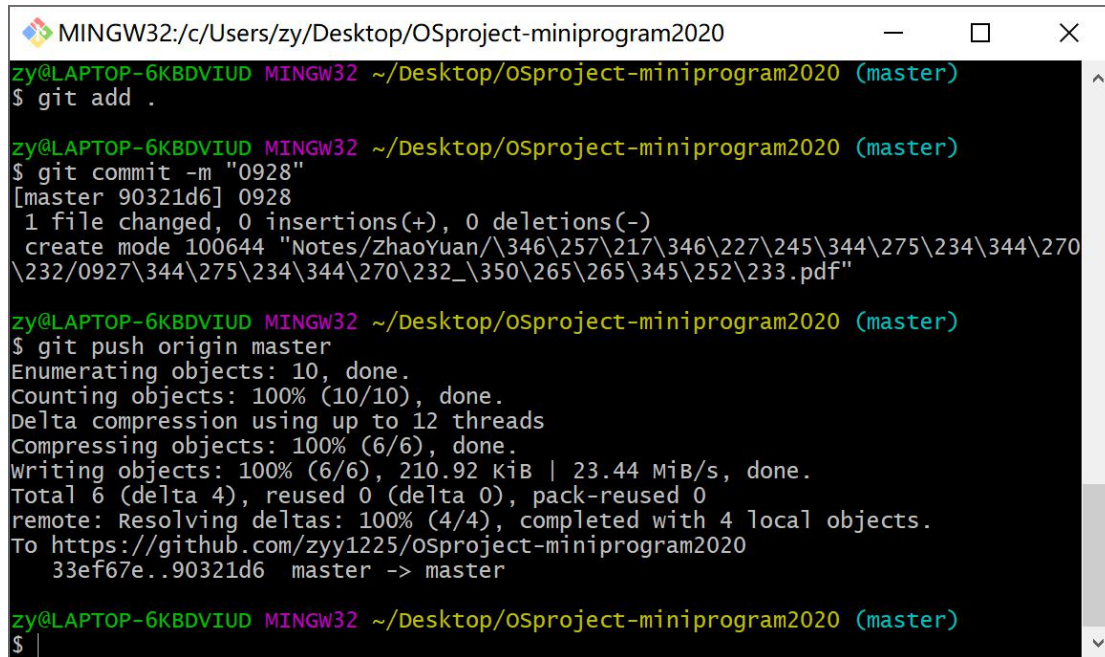
**数据链路层：**建立逻辑连接、进行硬件地址寻址、差错校验等功能。（由底层网络定义协议）

将比特组合成字节进而组合成帧，用 MAC 地址访问介质，错误发现但不能纠正。

**物理层：**建立、维护、断开物理连接。（由底层网络定义协议）

## ● 开源软件特训营总结

通过 Git 将 9 月 27 日作业文件添加到了 GitHub 仓库中，熟悉了相关 git 操作。

A screenshot of a Windows command prompt window titled "MINGW32:/c/Users/zy/Desktop/OSproject-miniprogram2020". The window shows a series of Git commands and their outputs. The user is in the directory ~/Desktop/osproject-miniprogram2020 on the master branch. They run 'git add .' to stage all files. Then they run 'git commit -m "0928"' to create a new commit. The output shows that one file was changed, and a new mode was created for a PDF file. Finally, they run 'git push origin master' to push the changes to the remote repository. The output shows the progress of enumerating, counting, compressing, and writing objects, and the successful push to the remote repository.

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

zy@LAPTOP-6KBDVIUD MINGW32 ~/Desktop/osproject-miniprogram2020 (master)
$ git commit -m "0928"
[master 90321d6] 0928
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "Notes/ZhaoYuan/\346\257\217\346\227\245\344\275\234\344\270\232\0927\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), 210.92 KiB | 23.44 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
33ef67e..90321d6 master -> master

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