方法一：修改Host

在DNS查询网站（例如）中查询github.com、github.global.ssl.fastly.net，记录ip地址

（一般直接去https://raw.hellogithub.com/hosts中获取，复制粘贴即可）

1. Hosts文件所在位置

```
Mac（苹果电脑）系统hosts位于 /etc/
Linux系统hosts位于 /etc/
Windows 系统位于 C:\Windows\System32\drivers\etc\
```

2. 文件的最后添加

```
记录的ip github.com
记录的ip github.global.ssl.fastly.net
```

3. 刷新dns缓存

```
windows输入：ipconfig /flushdns
Linux输入：sudo rcnscd restart
Mac OS X终端输入：sudo killall -HUP mDNSResponder
```

windows用户的同学可直接运行下面的python脚本即可实现GitHub的正常访问

```
import ctypes
import sys

def is_admin():
    try:
        return ctypes.windll.shell32.IsUserAnAdmin()
    except:
        return False

if is_admin():
    import requests
    from time import sleep
    import os
    import datetime
    r = requests.get("https://raw.hellogithub.com/hosts")
    now = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

    content = f"""
# Copyright (c) 1993-2009 Microsoft Corp.
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.

# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.

# Additionally, comments (such as these) may be inserted on individual
```

```
# lines or following the machine name denoted by a '#' symbol.

# For example:
#
# localhost name resolution is handled within DNS itself.
# 127.0.0.1 localhost
# ::1 localhost

{r.text}
    """
    with open('C:\\Windows\\System32\\drivers\\etc\\hosts', 'w+') as f:
        f.write(content)
        f.close()

    dnsFlush = "ipconfig /flushdns"
    os.system(dnsFlush)
    print("github DNS刷新成功")
    sleep(3)
else:
    # Re-run the program with admin rights
    ctypes.windll.shell32.ShellExecuteW(
        None, "runas", sys.executable, __file__, None, 1)
```

说明

只能对针对DNS污染
ip会变化，所以需要进行更换
要查询的域名其实不止那两个，只是两个例子，配置之后可以正常打开github，剩下要改的可以去自行了解，当然https://raw.hellogithub.com/hosts中给出的是全的

方法二：设置代理

1. 代理软件内设置

1. 在代理软件内勾选 `允许来自局域网的连接`
2. 记下端口号（例如：1080）
3. 开启 `全局模式`

2. 给 Git 全局配置代理

```
# 配置http代理
git config --global http.proxy http://127.0.0.1:【端口号】
git config --global https.proxy https://127.0.0.1:【端口号】
# 例如：git config --global http.proxy http://127.0.0.1:10808
#      git config --global https.proxy https://127.0.0.1:10808

# 配置socks5代理
git config --global http.proxy socks5://127.0.0.1:【端口号】
git config --global https.proxy socks5://127.0.0.1:【端口号】

# 如果只对 Github 进行代理，对国内的仓库不影响，例如gitee
git config --global http.https://github.com.proxy https://127.0.0.1:【端口号】
git config --global https.https://github.com.proxy https://127.0.0.1:【端口号】
# 只对 GitLab 进行代理，对国内的仓库不影响
git config --global https.https://https://gitlab.com.proxy
https://127.0.0.1:1080
```

查看配置文件的路径

```
git config -list -show-origin
```

恢复

如果不想用代理，可以用以下的方法恢复：

```
git config --global --unset http.proxygit config --global --unset https.proxy
```