

EdgeCloudSim仿真实验平台使用介绍

原创

[haha110ha](#) 于 2021-07-16 14:16:28 发布 3405 收藏 16

分类专栏: [MEC](#) 文章标签: [eclipse](#) [物联网](#) [云计算](#) [经验分享](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/haha110ha/article/details/118785205>

版权



[MEC](#) 专栏收录该内容

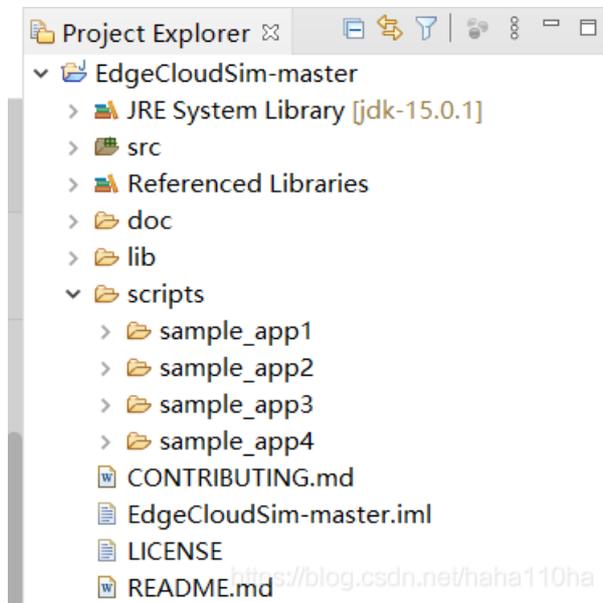
2 篇文章 1 订阅

订阅专栏

跑程序前的准备工作:

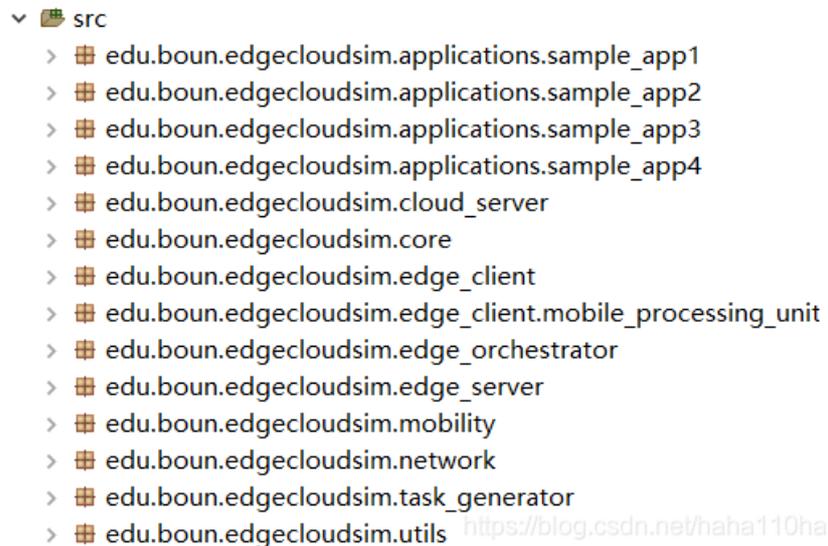
1. Java环境配置:网上搜教程吧, 按着提示做就可以
2. 我使用的工具: Eclipse
3. 项目链接: 其中包含EdgeCloudSim英文说明文档和EdgeCloudSim工程文件夹
链接: <https://pan.baidu.com/s/12dFGYggAymoKhfR0WRhwKg>
提取码: tcjp (项目)
链接: <https://pan.baidu.com/s/18cEBTHHtn67U-gPAKTeQlg>
提取码: 9zhx (英文说明文档)

1.项目整体结构



项目的导入很简单，直接把项目放到当前编辑器目录下就可以了。

上图所示就是EdgeCloudSim-master导入之后的目录了，作者在src文件夹下写了核心的代码，包括四个样例程序sample_app1-4。每一个样例都是可以自己跑通的，并会产生很多的日志文件和eclipse控制面板输出。



通过包的命名，我们可以知道各个包的含义是什么，比如edge_client,这当然就是边缘客户端的意思了，其他包类似翻译。正如英文文献中介绍的（英文文献在链接里，大家直接下载就好，还是建议大家仔细研读的，大体介绍的很清楚），该项目分为七个模块：

1. 核心仿真模块
2. 网络模块
3. 边缘协调器模块
4. 边缘协调器模块
5. 负载生成器模块
6. 核心仿真模块
7. 移动模块

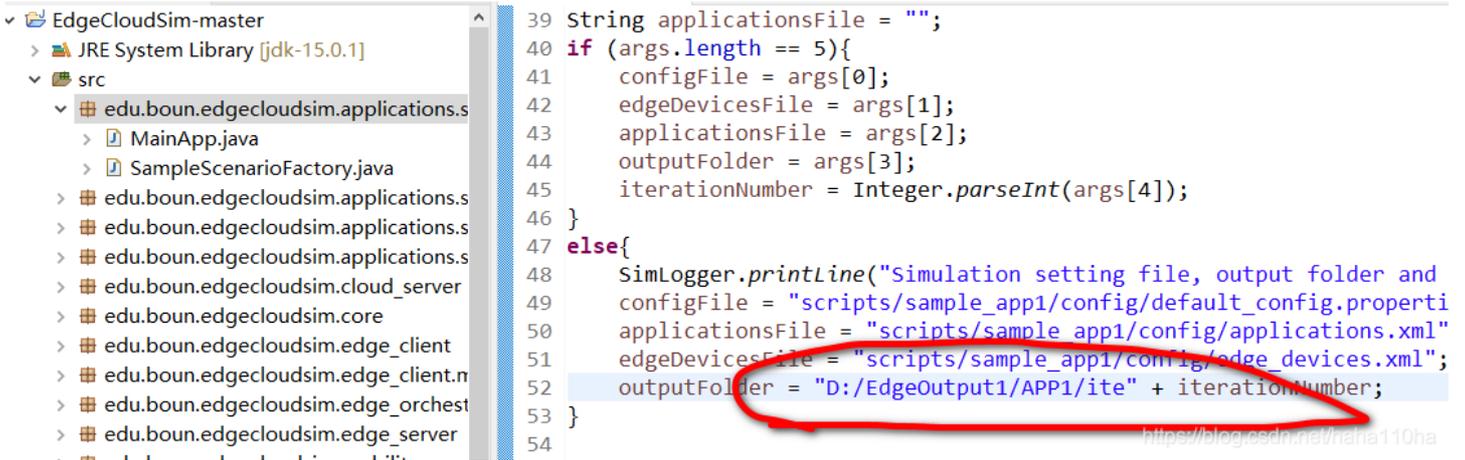
2.程序运行

当我们直接运行该项目也可以得到数据，如下图所示，不过说实话，看着数据是比较杂乱，我也不知道该怎么处理。

```
[2.0799999999999996 -2.4269104075787173 0.013025307906006325 -0.8942831427325004 7.
[2.0899999999999994 -2.4344920985453977 0.08172378844181016 -0.5080491562559051 6.3
[2.0999999999999999 -2.4376055251346593 0.14181535862773803 -0.013301439056361997 5.
[2.1099999999999999 -2.435606115460681 0.19197198500198184 0.4969043860859889 4.4395
[2.1199999999999988 -2.4285774859789826 0.2323803425134356 0.9808882514428331 3.503
[2.1299999999999986 -2.4168531316684527 0.26372736309301287 1.4293258721469675 2.64
[2.1399999999999983 -2.4007856023069465 0.286715927105407 1.8455795162234543 1.8392
[2.149999999999998 -2.380670775079413 0.3019102325501481 2.2364456574568043 1.08482
[2.159999999999998 -2.3567330586625954 0.3097096782636874 2.6088848455761666 0.3613
[2.1699999999999977 -2.329129698279227 0.31036251212457594 2.9691053075034786 -0.34
[2.1799999999999975 -2.29795966195811 0.3039887698613931 3.322367752814828 -1.04583
[2.1899999999999973 -2.26327362373867 0.29060546226149814 3.6728306509480992 -1.745
[2.199999999999997 -2.2250864681933002 0.2701573631697297 4.023045520880816 -2.4605
[2.209999999999997 -2.1833974497390383 0.24256463540842163 4.372676386996691 -3.178
[2.2199999999999966 -2.138228041117821 0.20780897573560492 4.715836216182483 -3.886
[2.2299999999999964 -2.0896940067100647 0.16609373108801856 5.036467137354494 -4.55
[2.239999999999996 -2.0381305691760034 0.11811794428372227 5.3026733465249825 -5.05
[2.249999999999996 -1.9842643945470013 0.06544920667682899 5.46616350505063 -5.4215
[2.259999999999996 -1.929334679083342 0.010782280286209138 5.4793702815871111 -5.4184
```

不过作者还是很牛的，想的很周到，所谓前人栽树后人乘凉。真的要感谢开发这个环境的大佬了，啧啧。

我们可以选择直接运行样例代码，注意项目的跑通核心之处在于修改日志的输出文件夹路径，如下图所示：



```
39 String applicationsFile = "";
40 if (args.length == 5){
41     configFile = args[0];
42     edgeDevicesFile = args[1];
43     applicationsFile = args[2];
44     outputFolder = args[3];
45     iterationNumber = Integer.parseInt(args[4]);
46 }
47 else{
48     SimLogger.println("Simulation setting file, output folder and
49     configFile = "scripts/sample_app1/config/default_config.properti
50     applicationsFile = "scripts/sample_app1/config/applications.xml"
51     edgeDevicesFile = "scripts/sample_app1/config/edge_devices.xml";
52     outputFolder = "D:/EdgeOutput1/APP1/ite" + iterationNumber;
53 }
54
```

上面的路径是一定要修改的，否则系统找不到原来指定的文件夹，就会报错！另外：

诚践徊鞠妃专霸呢C盞 = 困\餓丿弄姑孩甯齡鱸呢C盞 佢\叨忝竟任轍刀踐徊 = 给枢受
坪叙绥丿昕披赜 = 咄杜儉政二盞丕咄 = 鱸叵佗二 { 刂枋佗咄餓觀值呢C盞 肫侯捺杀
劫 -

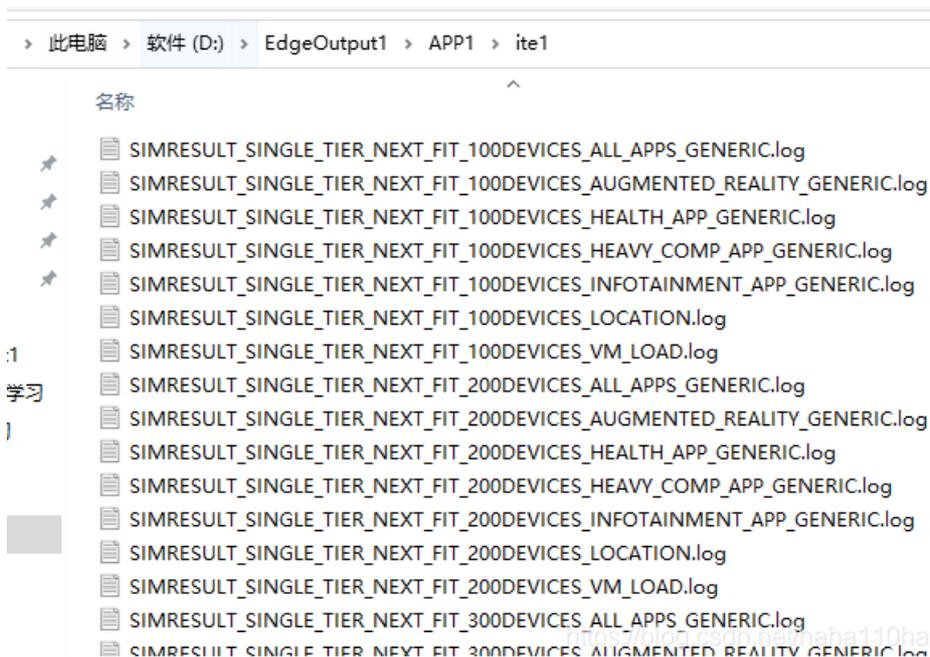
程序运行以后就会生成一堆日志文件，这个也很重要，因为后面作者还写了matlab绘图程序。在MatLab里直接调用函数，利用日志文件作图，爽歪歪，有一种办公自动化的感觉！

```

Creating tasks...Done,
Creating device locations...Done.
SimManager is starting...Done.
.....10.....20.....30.....40.....50.....60.....70
# of tasks (Edge/Cloud/Mobile): 49824(39640/10184/0)
# of failed tasks (Edge/Cloud/Mobile): 519(484/35/0)
# of completed tasks (Edge/Cloud/Mobile): 49305(39156/10149/0)
# of uncompleted tasks (Edge/Cloud/Mobile): 63(57/4/2)
# of failed tasks due to vm capacity (Edge/Cloud/Mobile): 0(0/0/0)
# of failed tasks due to Mobility/Network(WLAN/MAN/WAN): 519/0(0/0/0)
percentage of failed tasks: 1.041667%
average service time: 2.122846 seconds. (on Edge: 2.494956, on Cloud: 0.68720)
average processing time: 1.996365 seconds. (on Edge: 2.465088, on Cloud: 0.18)
average network delay: 0.126481 seconds. (LAN delay: 0.029868, MAN delay: NaN)
average server utilization Edge/Cloud/Mobile: 41.517559/0.602843/0.000000
average cost: 0.0$
Scenario finished at 16/07/2021 10:08:08. It took 16 Seconds
-----
Simulation finished at 16/07/2021 10:08:08. It took 1 Minute 44 Seconds

```

从上图可以看出，样例sample_app1在设备数量是500的情况下，整体运行时间是1分44秒。系统原来设定的是1000，结果我为了让它跑快点，设置成了500。这个设备数量设置在配置文件里，大家可以自行修改。



3.场景介绍

文章中介绍了实验对应的三个场景，在样例代码中，程序运行结果也有体现，代码输出那一块写的还是蛮清晰的。分别是：

1. SINGLE_TIER - Policy
2. TWO_TIER - Policy
3. TWO_TIER_WITH_EO - Policy

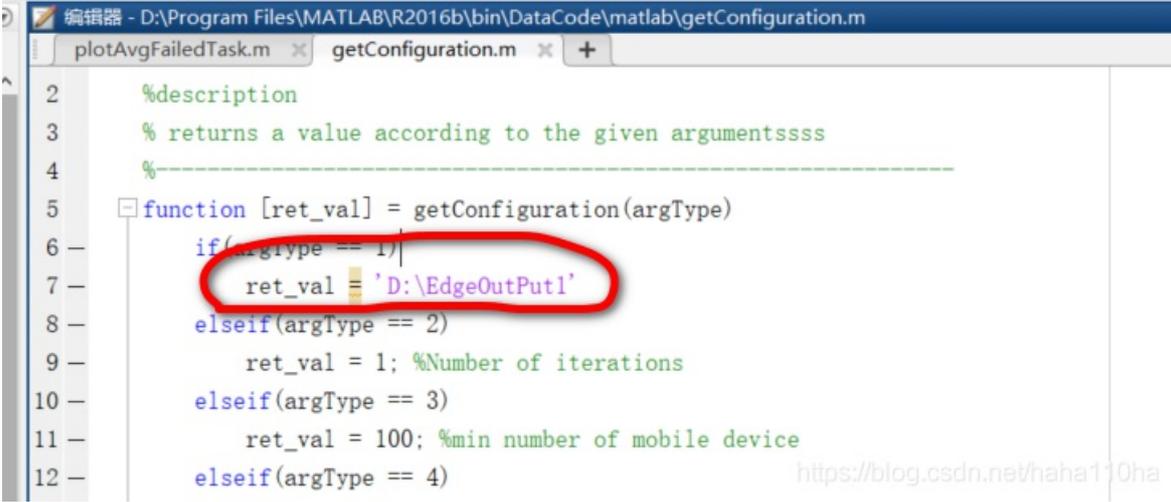
每一个场景的具体情况，大家可以在论文里查看。

4.数据可视化

不得不说作者很贴心，还给我们写了MatLab文件，方便我们对日志文件数据进行绘图。代码在scripts->sample_app1->matlab中，其实scripts文件夹下所有的样例都有matlab代码。

这一部分的使用，就要转移到MATLAB中去了，大家可以复制粘贴matlab文件夹到MATLAB中，我用的是R2016b版本。

同样的第一步先修改配置文件。即getConfiguration.m文件，这个文件很重要，配置了很多关键数据，所有的.m文件都是根据这个文件夹中的配置执行的。我在执行的过程中，还碰到个小问题，本来argType==1时，对应的ret_val应该设置为我的日志保存文件夹，但是系统又报错了！



```
2 %description
3 % returns a value according to the given argumentssss
4 %-----
5 function [ret_val] = getConfiguration(argType)
6     if(argType == 1)
7         ret_val = 'D:\EdgeOutPut1'
8     elseif(argType == 2)
9         ret_val = 1; %Number of iterations
10    elseif(argType == 3)
11        ret_val = 100; %min number of mobile device
12    elseif(argType == 4)
```

通往成功的道路不会一帆风顺的，总带被搓搓棱角。



```
命令窗口
>> plotAvgFailedTask

ret_val =

D:\EdgeOutPut1

filePath =

D:\EdgeOutPut1\itel\SIMRESULT_SINGLE_TIER_NEXT_FIT_100DEVICES_ALL_APPS_GENERIC.log

错误使用 error
信息必须指定为字符矢量或信息结构体。

出错 plotGenericResult (line 44)
error(err)
```

后来根据错误的信息，我修改了路径信息，以及最大设备数量，就可以了

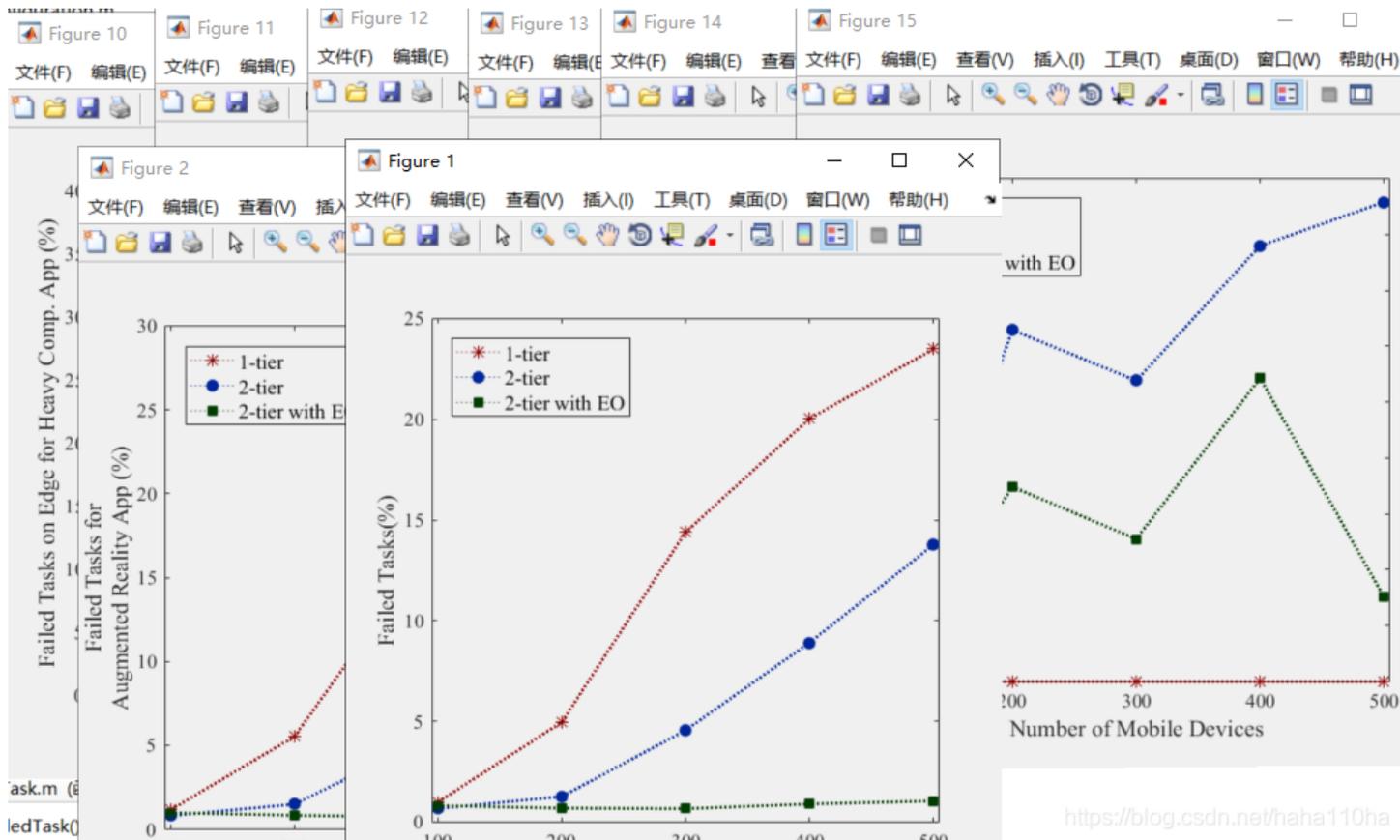
```

编辑器 - D:\Program Files\MATLAB\R2016b\bin\matlab\getConfiguration.m
getConfiguration.m x plotAvgFailedTask.m x +
4 %-----
5 function [ret_val] = getConfiguration(argType)
6     if(argType == 1)
7         ret_val = 'D:\EdgeOutPut1\APP1'
8     elseif(argType == 2)
9         ret_val = 1; %Number of iterations
10    elseif(argType == 3)
11        ret_val = 100; %min number of mobile device
12    elseif(argType == 4)
13        ret_val = 100; %step size of mobile device count
14    elseif(argType == 5)
15        ret_val = 500; %max number of mobile device
16    elseif(argType == 6)
17        ret_val = 1; %xTickLabelCoefficient
命令窗口

```

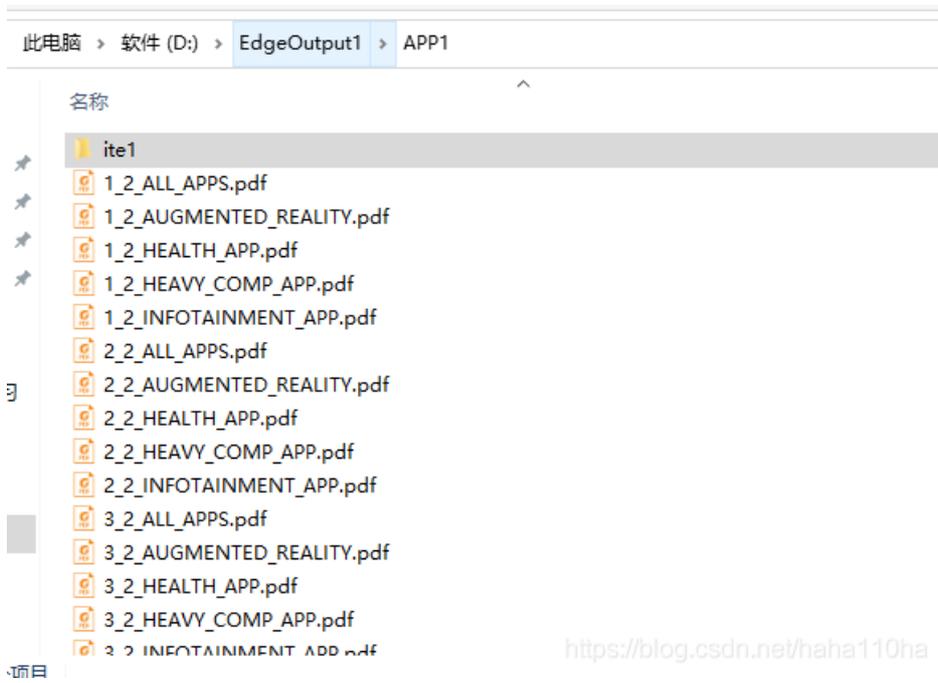
注意下我的日志文件夹，在APP1下又建了一个ite1文件夹。这个文件夹也是最终保存日志的文件夹。

结果出来了！



其实图片不是这么乱的，只是我为了让大家看着方便，把他们拉开了。

同时，你也可以看到系统自动生成了图片pdf版本，还是挺好的。



以上就是所有分享了，出现问题还是得多读读代码。本来昨天跑通了，今早为了写博客，重新搞了一遍，又出问题了。。。又改了一上午。说明一个问题：代码好了，只要能通就别动了，千万别动！