# SDN最优路由算法的答题提交规则和说明

## 一、综述

答案程序的测试运行环境、算法程序、判题程序、判题系统这四者之间，必须遵循一定的规则和契约。本文描述的是“答题者提交的算法答案程序”和“运行环境”、以及“判题程序”之间的约定和规则，请各位答题者务必仔细阅读。

判卷和测试运行的OS

提交的算法答案程序

测试和判题的程序 judge

判题系统和用户交互Web平台

本文描述的就是该部分和上下两层之间的接口。

如果违背本文描述的规则，很有可能导致自动判卷程序失败，从而没有得分。

## 二、运行环境

判题环境提供如下的硬件执行环境：

CPU Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz，内存：8G，8核心。

判题环境提供两种操作系统：

1. 64位 Windows 7 SP3
2. 64 位 OpenSUSE 11 SP3

答题者可以自己选择使用哪一个系统来判卷运行。在提交答案的时候，请在提交界面上选择你的答案所使用的操作系统；

## 三、编程语言的要求

本试题不对选手所使用的语言做限制，答题者可以使用任何自己拿手的语言来编程、提交答案；但是，应该遵循如下规则：

1. 如果使用编译型语言，必须提交编译后可直接执行的可执行程序；windows下是exe、linux下是编译后的二进制可执行文件。这些语言包括 go、C、C++、C#等；
2. 如果使用Java语言，请打包成 .jar 包。
3. 如果用到了第三方库，比如MFC、boost等、请使用静态链接的方式；确保可执行程序被执行起来；
4. 如果是解释型的脚本语言，那么必须可以通过文件后缀直接判断脚本类型；例如，ruby脚本应该是.rb，TCL脚本应该是.tcl，Python脚本是.py等。
5. 执行环境只会提供各脚本语言最新的stable版本的解释器，如果答题者使用脚本语言，请确保脚本所使用语法的兼容性；如果用到了第三方库，请在提交答案的时候，将第三方库一起打包提交。

## 四、判题程序的命名要求

1. 答题者实现算法的答案可能包含了很多文件，但是必须有一个可执行入口文件，该文件必须以sdn.\*的形式来命名（sdn必须是小写）。例如：
2. Windows下实现的答案，必须是 sdn.exe；
3. Linux 下编译出来的可执行文件，入口文件名就是 sdn；
4. 如果是脚本，必须命名为 sdn.tcl、sdn.rb、sdn.py等等；
5. 如果是Java，必须命名为 sdn.jar；
6. 提交答案的时候，必须先将答案打包，并且不能有密码。包内的目录结构、目录内容都不做限制，由答题者自行设计和决定；判题程序只关注 sdn 入口程序是否存在。压缩包的文件名，也不做限制和要求。
   1. Windows 下，以 zip 格式进行压缩；
   2. Linux下，以 tar.gz 的格式进行压缩；

## 五、答题程序的输入输出要求

答案程序必须接受如下命令行格式的参数。

**sdn node\_csv\_file link\_csv\_file service\_csv\_file output\_reslut\_file**

|  |  |
| --- | --- |
| 参数 | 参数含义 |
| node\_csv\_file | 网络节点数据的CSV文件的文件路径 |
| link\_csv\_file | 网络的链路数据的CSV文件的文件路径 |
| service\_csv\_file | 含有“业务的信息数据”的CSV文件的文件路径 |
| output\_result\_file | 结果输出文件的路径 |

1. 判题系统每次调用 sdn 一次，都会给出这四个参数，其中前面三个参数所指定的三个csv文件，组成了一个完整的网络描述信息。答题程序必须从这三个参数指定的文件中读取网路信息，进行计算；
2. 计算结束后，程序必须将运算结果按照题目要求，写入到第四个参数所指定的文件路径中。如果这个文件已经存在，则完全覆写它，否则创建它；
3. 这四个文件路径参数，有可能是绝对路径，也有可能是相对路径；如果是相对路径，则是相对于 sdn 可执行程序所在位置的相对路径；
4. 注意，这四个参数是文件路径，它们的顺序是确定的，第一个参数表示节点信息、第二个表示链路信息、第三个表示业务信息、第四个是输出文件。其文件名不是确定的。请务必根据参数位置和顺序来决定其文件内容以及含义，而不能依赖文件名。

前三个参数指定的 csv文件的内容格式，以及输出内容的格式，请参考题目的详细说明；

## 六、其它要求

1. sdn 必须是一个命令行程序，从命令行参数中接受测试用例信息，程序运行的时候，不能有任何键盘读取操作，不能有任何 GUI 的交互，运算结束后，进程必须自行退出。
2. 如果提交的答案程序中有等待键盘输入的操作，判题系统一律按超时处理。

## 七、举例

下面是一个答题者提交答案，判卷系统判卷的过程例子：

1. 答题者提交了一份Linux 下的答卷，该程序用C++写成，其中可执行入口文件是 sdn。
2. 答题者将试题打包成 myteam\_name.tar.gz。这个包里面包含了可执行程序 sdn，以及其它答题者认为必须的文件；
3. 答题者将myteam\_name.tar.gz 提交给竞赛平台；
4. 判题系统准备了 10 个用例来进行判题和测试。每一个用例都包含了“节点”，“链路”以及“业务信息”这三个csv文件。
5. 判题系统将答题者提交的 tar.gz 解压到某个目录下，例如 /opt/aa。
6. 判卷系统找到预先写好的该试题对应的判卷程序 judge（由出题者和主办方实现），按如下格式启动判卷程序：  
    judge /opt/aa
7. judge会针对每一个用例，都启动一次 sdn程序，并且将这个用例对应的三个 csv 文件作为命令行参数，传递给 sdn 进程，同时指定一个output.csv文件路径作为sdn的第四个参数，要求sdn必须将运算结果写入到这个规定的 output.csv 中去。Judge会根据output.csv中的内容，判断此用例是否测试通过。
8. judge依次测试多个用例，最后根据用例的通过情况以及其它维度（比如运行时间等）的成绩，汇总得出该份试卷的原始成绩，并且将这份原始成绩，通过和判卷系统协商好的接口与形式，反馈给判卷系统。

<完>