

1、整体指引：

- 1)使用一键式脚本编译、链接并打包压缩，如果编译失败请自行解决编译问题；
- 2)如果编译成功会在bin/目录下生成可执行二进制文件"cdn"；
- 3)使用如下格式调用并调试程序"./cdn /xxx/topo.txt /xxx/result.txt"，其中topo.txt是输入文件，result.txt是输出文件；
- 4)调试成功后到竞赛官网提交SDK-gcc/目录下的压缩包"cdn.tar.gz"，稍后查询成绩。

2、目录结构：

SDK-gcc/

└── bin/ 可执行二进制文件目录，shell脚本在编译前删除此目录并重新创建此目录，故没有此目录不会影响脚本运行

└── build/ 构建目录，shell脚本在编译前删除此目录并重新创建此目录，故没有此目录不会影响脚本运行

└── cdn/ 代码目录

│ └── lib/ lib头文件目录，此文件夹不可修改亦不可在其下新增文件

│ │ └── lib_io.h 读写文件的头文件，不可修改

│ │ └── lib_record.h 将输出结果记录到缓冲区的头文件，不可修改

│ │ └── lib_time.h 打印时间的头文件，不可修改

│ └── CMakeLists.txt cmake，不可修改

│ └── cdn.cpp main函数源文件，不可修改

│ └── io.cpp 提供读写文件等功能的源文件，不可修改

│ └── deploy.cpp 你要写代码的源文件

│ └── deploy.h 你要写代码的头文件

└── build.sh 编译、链接、打包批处理脚本，不可修改

└── readme.txt 你正在看的文件 -_- " 这不用介绍了吧

3、shell脚本说明：

执行此脚本可以一键编译、链接、打包。如果编译和链接正确，会在bin/下生成cdn二进制文件，并按照大赛要求生成代码源文件的压缩打包文件存处于SDK/下。

注意：

1)shell脚本会删除bin/和build/目录，以及这两个目录下的所有文件和目录。请不要在此保存你的任何文档；

2)如果想使用shell脚本一键功能，请保持SDK-gcc/目录下所有内容的完整，请不要修改任何目录名和文件名，并保持各目录和文件的位置关系不变。

4、手工操作说明：(非必须。如果选择使用shell脚本构建，可忽略本节内容)

1)在SDK-gcc/目录下创建build_private/目录，并在build_private/下编写makefile文件；

2)进入build_private/，执行make完成编译和链接；

3)将代码源文件打包压缩生成"cdn.tar.gz"。

注意：

1)不要在build/下保存你的makefile文件，一旦调用batch.sh将会删除你的makefile文件；

2)生成压缩包时，应确保将代码源文件置于压缩包的最外层，即打开压缩包无需进入任何目录即可看全部源文件，否则可能编译失败。

5、SDK代码说明：

我们已经提供了保姆式的服务，你只需要做：

1)实现deploy.cpp文件中的XXXX接口；

2)依次调用record_result将路径结果写入缓冲区；

3)如果要清空缓冲区，可以调用clear_result将缓冲区的内容清空，以便重新写入；

4)如果计算结果为没有路径，则不需要调用record_result接口即可直接输出NA。

SDK已经实现了读取文件、按要求格式写文件以及打印开始和结束时间的功能。

注意：读取文件功能是指，将链路信息文件按行读取到内存，其在内存中的存储格式仍是字符串格式。因为这些信息以什么格式存储涉及到算法设计，这样做是为了不禁锢你的思路。

6、重要提示：

复赛和决赛阶段不再提交可执行二进制文件，而是直接提交你修改和增加的代码源文件，在判题服务器端执行编译。因此需要特别注意：

1、必须基于本SDK开发，否则会编译不过；

2、在SDK的源文件中，只有deploy.cpp和deploy.h两个文件允许修改，其他文件在编译前统一使用官方SDK版本的文件替换；

3、如果需要新增文件，请一定要增加到与deploy.cpp文件同级目录下。

