南開大學
Nankai University

计算机学院

软件工程

结对编程作业用户手册

姓名：赵一名　宋昊雨

学号：2013922　2010249

专业：计算机科学与技术

2023 年 6 月 23 日

目录

1 概述

本文介绍了我们结对编程作业中的数独程序的相关使用方法

2 具体介绍

给出的压缩包中给出的 sudoku.exe 文件可直接在终端中运行，运行时需要添加相关的参数，可添加的参数与相应的效果在下面进行了逐一描述

2.1 -c 参数

-c 参数表示生成相应数量的数独终盘，其后必须有一个范围在 0-1000000 之间的整数，表示需要生成的终盘的数量，这个参数可以单独使用，使用这个参数的一个例子如下：

./sudoku.exe -c 10

这条命令会在当前文件夹下生成一个“result0.txt”文件，这个文件里存放了对应的输出。

2.2 -s 参数

-s 参数表示求解一个数独题目，其后跟一个文件的相对或绝对路径表示待求解的数独文件所在的路径，这个参数可以单独使用，使用这个参数的一个例子如下：

./sudoku.exe -s ./game.txt

这条命令将会读取当前文件夹下的 game.txt 文件，并求解这个文件终的数独问题，然后会在当前文件夹下生成一个“result.txt”文件，这个文件里存放了数独问题的答案。

2.3 -n 参数

-n 参数表示生成一定数量的数独游戏，其后必须接一个范围在 1-10000 之间整数，表示需要生成的游戏数量，这个参数可以单独使用，使用这个参数的一个例子如下：

./sudoku.exe -n 10

这条命令会在当前文件夹下生成一个 RandomSudokuQuestion.txt 文件，这个文件中将会有 10 个数独问题，其解是否唯一与难度均是随机的

2.4 -m 参数

-m 参数表示生成规定难度的数独游戏，其后必须接一个范围在 1-3 之间的整数表示数独的难度，这个参数不可以单独使用，必须跟在-n 参数后使用，，使用这个参数的一个例子如下：

./sudoku.exe -n 10 -m 1

这条命令将会在当前文件夹下生成一个 level.txt 文件，这个文件中有 10 个难度为 1 的数独游戏，但不一定只有唯一解

2.5 -r 参数

-m 参数表示生成规定难度的数独游戏，其后必须接两个个范围在 20 到 55 之间的两个整数表示数独挖空的个数，这个参数不可以单独使用，必须跟在-n 参数后使用，使用这个参数的一个例子如下：

./sudoku.exe -n 20 -r 20 55

这条命令将会在当前文件夹下生成一个 blank.txt 文件，这个文件中有 20 个挖空数在 20 到 55 之间的数独游戏，但不一定只有唯一解

2.6 -u 参数

-u 参数表示生成数独具有唯一解，这个参数不可以单独使用，必须跟在-n 参数后使用，使用这个参数的一个例子如下：

./sudoku.exe -n 20 -u

这条命令将会在当前文件夹下生成一个 unique.txt 文件，这个文件中有 20 个只有唯一解数独问题

3 质量分析

质量分析使用的工具是 cppcheck，这个工具可在 linux 平台上之间使用 sudo apt install cppcheck 安装，使用时仅需使用如下命令即可

```
cppcheck --enable=warning,performance sudoku.cpp
```

最终测试结果如下图所示：

```
(py37) root@dl-230605163731911-pod-jupyter-67cf765dc6-t4gsb:~/cppcheck# cppcheck --enable=warning,performance sudoku.cpp
Checking sudoku.cpp ...
(py37) root@dl-230605163731911-pod-jupyter-67cf765dc6-t4gsb:~/cppcheck#
```

图 3.1: Caption

没有额外的输出，说明已经消除了所有的警告。

4 代码覆盖率

4.1 工具介绍

使用的工具是 gcov + lcov 。

使用 gcov 检测代码覆盖率，在使用 gcc 编译 cpp 程序的时候要加上 -fprofile-arcs -ftest-coverage 两个参数：

```
g++ -fprofile-arcs -ftest-coverage sudoku.cpp -o test
```

然后测试程序，这时 gcov 会记录程序测试过程中代码的执行情况，然后使用如下命令生成测试结果：

```
gcov sudoku.cpp
```

这时就可以直接查看测试结果了，但是为了直观和美观，可以使用 lcov 来对结果统计，次序执行如下命令，即可生成 html 文件：

```
lcov -c -o collector.info -d .
genhtml collector.info -o collector_result
```

在测试的过程中，为了首先快速得到测试结果并观察每次测试间结果的变化，往往会多次测试，每次进行一部分，使用如下命令将测试结果合并：

```
lcov -a a.info -a b.info -o all.info
```

再使用 genhtml 生成 html 即可，最终测试结果如下图：

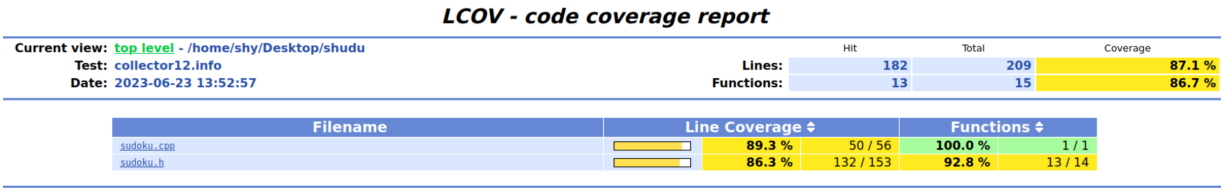

图 4.2: 测试结果

从测试结果中可以看出，代码覆盖率还是比较高的，其中 sudoku.h 中的 printsudoku 用于调试使用，在最终的程序中没有使用到，故函数测试结果为 13/14，其余函数全部覆盖。