数独解题与生成器测试文档

## 编写目的

本测试文档用于验证程序运行的正确性，检测程序性能。同时，指出软件下一步的改进方向。

## 背景

1. 待开发软件名称：数独解题与生成器。
2. 本项目的任务提出者：汤世平。
3. 本项目的设计人员：周赫斌。
4. 本项目的开发人员：周赫斌
5. 本项目的用户：未定。

## 测试环境

测试基于Windows 10 x64系统，编译器为Visual Studio 2017。

## 软件目标

1. 功能需求

软件需要实现的功能分为两部分。

当命令行传入参数为-c，后加一个数字n，则生成n个互不相同的数独解，输出到同目录的sudoku.txt中。

当命令行传入参数为-s，后加一个文件d（同时支持绝对路径与相对路径），则从d中读取若干个未完成的数独，分别求出它们的任意一个可行解，输出到同目录的sudoku.txt中。

异常处理：遇到以下行为之一，应当抛出错误，并退出程序。

* sudoku -c 后无输入或不是合法的数字输入。错误编号：233。
* sudoku -s 后无输入或不是合法的文件输入。错误编号：2333。
* sudoku -s d 在文件d中有一个无解的待求解数独。错误编号：666。
* sudoku -s d 在文件d中没有以一个完整的数独输入结尾，即不合法输入。错误编号：23333。

1. 性能需求

测试数据满足：

* 当输入为sudoku -c n 时，保证。
* 当输入为sudoku -s d 时，保证文件d中的待求解数独数不超过。

时间限制：一分钟。空间限制：512MB。

## 测试说明

测试针对二进制可执行文件bin/sudoku.exe。

由于单元测试不好对整个可执行文件进行测试，因此部分测试用例将使用cmd进行测试。

在测试中，可能用到的其他文件有:

test/assert\_1.exe 用于检测-c操作生成的是不是互不相同的数独。

test/assert\_2.exe 用于检测-c和-s操作生成的是不是合法数独。

test/gen.exe 用于生成100000个数独残局，测试-s操作的运行速度。

test/testcase\_1.txt 一个合法的输入，用于检测-s操作生成的是不是合法数独。

test/testcase\_2.txt 一个非法的输入，用于检测-s操作的非法判定。

test/testcase\_3.txt 一个非法的输入，用于检测-s操作的非法判定。

为简单起见，把这些不同文件夹的程序放在同一文件夹下测试。

## 测试用例

1. 参数检测

* 执行 “sudoku -c”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : no input number.”

* 执行 “sudoku -c 0”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : not a valid number.”

* 执行 “sudoku -c 1000001”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : not a valid number.”

* 执行 “sudoku -c0”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error ：no valid command arguments.”

* 执行 “sudoku -C 123”

预期结果：程序正常执行。

实际结果：在cmd下输出 “time = 0 ms”

* 执行 “sudoku -c -123”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : not a valid number.”

* 执行 “sudoku -c abc

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : not a valid number.”

* 执行 “sudoku -c 1000000”

预期结果：程序正常执行。

实际结果：在cmd下输出 “time = 1524 ms”

* 执行 “sudoku -s”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : no input directory.”

* 执行 “sudoku -s .”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : fail to open the file.”

* 执行 “sudoku -s ../”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : fail to open the file.”

* 执行 “sudoku -s ./”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : fail to open the file.”

* 执行 “sudoku -s c:\”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : fail to open the file.”

* 执行 “sudoku -ssudoku.in”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : no valid command arguments.”

* 执行 “sudoku -S sudoku.in”

预期结果：程序正常执行。

实际结果：在cmd下输出 “time = 8100 ms”

* 执行 “sudoku -S testcase.txt”

预期结果：程序正常执行。

实际结果：在cmd下输出 “time = 3 ms”

* 执行 “sudoku aaa”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : no valid command arguments.”

* 执行 “sudoku”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : No args found”

1. 功能检测

* 执行“sudoku -c 1000000”, 并执行“assert\_1 sudoku.txt”

预期结果：assert\_1不报错。

实际结果：不报错。 “time = 1366 ms”

* 执行“sudoku -c 1000000”, 并执行“assert\_2 sudoku.txt”

预期结果：assert\_2不报错。

实际结果：不报错。 “time = 1520 ms”

* 执行“gen”, “sudoku -s sudoku.in”, 并执行“assert\_2 sudoku.txt”

预期结果：assert\_2不报错。

实际结果：不报错。 “time = 7396 ms”

* 执行“sudoku -s testcase\_1.txt”, 并执行“assert\_2 sudoku.txt”

预期结果：assert\_2不报错。

实际结果：不报错。 “time = 1 ms”

* 执行“sudoku -s testcase\_2.txt”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : not valid input.”

* 执行“sudoku -s testcase\_2.txt”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : not valid input.”

* 执行“sudoku -s testcase\_3.txt”

预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : not valid input.”

* 执行“sudoku -s testcase\_4.txt”

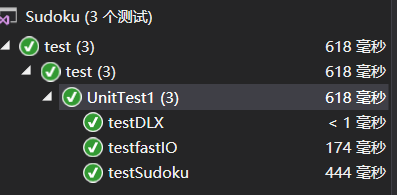
预期结果：程序异常退出。

实际结果：在cmd下输出 “Error : no solutions found.”

1. 单元测试

单元测试选择了3个模块进行测试，即输入输出模块(namespace fastIO)，DLX模块(class DLX)，数独实现模块(class Sudoku)。

测试结果：



具体的单元测试代码，详见test/unittest1.cpp。

## 测试总结

在功能需求方面，程序通过了所有测试用例。

在性能需求方面，程序输出的解的正确性得到验证，但是有一处地方有所不足。执行100000级别的-s操作时，用时达到了7.3s左右。如果测试1000000级别的数据达不到60s内出解的要求，但各方面优化无法再提升了。下一步如果可能，应当尝试继续优化-s操作的执行速度。