

数独程序-用例测试报告

小组成员：牟迪 宋彦艳

学号：2011360 2013471

班级：徐思涵老师班

专业：计算机科学与技术

本次实验中，完成了一个能够生成数独游戏，并求解数独问题的控制台程序。在对原始程序进行质量分析之后，得到了符合代码规范的程序代码。对于完成的控制台应用程序，为其编写了相关的测试样例以测试其各项功能的正常。并统计所有测试下总的测试覆盖率。

一.用例测试

本次实验中数独程序的主要功能分为，产生数独终盘，产生数独初盘，求解数独初盘。其中产生数独初盘又分为若干情况。根据实验要求中说明的参数需求，程序需要支持如下参数：

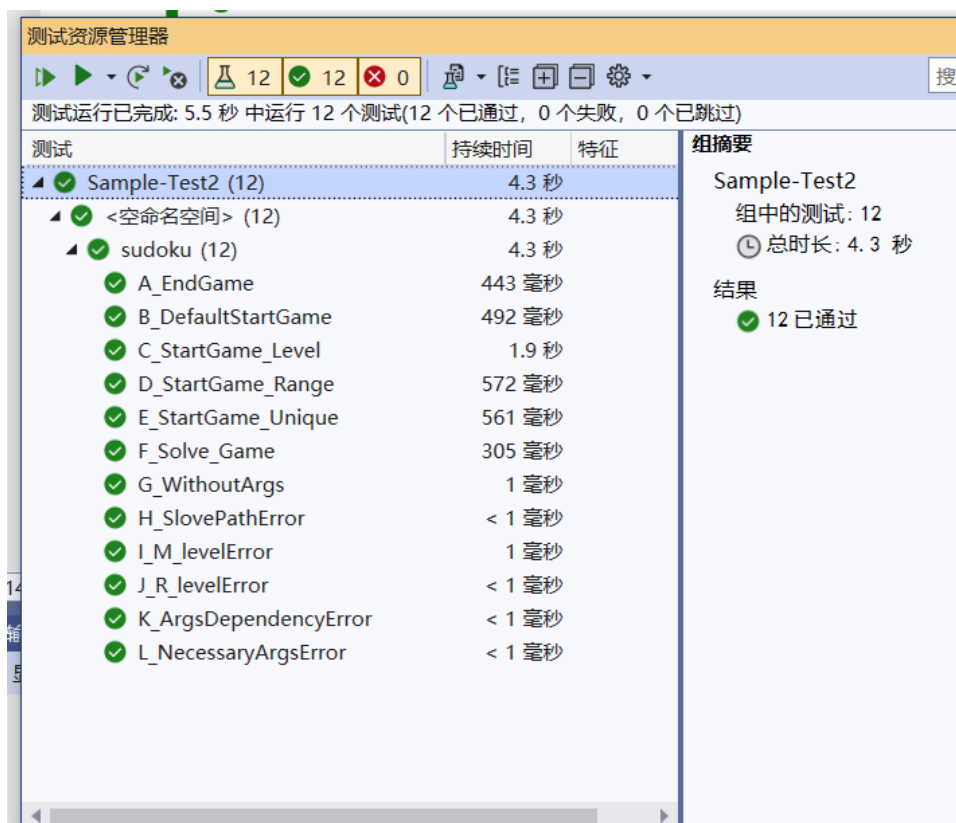
参 数 名 字	参数含义	范围限制	用法示例
-c	需要的数独终盘的数量	1-10000000	sudoku.exe -c 20 [表示生成20个数独终盘]
-s	需要求解的数独初盘路径	绝对或相对路径	sudoku.exe -s game.txt [求解相对路径下game.txt文件中的数独]
-n	需要的游戏数量	1-10000	sudoku.exe -n 1000 [生成1000个数独游戏]
-m	生成游戏的难度	1-3	sudoku.exe -n 1000 -m 1[生成1000个简单数独游戏，m必须和n一起使用]
-r	生成游戏中挖空的数量范围	20-55	sudoku.exe -n 1000 -r 20~55[生成1000个挖空数在20到55之间的数独游戏，r必须和n一起使用]
-u	生成游戏的解唯一		sudoku.exe -n 1000 -u [生成20个解唯一的数独游戏，u必须和n一起使用]

针对目标功能需求以及必要的异常处理情况，本组成员编写了如下的测试用例：

用 例 编 号	测试用例	用例说明	期望结果
A	sudoku.exe -c 1000	生成1000个数独终盘	产生final.txt文件，其中内容通过终盘测试

用例编号	测试用例	用例说明	期望结果
B	sudoku.exe -n 1000	生成1000个默认の数独初盘	产生game.txt文件，其中内容通过初盘测试
C	sudoku.exe -n 1000 -m 1	生成1000个1阶の数独初盘，挖空数27	产生game.txt文件，每个数独中留空27处
	sudoku.exe -n 1000 -m 2	生成1000个2阶の数独初盘，挖空数36	产生game.txt文件，每个数独中留空36处
	sudoku.exe -n 1000 -m 3	生成1000个3阶の数独初盘，挖空数45	产生game.txt文件，每个数独中留空45处
D	sudoku.exe -n 1000 -r 25~40	生成1000个数独初盘，挖空数在25至40之间	产生game.txt文件，每个数独中留空数在25至40之间
E	sudoku.exe -n 1000 -u	生成1000个唯一解数独初盘	产生game.txt文件，通过唯一初盘测试
F	sudoku.exe -s game.txt	求解相对目录下game.txt文件中的数独初盘	产生sudoku.txt文件，通过终盘测试
G	sudoku.exe -n	给定的参数数量较少	提示错误信息
H	sudoku.exe -s something.txt	求解的文件不存在	提示错误信息
I	sudoku.exe -n 1000 -m 5	-m参数的取值超出范围	提示错误信息
J	sudoku.exe -n 1000 -r 10~60	-r参数的取值超出范围	提示错误信息
K	sudoku.exe -m 1	-m参数缺少-n的依赖	提示错误信息
	sudoku.exe -r 20~40	-r参数缺少-n的依赖	提示错误信息
	sudoku.exe -u	-u参数缺少-n的依赖	提示错误信息
L	sudoku.exe -a 1 -b 2	缺少必要的参数项	提示错误信息

实验中采用的开发环境为VisualStudio2022 社区版，用例测试采用集成进VisualStudio的GoogleTest平台。借助其中的断言宏与测试夹具来编写清晰灵活的用例测试。在进行测试过程中，对于原始程序的主函数进行了函数导出操作，以便于进行测试。具体的GoogleTest测试项目可以在[github](#)下进行获取。用例测试的结果如下图所示：



由上图结果可得，我们的程序用例测试的通过率达到100%，总计用时5.5s，其中构建用时1.2s，测试用时4.3s

二.覆盖率统计

在保证了程序的用例通过率100%下，我们还统计了程序在执行测试用例过程中的代码覆盖率，以确保我们测试用例编写的合理性和准确性。代码覆盖率的统计采用GNU工具链中的gcov工具，在通过gcc对程序进行编译时，附加编译选项来对程序进行插桩。从而在程序执行时获取各段代码的执行情况。据此，我们重新执行了全部的测试样例，得到了覆盖率统计文件sudoku-sudoku.gcda。为了便于进行覆盖率的展示。采用Linux环境下的lcov工具，对覆盖率文件进行可视化分析。得到如下的统计结果：

LCOV - code coverage report					
Current view: top level		Hit		Total	Coverage
Test: coverage.info		Lines:	797	862	92.5 %
Date: 2023-06-29 22:31:39		Functions:	264	283	93.3 %
Directory	Line Coverage			Functions	
/home/bia/PairProgramming	100.0 %	302 / 302	100.0 %	31 / 31	
ll	89.7 %	35 / 39	96.6 %	28 / 29	
ll/bits	89.0 %	430 / 483	92.3 %	180 / 195	
ll/ext	78.9 %	30 / 38	89.3 %	25 / 28	

Generated by: LCOV version 1.14

根据代码覆盖率的分析情况，我们的测试用例代码覆盖率为100%，覆盖了全部的代码与全部的函数。借助lcov工具的分析结果，查看各行代码的执行次数与各个函数的执行次数：

```

528 :
529 16 : int main(int argc, char** argv) {
530 16 :     ParameterList curr_para;
531 16 :     if (argc <= 2) {
532 2 :         cout << "please input some args...";
533 2 :         curr_para.clear();
534 2 :         clearState();
535 2 :         return 0;
536 :     }
537 :
538 14 :     int s_condition = parameterAnalysis_s(argc, argv);
539 14 :     if (s_condition>=0)
540 :     {
541 2 :         if (s_condition == 1)
542 :         {
543 :             //需要求解的数独不存在
544 1 :             curr_para.clear();
545 1 :             clearState();
546 1 :             return 0;
547 :         }
548 1001 :         while (!readsudoku()) { //每次读入一个棋盘，返回1代表文件读完了
549 1000 :             do_solve(curr_para.u); //对当前棋盘进行求解
550 :         }
551 1 :         curr_para.clear();
552 1 :         clearState();
553 1 :         return 0;
554 :     }

```

LCOV - code coverage report

Current view: top level - home/kid/PairProgramming - x.cpp (source / functions)			
Test: coverage.info	Lines:	Hit: 302	Total: 302
Date: 2023-06-29 22:31:39	Functions:	Hit: 31	Total: 31
Coverage: 100.0 %			
Coverage: 100.0 %			
Function Name ↕	Hit count		
_Z3semiiiiii	7		
_ZNK13ParameterList@paraPrintEv	7		
_Z20parameterAnalysis_cnR13ParameterListiPPc	12		
_Z21parameterAnalysis_mwuR13ParameterListiPPc	12		
_Z19parameterAnalysis_slPPc	14		
_Z10clearStatev	16		
_ZN11FileOperateC2Ev	16		
_ZN13GlobalVariableC2Ev	16		
_ZN13ParameterList5clearEv	16		
main	16		
_Z10readsudokuv	1001		
_Z8do_solvei	2001		
_ZN11array_stackC2Ev	2001		
_Z11solveUniquesiiii	6000		
_Z3dici	6001		
_Z7test_leviii	6001		
_Z12shuffleArrayv	7000		
_Z8init_semiiiiii	7000		
_Z7porttoflv	8000		
_Z7row_exdii	14000		

完成的覆盖率分析文件coverage_html可以在[github](#)下与作业附件中进行获取