

后卫小金 – project 2

2021.08.20

一、简要说明

后卫小金-project 2.zip 中主要包含三个文件：

1、project2 文件夹

代码 sudoku.cpp 所在文件夹。

sudoku.cpp 对于作业中的所有情况都进行了测试，均能在 1s 内给出答案。

代码主要包含的函数及主要功能如下：

- `void print_sudoku(std::vector<std::vector<int>>> sudoku)`
打印数独
- `void read_sudoku(std::vector<std::vector<int>>& sudoku, std::string& txt)`
读取数独
- `void flip(int i, int j, int digit)`
对记录数独状态的二进制数组进行反转
- `bool cmp(std::pair<int, std::pair<int, int>>& p1, std::pair<int, std::pair<int, int>>& p2)`
比较函数，用于对数独空位处所能填写的数字数量进行排序
- `void set_spaces(std::vector<std::vector<int>>& sudoku)`
记录数独在当前状态下空数字的位置以及所能填写的数字的数量
- `void dfs(std::vector<std::vector<int>>& sudoku, int pos)`
深度优先搜索算法解数独
- `void solve_sudoku(std::vector<std::vector<int>>& sudoku)`
解数独的主要函数，首先设置状态记录数组，然后先对只能填写一个数字的位置进行填写，然后对数独的空位进行记录并按照潜在数字数量进行排序，最后调用深度优先搜索算法
- `int main()`
主函数

详细注释请参考 sudoku.cpp。

注 1：路径相关请检测 sudoku.cpp 的 170 行和 read_sudoku 函数。

注 2：sudoku_1 中没有使用按位优化，sudoku_2 中使用了 bitset，其它部分相同，所以在此不做详细介绍。

2、sudoku_data 文件夹

测试数据所在文件夹。

3、说明文档

包含代码的简要说明与测试结果。

二、测试结果

sudoku.cpp 在 visual studio 2019 release 模式下的测试结果如下：

1、easy.txt

```
solution_count = 1
PrintDuration : duration_mili duration = 1 ms
the only solution is:
4 5 3 6 7 2 9 8 1
8 2 6 1 9 3 7 4 5
9 1 7 5 8 4 2 6 3
1 7 8 5 4 9 6 3 2
5 6 4 3 2 7 1 9 8
3 8 9 8 1 6 4 5 7
6 8 2 7 3 4 5 1 9
7 4 1 9 5 8 3 2 6
3 9 5 2 6 1 8 7 4
```

2、medium.txt

```
solution_count = 225
PrintDuration : duration_mili duration = 1 ms
```

3、hard.txt

```
solution_count = 6592
PrintDuration : duration_mili duration = 50 ms
```

4、hard+.txt

```
solution_count = 15540
PrintDuration : duration_mili duration = 42 ms
```

5、hard++.txt

```
solution_count = 31322
PrintDuration : duration_mili duration = 132 ms
```

6、总结

	easy	medium	hard	hard+	hard++
解数	1	225	6592	15540	31322
时间	1ms	1ms	50ms	42ms	132ms

三、优化思路

sudoku.cpp 中主要使用了三种优化思路。

1、位运算优化

例：数字的二进制表示的第 i 位为 1，表示数字 $i+1$ 已经出现过。例如二进制表示为 $(011000100)_2$ 时，就表示数字 3，7，8 已经出现过。

2、记录空位处潜在数字的数量

记录空位处潜在的所能填写的数字的数量，从低到高进行排序。先填写潜在数量少的位置，再填写潜在数量多的位置，以减少迭代的次数。

3、先填写只能填写一个数字的位置

先将只能填写 1 个数字的空位填写完成，具体为：先填写一个数字，之后更新数独数组与状态数组，若出现只能填写一个数字的空位则填写，之后循环这个过程，直到没有只能填写一个数字的位置出现。

四、参考

代码参考主要来自以下三个方面：

1、Leetcode

作者：LeetCode-Solution

链接：<https://leetcode-cn.com/problems/sudoku-solver/solution/jie-shu-du-by-leetcode-solution/>

2、Leetcode

作者：ikaruga

链接：<https://leetcode-cn.com/problems/sudoku-solver/solution/37-by-ikaruga/>

3、助教的思路提示