

C++基础与深度解析

各位同学们，大家好，你们期盼已久的第二个项目终于来了，提前祝愿大家能顺利通过挑战~

项目 2：数独

数独（sudoku）是源自 18 世纪瑞士的一种数学游戏。玩家需要根据 9×9 盘面上的已知数字，推理出所有剩余空格的数字，并满足每一行、每一列、每一个粗线宫内的数字均含 1-9，且不能重复。

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

使用搜索求解数独问题，是算法问题经典中的经典，希望同学们可以编写 C++ 程序来解决问题，巩固所学。一般来说，数独谜题答案是唯一的，但我们的程序完全可以处理多解的情况。

具体要求：

- 1.分别读入测试数据 easy.txt 和 medium.txt，每组数据共 9 行 9 列，若数字为 0 则代表空格；
- 2.输出搜索的结果
 - 2.1.对 easy 数据，输出数独的解，即一幅填好的数独；
 - 2.2.对 medium 数据，输出解数；
- 3.输出运行时间，可以参考附件中的代码框架；
- 4.在说明文档中展示搜索的结果、运行时间和剪枝的思路。

扩展：

尝试求解 hard, hard+, hard++这几份数据的解数。由于不同环境下程序运行时间会有差异，我们作业的评判将重点关注算法实现过程（比如剪枝的思路），当然，最终目标是希望 hard++能够在不开启编译优化时还能秒出结果，希望大家朝这个方向努力。下面这份表格供大家参考。

难度	easy	medium	hard	hard+	hard++
解数	1	225	6592	15540	31322

提示：

- 1.数独的状态不方便存储，因此建议使用深度优先搜索（DFS）；
- 2.剪枝的小技巧有很多。比如，使用位运算来检查行/列/宫格的状态；也可以结合启发式搜索的思想，看看怎样才能够让后续搜索的状态减少；
- 3.舞蹈链算法（Dancing Links）对数独问题的求解有奇效。