**实验结果说明**

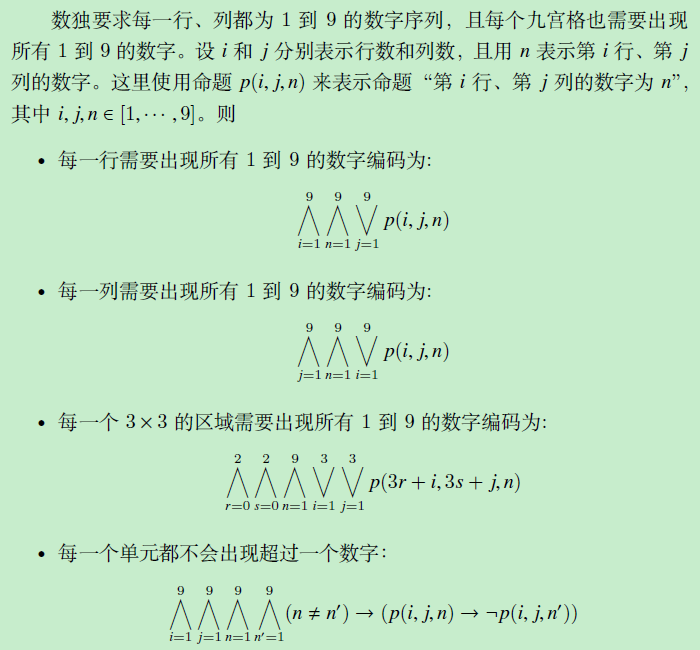
操作系统：Ubuntu 20.04.3 LTS

处理器：Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz

内存：64GB

SAT求解器：MiniSAT

**算法思想：**



**工程运行方式：**

进入到代码根目录下，修改Global.cpp文件中的sudokupath(数独问题文件存放路径)，resultpath（最后解得的数独文件存放路径），sudokucnf(MiniSAT输入文件)，resultcnf（MiniSAT输出文件）,设置数独的大小（row,col）。

然后运行make clean,make生成main可执行文件，./main运行可执行文件。

**代码解读：**

sudoku.cpp文件中的ToCnf()函数读取数独，然后转化成MiniSAT能读取的CNF格式。每个p(i,j,n)表示第i行，第j列，填入数字n，p(i,j,n)是第i\*(row\*row)+j\*(row)+n个变量，-p(i,j,n)表示┒p(i,j,n)。

**求解结果：**

Problem\_1.txt

0 0 0 1 0 0 3 2 0

5 0 0 0 9 0 4 0 6

2 6 0 0 7 3 0 0 0

0 0 9 0 3 0 0 5 0

0 0 0 0 0 0 0 0 0

0 2 0 0 4 0 9 0 0

0 0 0 3 1 0 0 4 9

8 0 5 0 6 0 0 0 3

0 9 3 0 0 5 0 0 0

Result\_1.txt

9 4 8 1 5 6 3 2 7

5 3 7 8 9 2 4 1 6

2 6 1 4 7 3 8 9 5

7 8 9 2 3 1 6 5 4

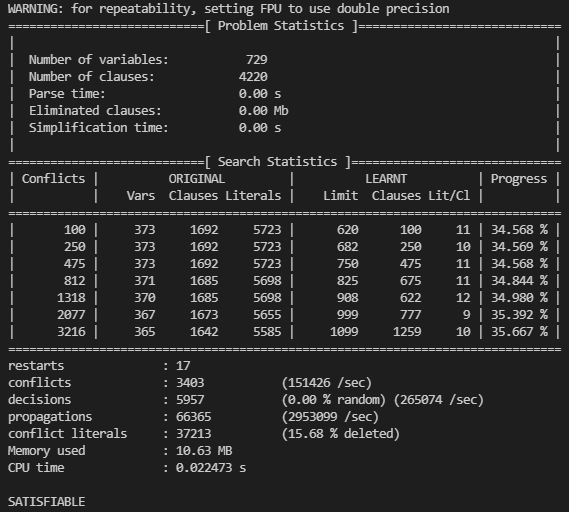
1 5 4 6 8 9 7 3 2

3 2 6 5 4 7 9 8 1

6 7 2 3 1 8 5 4 9

8 1 5 9 6 4 2 7 3

4 9 3 7 2 5 1 6 8



Problem\_2.txt

0 0 0 0 7 0 0 0 0

6 4 0 0 0 3 5 0 0

1 0 0 4 9 0 0 6 0

0 0 6 8 0 0 0 0 0

0 2 0 3 4 1 0 5 0

0 0 0 0 0 9 2 0 0

0 8 0 0 3 4 0 0 6

0 0 2 9 0 0 0 8 3

0 0 0 0 1 0 0 0 0

Result\_2.txt

2 9 8 5 7 6 3 1 4

6 4 7 1 8 3 5 9 2

1 3 5 4 9 2 8 6 7

3 1 6 8 2 5 7 4 9

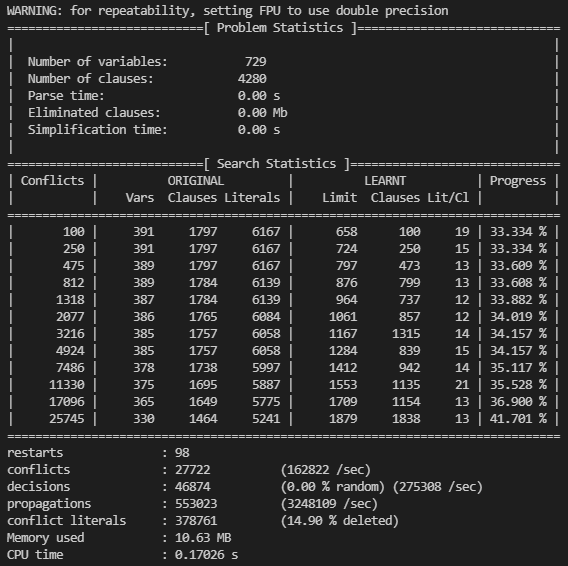
7 2 9 3 4 1 6 5 8

8 5 4 7 6 9 2 3 1

5 8 1 2 3 4 9 7 6

4 6 2 9 5 7 1 8 3

9 7 3 6 1 8 4 2 5



Problem\_3.txt

3 0 0 0 6 0 0 2 5

8 0 0 0 0 0 0 0 0

0 0 0 2 0 5 0 0 0

0 0 5 0 1 0 9 0 0

6 0 0 3 9 7 0 0 8

0 0 3 0 5 0 1 0 0

0 0 0 1 0 4 0 0 0

0 0 0 0 0 0 0 0 9

4 1 0 0 7 0 0 0 6

Result\_3.txt

3 9 4 8 6 1 7 2 5

8 5 2 7 4 9 6 3 1

1 7 6 2 3 5 8 9 4

7 8 5 4 1 2 9 6 3

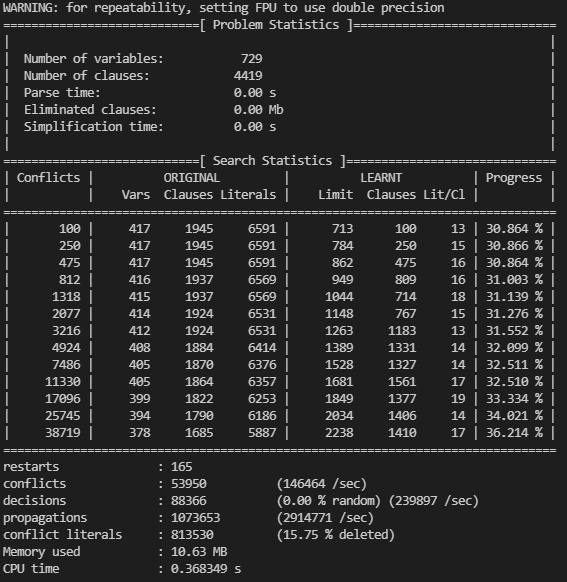
6 2 1 3 9 7 5 4 8

9 4 3 6 5 8 1 7 2

5 6 9 1 2 4 3 8 7

2 3 7 5 8 6 4 1 9

4 1 8 9 7 3 2 5 6



Problem\_4.txt

0 0 1 0 0 0 0 3 14 0 2 0 0 15 7 0

0 10 13 0 1 0 0 4 16 0 0 7 6 12 0 5

12 14 0 11 0 6 0 0 0 0 0 0 13 2 0 3

0 7 0 0 12 11 0 16 15 0 0 13 1 0 4 0

0 6 8 0 5 12 16 15 0 3 11 10 7 0 9 4

3 0 12 0 11 0 0 0 0 2 13 15 0 0 6 0

0 15 7 16 9 0 10 0 6 0 0 5 3 1 0 2

0 0 0 1 6 0 0 2 0 9 4 0 0 0 0 0

5 2 15 0 0 0 0 10 0 0 0 12 0 11 3 1

14 3 0 4 0 2 0 0 11 13 0 6 9 0 0 0

8 13 0 0 0 0 11 0 0 1 5 0 0 4 10 0

1 0 9 0 14 0 0 12 0 16 0 0 5 0 0 8

11 0 0 0 10 0 0 0 0 8 7 1 15 16 0 12

7 12 2 0 0 0 0 8 5 0 0 11 0 0 1 6

16 0 0 0 0 1 0 11 13 0 0 9 4 3 0 0

9 1 0 0 15 0 3 0 0 4 16 0 8 0 0 11

Result\_4.txt

6 16 1 8 13 10 5 3 14 12 2 4 11 15 7 9

15 10 13 9 1 14 2 4 16 11 3 7 6 12 8 5

12 14 4 11 7 6 15 9 10 5 1 8 13 2 16 3

2 7 3 5 12 11 8 16 15 6 9 13 1 10 4 14

13 6 8 2 5 12 16 15 1 3 11 10 7 14 9 4

3 9 12 14 11 4 7 1 8 2 13 15 16 5 6 10

4 15 7 16 9 8 10 13 6 14 12 5 3 1 11 2

10 5 11 1 6 3 14 2 7 9 4 16 12 8 15 13

5 2 15 6 16 13 9 10 4 7 8 12 14 11 3 1

14 3 10 4 8 2 1 5 11 13 15 6 9 7 12 16

8 13 16 12 3 7 11 6 9 1 5 14 2 4 10 15

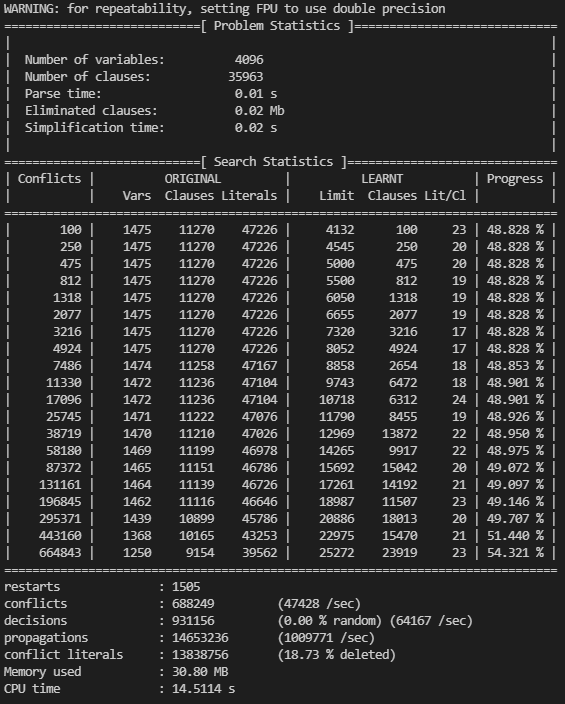
1 11 9 7 14 15 4 12 2 16 10 3 5 6 13 8

11 4 5 13 10 9 6 14 3 8 7 1 15 16 2 12

7 12 2 3 4 16 13 8 5 15 14 11 10 9 1 6

16 8 14 15 2 1 12 11 13 10 6 9 4 3 5 7

9 1 6 10 15 5 3 7 12 4 16 2 8 13 14 11



Problem\_5.txt

0 16 1 8 0 0 5 3 14 12 2 4 0 15 7 0

15 0 0 0 1 14 2 4 16 11 3 0 6 12 8 0

12 14 0 0 0 0 0 9 0 0 1 8 0 2 0 0

0 7 3 0 12 0 8 16 0 6 0 0 0 10 0 14

0 0 8 0 0 12 16 15 0 3 0 0 7 14 0 4

3 9 12 14 0 4 7 1 0 2 13 0 0 5 6 10

0 15 7 16 9 8 10 13 6 14 0 0 0 1 11 2

10 5 11 1 6 3 14 2 7 9 4 16 0 8 15 13

0 0 15 0 16 13 0 10 4 0 8 0 14 0 3 1

0 3 0 4 0 0 1 0 0 13 15 0 9 0 12 16

0 0 0 0 3 7 11 6 9 1 0 14 0 4 10 0

1 0 9 0 0 0 0 12 0 0 10 0 0 0 13 8

0 0 0 13 10 9 6 0 3 0 7 1 15 16 0 12

7 0 0 3 4 0 0 8 5 15 14 11 0 9 1 6

0 8 0 15 2 1 0 0 13 10 6 0 4 0 5 7

9 1 6 10 15 0 3 7 12 0 0 0 0 0 14 0

Result\_5.txt

6 16 1 8 13 10 5 3 14 12 2 4 11 15 7 9

15 10 13 9 1 14 2 4 16 11 3 7 6 12 8 5

12 14 4 11 7 6 15 9 10 5 1 8 13 2 16 3

2 7 3 5 12 11 8 16 15 6 9 13 1 10 4 14

13 2 8 6 5 12 16 15 1 3 11 10 7 14 9 4

3 9 12 14 11 4 7 1 8 2 13 15 16 5 6 10

4 15 7 16 9 8 10 13 6 14 12 5 3 1 11 2

10 5 11 1 6 3 14 2 7 9 4 16 12 8 15 13

5 11 15 2 16 13 9 10 4 7 8 12 14 6 3 1

14 3 10 4 8 2 1 5 11 13 15 6 9 7 12 16

8 13 16 12 3 7 11 6 9 1 5 14 2 4 10 15

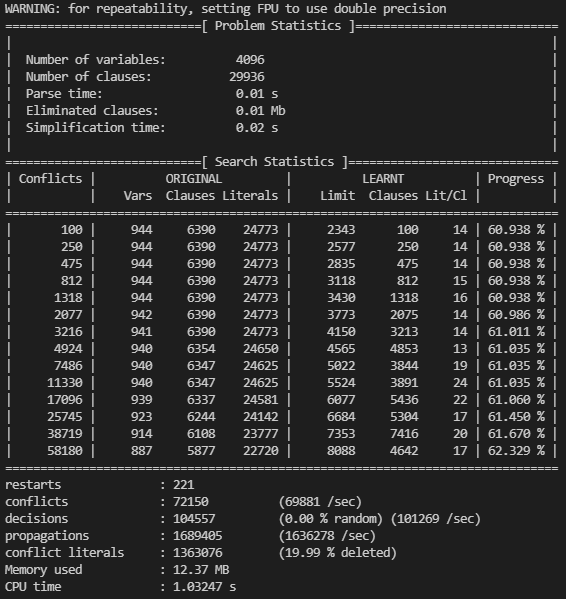
1 6 9 7 14 15 4 12 2 16 10 3 5 11 13 8

11 4 5 13 10 9 6 14 3 8 7 1 15 16 2 12

7 12 2 3 4 16 13 8 5 15 14 11 10 9 1 6

16 8 14 15 2 1 12 11 13 10 6 9 4 3 5 7

9 1 6 10 15 5 3 7 12 4 16 2 8 13 14 11



Problem\_6.txt

0 16 0 8 0 10 0 0 0 0 2 0 11 0 0 0

15 10 13 0 0 0 0 4 16 11 0 7 0 0 0 5

0 14 0 11 0 6 15 9 0 5 0 0 0 2 16 0

2 7 0 5 12 0 8 16 15 0 9 0 1 0 4 14

0 0 0 2 5 12 0 0 1 0 0 0 0 14 9 0

3 0 0 14 11 4 7 1 0 0 0 15 16 5 6 10

4 0 0 0 9 8 0 0 6 0 0 0 3 1 11 0

10 0 11 1 6 0 0 2 0 0 4 0 12 8 0 13

0 0 0 6 0 0 9 0 4 7 8 0 0 0 0 1

14 3 0 0 8 0 1 5 11 13 0 0 0 0 0 0

0 13 0 12 0 0 0 0 0 1 5 0 2 4 0 15

1 11 0 0 0 0 0 12 0 16 10 3 5 0 13 8

0 4 0 0 0 9 6 14 3 0 7 0 0 16 2 0

0 12 2 0 0 16 0 8 0 15 0 11 0 9 1 0

0 0 14 0 2 1 0 11 0 10 0 9 0 0 0 0

9 1 0 10 0 0 0 0 0 0 0 2 0 13 0 11

Result\_6.txt

6 16 1 8 3 10 5 13 14 12 2 4 11 15 7 9

15 10 13 9 1 14 2 4 16 11 3 7 6 12 8 5

12 14 4 11 7 6 15 9 10 5 1 8 13 2 16 3

2 7 3 5 12 11 8 16 15 6 9 13 1 10 4 14

16 6 8 2 5 12 13 15 1 3 11 10 4 14 9 7

3 9 12 14 11 4 7 1 8 2 13 15 16 5 6 10

4 15 7 13 9 8 16 10 6 14 12 5 3 1 11 2

10 5 11 1 6 3 14 2 7 9 4 16 12 8 15 13

5 2 15 6 16 13 9 3 4 7 8 12 14 11 10 1

14 3 10 4 8 2 1 5 11 13 15 6 9 7 12 16

8 13 16 12 10 7 11 6 9 1 5 14 2 4 3 15

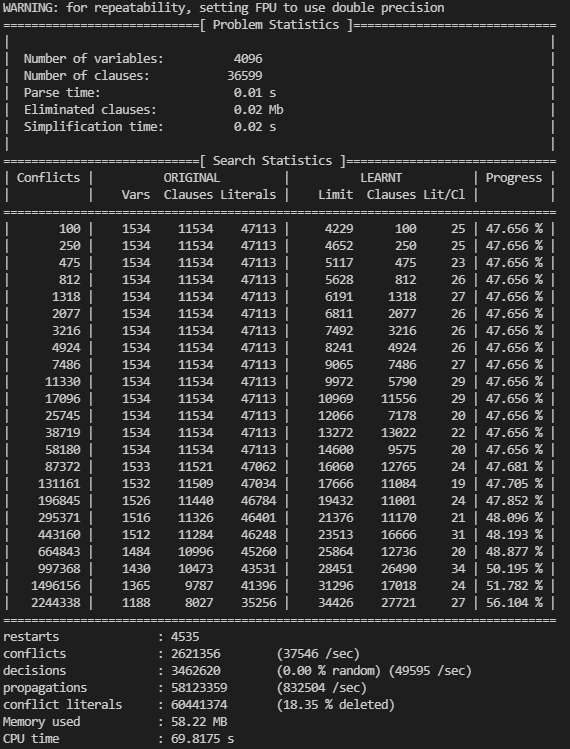
1 11 9 7 14 15 4 12 2 16 10 3 5 6 13 8

11 4 5 15 13 9 6 14 3 8 7 1 10 16 2 12

13 12 2 3 4 16 10 8 5 15 14 11 7 9 1 6

7 8 14 16 2 1 12 11 13 10 6 9 15 3 5 4

9 1 6 10 15 5 3 7 12 4 16 2 8 13 14 11



MiniSAT求解数独问题的时间随着数独规模的扩大呈现一个递增的态势，数独规模越大，需要的变量越多，9x9的数独是729个变量，16x16的数独有4096个变量，9x9数独生成的子句一般是4000左右的量级，16x16的数独生成的子句是30000左右的量级，两种规模数独的求解时间差了100倍左右。