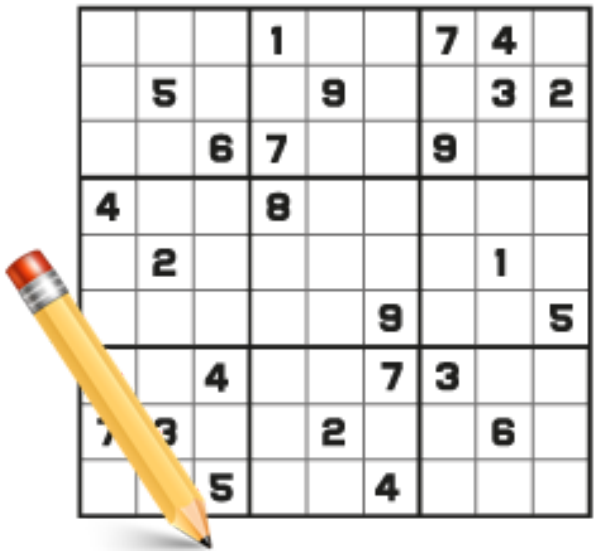

Project Report

CSC320 Summer 2018

Project - SAT Based Sudoku Solving

Team: Zijian Chen (V00867494), Ximing (Evan) Guo(V00866199), Shuwen Li(V00024025), Xiaole (Wendy) Tan(V00868734)

August 2018

Introduction

In this project, we will write two programs with a given SAT solver to generate a solution for a unsolved Sudoku puzzles.

Method

We decided to implement both programs using Java language. The first program, sud2sat read in a Sudoku puzzle and translate it to a CNF formula which allow us to input it into a miniSAT SAT solver. The miniSAT solver will output a solution file which the the second program, sat2sud takes in, decode and output the solved puzzle.

Experiment

In this experiment we test our program using 50 unsolved Sudoku puzzle, check the correctness and record time required to solve for each puzzle.

Result

50 unsolved Sudoku puzzle (input). The input is formatted as the following, each line contains 81 integers indicating one Sudoku puzzle.

003020600900305001001806400008102900700000008006708200002609500800203009005010300

200080300060070084030500209000105408000000000402706000301007040720040060004010003

000000907000420180000705026100904000050000040000507009920108000034059000507000000

030050040008010500460000012070502080000603000040109030250000098001020600080060020

020810740700003100090002805009040087400208003160030200302700060005600008076051090

100920000524010000000000070050008102000000000402700090060000000000030945000071006

043080250600000000000000109490000407000060800001020000382050000000000005034090710

480006902002008001900370060840010200003704100001060049020085007700900600609200018

000900002050123400030000160908000000070000090000000205091000050007439020400007000

001900003900700160030005007050000009004302600200000070600100030042007006500006800

000125400008400000420800000030000095060902010510000060000003049000007200001298000

062340750100005600570000040000094800400000006005830000030000091006400007059083260

300000000005009000200504000020000700160000058704310600000890100000067080000005437

630000000000500008005674000000020000003401020000000345000007004080300902947100080

000020040008035000000070602031046970200000000000501203049000730000000010800004000

361025900080960010400000057008000471000603000259000800740000005020018060005470329

050807020600010090702540006070020301504000908103080070900076205060090003080103040

080005000000003457000070809060400903007010500408007020901020000842300000000100080

003502900000040000106000305900251008070408030800763001308000104000020000005104800

000000000009805100051907420290401065000000000140508093026709580005103600000000000

020030090000907000900208005004806500607000208003102900800605007000309000030020050

005000006070009020000500107804150000000803000000092805907006000030400010200000600

040000050001943600009000300600050002103000506800020007005000200002436700030000040

004000000000030002390700080400009001209801307600200008010008053900040000000000800

360020089000361000000000000803000602400603007607000108000000000004180009700300 14

500400060009000800640020000000001008208000501700500000000090084003000600060003002

007256400400000005010030060000508000008060200000107000030070090200000004006312700

000000000079050180800000007007306800450708096003502700700000005016030420000000000

030000080009000500007509200700105008020090030900402001004207100002000800070000090

200170603050000100000006079000040700000801000009050000310400000005000060906037002

000000080800701040040020030374000900000030000005000321010060050050802006080000000

000000085000210009960080100500800016000000000890006007009070052300054000480000000

608070502050608070002000300500090006040302050800050003005000200010704090409060701

050010040107000602000905000208030501040070020901080406000401000304000709020060010

053000790009753400100000002090080010000907000080030070500000003007641200061000940

006080300049070250000405000600317004007000800100826009000702000075040190003090600

005080700700204005320000084060105040008000500070803010450000091600508007003010600

000900800128006400070800060800430007500000009600079008090000401000360028400100700 0

000080000270000054095000810009806400020403060006905100017000620460000038000090000

000602000400050001085010620038206710000000000019407350026040530900020007000809000

000900002050123400030000160908000000070000090000000205091000050007439020400007000

380000000000400785009020300060090000800302009000040070001070500495006000000000092

000158000002060800030000040027030510000000000046080790050000080004070100000325000

010500200900001000002008030500030007008000500600080004040100700000700006003004050

080000040000469000400000007005904600070608030008502100900000005000781000060000010

904200007010000000000706500000800090020904060040002000001607000000000030300005702

000700800006000031040002000024070000010030080000060290000800070860000500002006000

001007090590080001030000080000005800050060020004100000080000030100020079020700400

000003017015009008060000000100007000009000200000500004000000020500600340340200000

300200000000107000706030500070009080900020004010800050009040301000702000000008006

50 solved Sudoku puzzle with their solving time:

Solution	CPU Time
4 8 3 9 2 1 6 5 7 9 6 7 3 4 5 8 2 1 2 5 1 8 7 6 4 9 3 5 4 8 1 3 2 9 7 6 7 2 9 5 6 4 1 3 8 1 3 6 7 9 8 2 4 5 3 7 2 6 8 9 5 1 4 8 1 4 2 5 3 7 6 9 6 9 5 4 1 7 3 8 2	0s
2 4 5 9 8 1 3 7 6 1 6 9 2 7 3 5 8 4 8 3 7 5 6 4 2 1 9 9 7 6 1 2 5 4 3 8 5 1 3 4 9 8 6 2 7 4 8 2 7 3 6 9 5 1 3 9 1 6 5 7 8 4 2 7 2 8 3 4 9 1 6 5 6 5 4 8 1 2 7 9 3	0.004114 s
4 6 2 8 3 1 9 5 7 7 9 5 4 2 6 1 8 3 3 8 1 7 9 5 4 2 6 1 7 3 9 8 4 2 6 5 6 5 9 3 1 2 7 4 8 2 4 8 5 6 7 3 1 9 9 2 6 1 7 8 5 3 4 8 3 4 2 5 9 6 7 1 5 1 7 6 4 3 8 9 2	0.001328 s
1 3 7 2 5 6 8 4 9	0.005948 s

928 314 567 465 897 312 673 542 981 819 673 254 542 189 736 256 731 498 391 428 675 784 965 123	
523 816 749 784 593 126 691 472 835 239 145 687 457 268 913 168 937 254 342 789 561 915 624 378 876 351 492	0.003188 s
176 923 584 524 817 639 893 654 271 957 348 162 638 192 457 412 765 398 265 489 713 781 236 945 349 571 826	0.00557 s
143 986 257 679 425 381 285 731 694 962 354 178 357 618 942 418 279 563 821 567 439 796 143 825 534 892 716	0.005825 s
487 156 932 362 498 751 915 372 864 846 519 273 593 724 186 271 863 549 124 685 397 738 941 625 659 237 418	0.001164 s
814 976 532 659 123 478 732 854 169 948 265 317 275 341 896 163 798 245	0.005632 s

3 9 1 6 8 2 7 5 4 5 8 7 4 3 9 6 2 1 4 2 6 5 1 7 9 8 3	
7 6 1 9 2 8 4 5 3 9 2 5 7 4 3 1 6 8 4 3 8 6 1 5 9 2 7 3 5 7 4 6 1 2 8 9 8 9 4 3 7 2 6 1 5 2 1 6 5 8 9 3 7 4 6 8 9 1 5 4 7 3 2 1 4 2 8 3 7 5 9 6 5 7 3 2 9 6 8 4 1	0.002033 s
9 7 6 1 2 5 4 3 8 1 5 8 4 3 6 9 2 7 4 2 3 8 7 9 1 5 6 2 3 4 7 6 1 8 9 5 8 6 7 9 5 2 3 1 4 5 1 9 3 8 4 7 6 2 7 8 2 5 1 3 6 4 9 3 9 5 6 4 7 2 8 1 6 4 1 2 9 8 5 7 3	0.005895 s
9 6 2 3 4 1 7 5 8 1 4 8 9 7 5 6 2 3 5 7 3 2 6 8 1 4 9 3 2 1 6 9 4 8 7 5 4 8 7 5 1 2 9 3 6 6 9 5 8 3 7 4 1 2 8 3 4 7 2 6 5 9 1 2 1 6 4 5 9 3 8 7 7 5 9 1 8 3 2 6 4	0.00412 s
3 9 7 6 8 1 5 2 4 6 4 5 2 7 9 8 1 3 2 1 8 5 3 4 9 7 6 8 2 3 9 5 6 7 4 1 1 6 9 7 4 2 3 5 8 7 5 4 3 1 8 6 9 2 4 7 2 8 9 3 1 6 5 5 3 1 4 6 7 2 8 9 9 8 6 1 2 5 4 3 7	0.001802 s
6 3 9 2 1 8 4 5 7 4 7 1 5 3 9 2 6 8 8 2 5 6 7 4 1 3 9 5 6 4 8 2 3 7 9 1 7 9 3 4 5 1 8 2 6 2 1 8 7 9 6 3 4 5 3 5 2 9 8 7 6 1 4 1 8 6 3 4 5 9 7 2 9 4 7 1 6 2 5 8 3	0.004754 s
6 9 7 1 2 8 3 4 5	0.005914 s

4 2 8 6 3 5 1 9 7 3 1 5 4 7 9 6 8 2 5 3 1 2 4 6 9 7 8 2 8 6 3 9 7 4 5 1 9 7 4 5 8 1 2 6 3 1 4 9 8 5 2 7 3 6 7 5 2 9 6 3 8 1 4 8 6 3 7 1 4 5 2 9	
3 6 1 7 2 5 9 4 8 5 8 7 9 6 4 2 1 3 4 9 2 8 3 1 6 5 7 6 3 8 2 5 9 4 7 1 1 7 4 6 8 3 5 9 2 2 5 9 1 4 7 8 3 6 7 4 6 3 9 2 1 8 5 9 2 3 5 1 8 7 6 4 8 1 5 4 7 6 3 2 9	0.004056 s
3 5 9 8 6 7 1 2 4 6 4 8 3 1 2 5 9 7 7 1 2 5 4 9 8 3 6 8 7 6 9 2 4 3 5 1 5 2 4 7 3 1 9 6 8 1 9 3 6 8 5 4 7 2 9 3 1 4 7 6 2 8 5 4 6 5 2 9 8 7 1 3 2 8 7 1 5 3 6 4 9	0.001307 s
7 8 6 9 4 5 3 1 2 2 1 9 8 6 3 4 5 7 5 3 4 2 7 1 8 6 9 1 6 5 4 8 2 9 7 3 3 2 7 6 1 9 5 4 8 4 9 8 5 3 7 1 2 6 9 5 1 7 2 8 6 3 4 8 4 2 3 5 6 7 9 1 6 7 3 1 9 4 2 8 5	0.004051 s
7 4 3 5 1 2 9 8 6 5 8 9 3 4 6 2 1 7 1 2 6 9 8 7 3 4 5 9 3 4 2 5 1 7 6 8 6 7 1 4 9 8 5 3 2 8 5 2 7 6 3 4 9 1 3 9 8 6 7 5 1 2 4 4 1 7 8 2 9 6 5 3 2 6 5 1 3 4 8 7 9	0.004005 s
7 8 2 6 1 4 3 5 9 4 3 9 8 2 5 1 7 6 6 5 1 9 3 7 4 2 8 2 9 3 4 7 1 8 6 5 5 6 8 3 9 2 7 1 4 1 4 7 5 6 8 2 9 3	0.004246 s

3 2 6 7 4 9 5 8 1 9 7 5 1 8 3 6 4 2 8 1 4 2 5 6 9 3 7	
4 2 8 5 3 1 7 9 6 3 6 5 9 4 7 1 8 2 9 7 1 2 6 8 4 3 5 2 1 4 8 9 6 5 7 3 6 9 7 4 5 3 2 1 8 5 8 3 1 7 2 9 6 4 8 4 9 6 1 5 3 2 7 7 5 2 3 8 9 6 4 1 1 3 6 7 2 4 8 5 9	0.002075 s
4 2 5 7 8 1 9 3 6 1 7 8 3 6 9 5 2 4 3 6 9 5 2 4 1 8 7 8 9 4 1 5 7 3 6 2 6 5 2 8 4 3 7 9 1 7 1 3 6 9 2 8 4 5 9 8 7 2 1 6 4 5 3 5 3 6 4 7 8 2 1 9 2 4 1 9 3 5 6 7 8	0.004597 s
3 4 8 2 6 7 9 5 1 5 7 1 9 4 3 6 2 8 2 6 9 1 8 5 3 7 4 6 9 7 3 5 1 4 8 2 1 2 3 8 7 4 5 9 6 8 5 4 6 2 9 1 3 7 4 1 5 7 9 8 2 6 3 9 8 2 4 3 6 7 1 5 7 3 6 5 1 2 8 4 9	0.00408 s
1 2 4 9 8 6 7 3 5 8 6 7 4 3 5 9 1 2 3 9 5 7 1 2 6 8 4 4 7 8 3 5 9 2 6 1 2 5 9 8 6 1 3 4 7 6 3 1 2 7 4 5 9 8 7 1 2 6 9 8 4 5 3 9 8 3 5 4 7 1 2 6 5 4 6 1 2 3 8 7 9	0.005004 s
3 6 1 5 2 4 7 8 9 7 8 9 3 6 1 4 2 5 5 2 4 8 7 9 3 6 1 8 9 3 1 5 7 6 4 2 4 1 2 6 8 3 5 9 7 6 5 7 9 4 2 1 3 8 1 4 8 7 9 6 2 5 3 2 3 5 4 1 8 9 7 6 9 7 6 2 3 5 8 1 4	0.004901 s
5 8 1 4 7 9 2 6 3	0.005705 s

3 2 9 1 5 6 8 4 7 6 4 7 3 2 8 1 5 9 9 5 6 7 3 1 4 2 8 2 3 8 9 6 4 5 7 1 7 1 4 5 8 2 9 3 6 1 7 2 6 9 5 3 8 4 8 9 3 2 4 7 6 1 5 4 6 5 8 1 3 7 9 2	
3 8 7 2 5 6 4 1 9 4 6 9 7 8 1 3 2 5 5 1 2 4 3 9 8 6 7 1 2 3 5 4 8 9 7 6 7 5 8 9 6 3 2 4 1 6 9 4 1 2 7 5 8 3 8 3 5 6 7 4 1 9 2 2 7 1 8 9 5 6 3 4 9 4 6 3 1 2 7 5 8	0 s
3 4 5 8 7 1 2 6 9 2 7 9 6 5 3 1 8 4 8 6 1 4 2 9 5 3 7 1 9 7 3 4 6 8 5 2 4 5 2 7 1 8 3 9 6 6 8 3 5 9 2 7 4 1 7 3 8 2 6 4 9 1 5 5 1 6 9 3 7 4 2 8 9 2 4 1 8 5 6 7 3	0.006232 s
2 3 5 7 6 1 4 8 9 4 1 9 3 2 8 5 7 6 8 6 7 5 4 9 2 1 3 7 4 6 1 3 5 9 2 8 5 2 1 8 9 6 7 3 4 9 8 3 4 7 2 6 5 1 3 9 4 2 8 7 1 6 5 6 5 2 9 1 3 8 4 7 1 7 8 6 5 4 3 9 2	0.007798 s
2 9 8 1 7 5 6 4 3 6 5 7 3 9 4 1 2 8 1 3 4 2 8 6 5 7 9 8 2 1 6 4 9 7 3 5 5 7 3 8 2 1 4 9 6 4 6 9 7 5 3 2 8 1 3 1 2 4 6 8 9 5 7 7 8 5 9 1 2 3 6 4 9 4 6 5 3 7 8 1 2	0.005037 s
7 6 1 5 4 3 2 8 9 8 3 2 7 9 1 6 4 5 5 4 9 6 2 8 1 3 7 3 7 4 2 1 5 9 6 8 1 2 8 9 3 6 5 7 4 6 9 5 4 8 7 3 2 1	0.006078 s

4 1 7 3 6 9 8 5 2 9 5 3 8 7 2 4 1 6 2 8 6 1 5 4 7 9 3	
1 3 2 6 4 9 7 8 5 7 5 8 2 1 3 6 4 9 9 6 4 7 8 5 1 2 3 5 4 3 8 9 7 2 1 6 2 7 6 5 3 1 8 9 4 8 9 1 4 2 6 5 3 7 6 1 9 3 7 8 4 5 2 3 2 7 1 5 4 9 6 8 4 8 5 9 6 2 3 7 1	0.004562 s
6 9 8 1 7 3 5 4 2 3 5 4 6 2 8 1 7 9 1 7 2 5 4 9 3 6 8 5 3 1 8 9 7 4 2 6 9 4 6 3 1 2 8 5 7 8 2 7 4 5 6 9 1 3 7 6 5 9 3 1 2 8 4 2 1 3 7 8 4 6 9 5 4 8 9 2 6 5 7 3 1	0.005901 s
8 5 2 7 1 6 9 4 3 1 9 7 8 4 3 6 5 2 4 6 3 9 2 5 1 8 7 2 7 8 6 3 4 5 9 1 6 4 5 1 7 9 3 2 8 9 3 1 5 8 2 4 7 6 7 8 6 4 9 1 2 3 5 3 1 4 2 5 8 7 6 9 5 2 9 3 6 7 8 1 4	0.004444 s
4 5 3 2 1 8 7 9 6 6 2 9 7 5 3 4 8 1 1 7 8 4 9 6 5 3 2 7 9 6 5 8 2 3 1 4 3 1 4 9 6 7 8 2 5 2 8 5 1 3 4 6 7 9 5 4 2 8 7 9 1 6 3 9 3 7 6 4 1 2 5 8 8 6 1 3 2 5 9 4 7	0.006334 s
5 1 6 2 8 9 3 4 7 8 4 9 1 7 3 2 5 6 7 3 2 4 6 5 9 1 8 6 9 8 3 1 7 5 2 4 3 2 7 9 5 4 8 6 1 1 5 4 8 2 6 7 3 9 9 6 1 7 3 2 4 8 5 2 7 5 6 4 8 1 9 3 4 8 3 5 9 1 6 7 2	0.003352 s
9 4 5 6 8 1 7 2 3	0 s

7 8 1 2 3 4 9 6 5 3 2 6 7 5 9 1 8 4 2 6 9 1 7 5 3 4 8 1 3 8 9 4 2 5 7 6 5 7 4 8 6 3 2 1 9 4 5 7 3 2 6 8 9 1 6 1 2 5 9 8 4 3 7 8 9 3 4 1 7 6 5 2	
3 6 5 9 4 2 8 7 1 1 2 8 7 5 6 4 9 3 9 7 4 8 1 3 5 6 2 8 1 9 4 3 5 6 2 7 5 3 7 2 6 8 1 4 9 6 4 2 1 7 9 3 5 8 2 9 6 3 8 4 7 1 5 7 5 3 6 9 1 2 8 4 4 8 1 5 2 7 9 3 6	0.003976 s
1 3 4 5 8 7 2 9 6 2 7 8 1 6 9 3 5 4 6 9 5 2 3 4 8 1 7 3 5 9 8 1 6 4 7 2 8 2 1 4 7 3 5 6 9 7 4 6 9 2 5 1 8 3 9 1 7 3 4 8 6 2 5 4 6 2 7 5 1 9 3 8 5 8 3 6 9 2 7 4 1	0.001424 s
1 9 3 6 7 2 4 8 5 4 6 2 3 5 8 9 7 1 7 8 5 9 1 4 6 2 3 5 3 8 2 9 6 7 1 4 6 7 4 1 3 5 2 9 8 2 1 9 4 8 7 3 5 6 8 2 6 7 4 1 5 3 9 9 4 1 5 2 3 8 6 7 3 5 7 8 6 9 1 4 2	0.001135 s
8 1 4 9 7 6 5 3 2 6 5 9 1 2 3 4 7 8 7 3 2 8 5 4 1 6 9 9 4 8 2 6 5 3 1 7 2 7 5 3 4 1 8 9 6 1 6 3 7 9 8 2 4 5 3 9 1 6 8 2 7 5 4 5 8 7 4 3 9 6 2 1 4 2 6 5 1 7 9 8 3	0.005472 s
3 8 4 5 6 7 9 2 1 1 2 6 4 3 9 7 8 5 7 5 9 8 2 1 3 4 6 5 6 3 7 9 8 2 1 4 8 4 7 3 1 2 6 5 9 9 1 2 6 4 5 8 7 3	0.006087 s

2 3 1 9 7 4 5 6 8 4 9 5 2 8 6 1 3 7 6 7 8 1 5 3 4 9 2	
4 6 9 1 5 8 3 7 2 7 1 2 4 6 3 8 5 9 5 3 8 2 9 7 6 4 1 9 2 7 6 3 4 5 1 8 3 8 5 7 1 9 4 2 6 1 4 6 5 8 2 7 9 3 6 5 3 9 4 1 2 8 7 2 9 4 8 7 6 1 3 5 8 7 1 3 2 5 9 6 4	0.001992 s
3 1 6 5 4 9 2 7 8 9 8 7 3 2 1 6 4 5 4 5 2 6 7 8 9 3 1 5 9 4 2 3 6 8 1 7 2 3 8 4 1 7 5 6 9 6 7 1 9 8 5 3 2 4 8 4 5 1 6 2 7 9 3 1 2 9 7 5 3 4 8 6 7 6 3 8 9 4 1 5 2	0.006149 s
5 8 6 1 2 7 9 4 3 7 2 3 4 6 9 8 5 1 4 9 1 8 5 3 2 6 7 1 3 5 9 7 4 6 2 8 2 7 9 6 1 8 5 3 4 6 4 8 5 3 2 1 7 9 9 1 7 2 4 6 3 8 5 3 5 2 7 8 1 4 9 6 8 6 4 3 9 5 7 1 2	0.005894 s
9 5 4 2 1 3 6 8 7 6 1 7 5 4 8 9 2 3 8 3 2 7 9 6 5 4 1 7 6 3 8 5 1 2 9 4 1 2 8 9 7 4 3 6 5 5 4 9 3 6 2 1 7 8 2 8 1 6 3 7 4 5 9 4 7 5 1 2 9 8 3 6 3 9 6 4 8 5 7 1 2	0.001786 s
1 5 9 7 4 3 8 6 2 2 7 6 5 8 9 4 3 1 3 4 8 6 1 2 7 5 9 6 2 4 9 7 8 3 1 5 9 1 7 2 3 5 6 8 4 5 8 3 1 6 4 2 9 7 4 3 5 8 2 1 9 7 6 8 6 1 4 9 7 5 2 3 7 9 2 3 5 6 1 4 8	0.0057 s
8 6 1 3 5 7 2 9 4	0.004436 s

5 9 7 4 8 2 3 6 1 4 3 2 6 1 9 7 8 5 9 1 6 2 7 5 8 4 3 3 5 8 9 6 4 1 2 7 2 7 4 1 3 8 9 5 6 7 8 9 5 4 1 6 3 2 1 4 3 8 2 6 5 7 9 6 2 5 7 9 3 4 1 8	
2 9 4 8 6 3 5 1 7 7 1 5 4 2 9 6 3 8 8 6 3 7 5 1 4 9 2 1 5 2 9 4 7 8 6 3 4 7 9 3 8 6 2 5 1 6 3 8 5 1 2 9 7 4 9 8 6 1 3 4 7 2 5 5 2 1 6 7 8 3 4 9 3 4 7 2 9 5 1 8 6	0.008426 s
3 5 1 2 8 6 4 9 7 4 9 2 1 5 7 6 3 8 7 8 6 9 3 4 5 1 2 2 7 5 4 6 9 1 8 3 9 3 8 5 2 1 7 6 4 6 1 4 8 7 3 2 5 9 8 2 9 6 4 5 3 7 1 1 6 3 7 9 2 8 4 5 5 4 7 3 1 8 9 2 6	0.006742 s

Analysis

The total time for solving the easy 50 Sudoku puzzle is 0.210271 and average time for each puzzle is 0.00420542s.

Extended Tasks

To further study our project the team completed three extended tasks.

- Task 1: Solved the “hard” input provided at magictour.free.fr/top95

  Solution and time required of 95 hard Sudoku puzzles:

  See *top95.txt.sol* in the *output* folder for each Sudoku puzzles solution and required time.

  The total time is 0.478208s, and the average time is 0.00503377s.

- Task 2: Make at least one alternation to the minimal encoding.

Originally in the minimal encoding, we have clauses ensure each number appears at most once in every row and each number appears at most once in every column, with formula: $\bigwedge_{i=1}^{9}\bigwedge_{k=1}^{9}\bigwedge_{j=1}^{8}\bigwedge_{\ell=j+1}^{9}(\neg x_{ijk}\vee\neg x_{i\ell k})$ and $\bigwedge_{j=1}^{9}\bigwedge_{k=1}^{9}\bigwedge_{i=1}^{8}\bigwedge_{\ell=i+1}^{9}(\neg x_{ijk}\vee\neg x_{\ell jk})$. We modify these two parts into clauses with each number appears at least once in each row, with formula $\bigwedge_{y=1}^{9}\bigwedge_{z=1}^{9}\bigvee_{x=1}^{9}s_{xyz}$, and each number appears at least once in each column, with formula $\bigwedge_{x=1}^{9}\bigwedge_{z=1}^{9}\bigvee_{y=1}^{9}s_{xyz}$. See sud2sat_v2.java for detailed implementation. Both implementations provide correct solutions. The implementation with alternation produces cnf with less clauses count but increases running time. We take the first Sudoku puzzle in the easy50 file, clauses count before the modification is 2968, and after modification is 2217. The required time before is 0.00306s, and after is 0.00533s.

- Task 3: Use of SAT-solvers other than miniSAT.

  We used Satz231 (new version) contributed by Chu-Min Li found at http://www.cs.ubc.ca/~hoos/SATLIB/solvers.html to solve the easy 50 Sudoku puzzles. Some modification was made to the original code by the team. All 'CLK_TCK' were changed to CLOCKS_PER_SECOND due to compiled error in the original code. See *satz.c* for detailed implementation.