	Department of Computer Engineering Faculty of Engineering University of Sri Jayewardenepura
Course	Advanced Algorithms
Course Code	CO4352
Title	Sudoku Solving Algorithm
Project Number	1
Outcomes	● Using algorithm design techniques to solve a given problem ● Understanding and implementing optimization techniques
Submission Deadline	Will be announced later

General Instructions:

- Please archive all files into a zip file, upload the zip file into LMS.
- Use the following format when you are naming the zip file: yy_ENG_xxx_Project.zip, yy_ENG_xxx is your registration number.

Sudoku Solving Algorithm

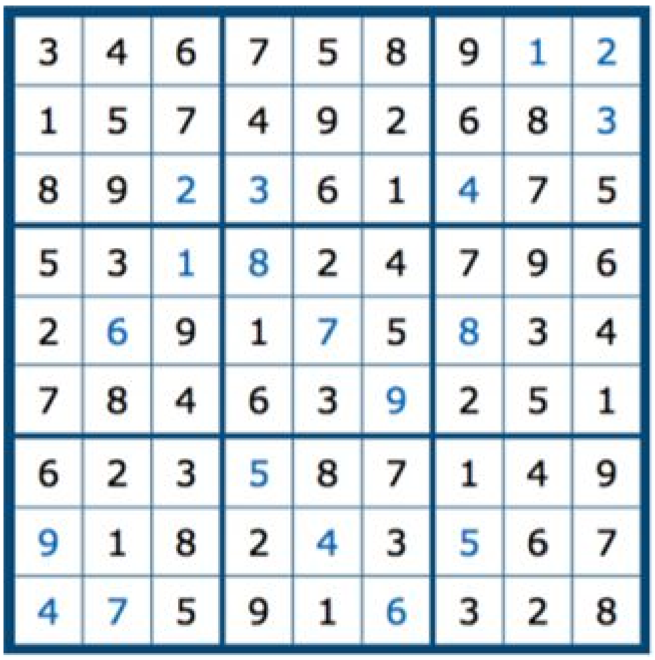
1. Introduction

Sudoku is a logic-based number-placement puzzle. The standard Sudoku puzzle consists of a 9x9 numeric grid that is divided into 9 3x3 sub-grids called “blocks”. The objective of the puzzle is to fill in the blank cells in a way such that none of the Sudoku rules is violated. The rules of the standard Sudoku puzzle are as follows:

Rules of the Standard Sudoku Puzzle:

- Each cell can only contain a number in the range of 1 to 9 (inclusive).
- For each 3 x 3 block, each number can only occur once.
- Similarly, for each row and column in the puzzle, each number can only occur once.

The blanks can be filled in any order as long as none of the rules above is violated. Once all the blank spaces are filled in without violating the rules, the puzzle is said to be solved.



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

Fig 1: Example of Sudoku Puzzle

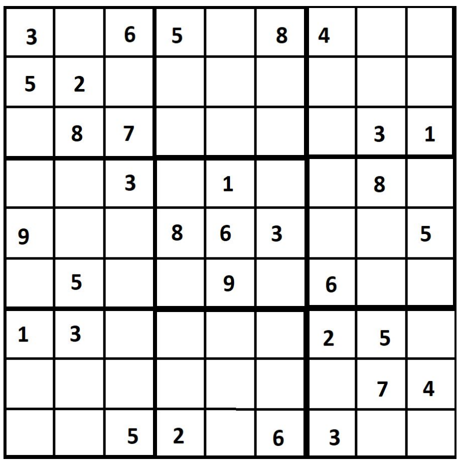
2. Implementing an algorithm to solve the standard Sudoku puzzle

Objective

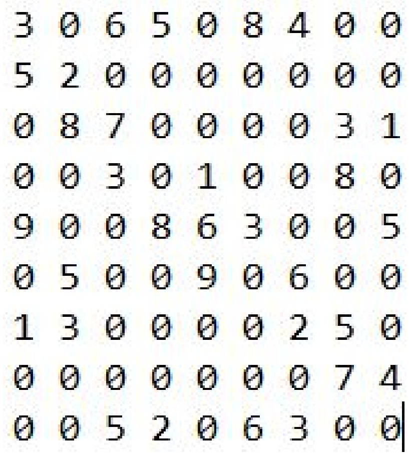
You are required to implement an algorithm capable of solving any given standard Sudoku puzzle as described above. Your implementation must focus on efficiency, and the time complexity of your solution will be tested (Refer to section 4 below).

Implementation and Requirements

- Your algorithm should be implemented in either **Python** or **C++**.
- The algorithm must be able to solve any general solvable standard Sudoku puzzle.
- The input puzzle must be a text file in the root folder of the program and your program must accept a command-line argument with the name of the input file.
  Eg: The command “./sudoku_solver input1.txt” where ‘sudoku_solver’ is the name of your program and ‘input1.txt’ is the text file containing the input puzzle.
- The input text file must contain the input puzzle in the following format:
  - 9 lines corresponding to the 9 rows
  - Each line contains 9 numbers separated by a single space corresponding to the 9 columns
  - The blank spaces will be denoted by zeros
  - The text file must not contain any other characters or unnecessary spaces, tabs etc



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



```
3 0 6 5 0 8 4 0 0
5 2 0 0 0 0 0 0 0
0 8 7 0 0 0 0 3 1
0 0 3 0 1 0 0 8 0
9 0 0 8 6 3 0 0 5
0 5 0 0 9 0 6 0 0
1 3 0 0 0 0 2 5 0
0 0 0 0 0 0 0 7 4
0 0 5 2 0 6 3 0 0
```

Fig 2: The puzzle in the right will be denoted by the text on the left in the input file.

- The solution must also be written to a text file in the root folder of the program. The output file must be named the same as the input file name with the suffix “_output” added at the end.
  Eg: If “input1.txt” is the input file, the output must be written to a file named “input1_output.txt”.
- There may be instances where a given puzzle may not have any correct solutions. In this case, the algorithm must write the string “No Solution” to the output file.

3. Large Sudoku Puzzle - Hexadoku

Hexadoku is a variant of the standard Sudoku puzzle where a 16x16 grid is used instead of the standard 9x9 grid. The 16x16 grid is then divided into 16 4x4 blocks. Instead of using integers only from 1-9 to fill in the cells, we use the integers from 1- 16 inclusive. The rest of the rules remain the same as in the standard Sudoku puzzle.

Extend your algorithm in the earlier section to be able to solve Hexadoku puzzles as well. Your program must be able to detect whether the given input puzzle is a standard 9x9 puzzle or a 16x16 puzzle automatically and adjust accordingly.

4. Efficiency and Time Complexity

The efficiency of your algorithm will be tested. You should aim to have your algorithm solve 9x9 Sudoku puzzles in under 4 seconds on average and 16x16 hexadoku puzzles in under 10 seconds on average.

Hint: See how you could reduce the number of possibilities for each cell and the order in which you fill in the blank cells.

5. Submission Guidelines

Along with all the source files, submit a report (maximum 5 pages including the cover page) which includes the following information.

- Introduction
- Background and constraints
- Implementation of the Algorithm
- Optimization techniques
- Challenges faced
- Limitations

- Future Improvements

6. Evaluation Criteria

70% will be allocated to the individual project evaluation. 30% will be allocated according to the ranking of your project based on the average time it takes to solve a set of Sudoku Puzzles. If your rank is 1st, you’ll get 30 marks. If your rank is 30, you’ll get 0 marks.