Artificial Intelligence

Assignment 3

SUDOKU SOLVER

**Atanu Ghosh** (**110280569)**

**Bhargava Mourya Venishetty (110308227)**

**ReadMe:**

**Files Required:**

csp.py

sudoku.py

min\_conflict\_sudoku.py

game.txt

**Steps to run:**

1. Update the game.txt with the puzzle you want to solve. Format has to be

N,M,K;

Followed by N lines. Each line describes a row of the puzzle. Each line should contain N comma separated values (1 to N or ‘-‘) and must end with semi-colon.

1. Run Sudoku.py with –input game.txt as arguments
2. Note : All the required algorithms are implemented in csp.py

**Which Algorithm Should be used?**

**Backtracking:**

Backtracking is a brute-force way to solve the problem. It works fine if the puzzle size is small (N = 10) or puzzle is almost empty.

**Backtracking + MRV:**

The MRV heuristic is used to alter the order in which squares are guessed in order to reduce the number of branches at each level. Basically instead of choosing the first empty square, the square with the least number of possible values is chosen. Use Backtracking + MRV heuristic if the puzzle size is just greater than what is recommended for simple backtracking.

**Backtracking + MRV + Forward Checking:**

Forward Checking is used along with MRV quite often. The simple version of backtracking search had to place a value and then check for conflicts. Instead it is easier to just maintain a list of which possible values each square can possibly have given the other numbers that have been assigned. Then when the values are being assigned to that square, only consider the ones that do not directly conflict with the other already placed numbers.

Backtracking + MRV + Forward Checking reduces the recursion depth and often outputs the solution much faster than general backtracking or Backtracking + MRV. For small size puzzles, it is advised to use general backtracking as forward checking has a huge space complexity. But if puzzle size is greater than 12 \* 12, use Backtracking + MRV + Forward Checking.

**Backtracking + MRV + Constraint Checking:**

Forward checking can only catch conflicts right before they cause a certain branch to fail. It is possible to detect errors even earlier and prune off entire branches using Constraint Propagation. For the puzzle size greater than 14, Constraint Propagation performs better than any of the above techniques.

**Min Conflict:**

Very powerful in practice, requires very less number of consistency checks. Can be used for large boards. Eg: N-Queens where N = 50 can be solved really fast with Min Conflict.

**Sample Test Results:**

**INPUT # 1**

**12,3,4;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**-,-,-,-,-,-,-,-,-,-,-,-;**

**Backtracking:**

Execution Time: 0.160035989667

Consistency Checks: 7425

Solution: [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4], [9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8], [2, 1, 4, 3, 6, 5, 8, 7, 10, 9, 12, 11], [6, 5, 8, 7, 10, 9, 12, 11, 2, 1, 4, 3], [10, 9, 12, 11, 2, 1, 4, 3, 6, 5, 8, 7], [3, 4, 1, 2, 7, 8, 5, 6, 11, 12, 9, 10], [7, 8, 5, 6, 11, 12, 9, 10, 3, 4, 1, 2], [11, 12, 9, 10, 3, 4, 1, 2, 7, 8, 5, 6], [4, 3, 2, 1, 8, 7, 6, 5, 12, 11, 10, 9], [8, 7, 6, 5, 12, 11, 10, 9, 4, 3, 2, 1], [12, 11, 10, 9, 4, 3, 2, 1, 8, 7, 6, 5]]

**backtrackingMRV:**

Execution Time: 0.12821447036

Consistency Checks: 144

Solution: [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4], [9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8], [2, 7, 1, 3, 10, 9, 4, 5, 11, 12, 8, 6], [10, 11, 12, 5, 7, 8, 2, 6, 3, 1, 4, 9], [4, 8, 9, 6, 11, 12, 1, 3, 7, 5, 2, 10], [3, 1, 6, 10, 2, 4, 9, 11, 8, 7, 12, 5], [12, 5, 4, 7, 8, 3, 10, 1, 6, 11, 9, 2], [8, 9, 2, 11, 6, 5, 12, 7, 4, 3, 10, 1], [6, 3, 8, 2, 12, 7, 5, 9, 10, 4, 1, 11], [7, 12, 5, 1, 4, 11, 8, 10, 2, 9, 6, 3], [11, 4, 10, 9, 3, 1, 6, 2, 12, 8, 5, 7]]

**backtrackingMRVfwd:**

Execution Time: 0.128973948117

Consistency Checks: 144

Solution: [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4], [9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8], [2, 7, 1, 3, 10, 9, 4, 5, 11, 12, 8, 6], [10, 11, 12, 5, 7, 8, 2, 6, 3, 1, 4, 9], [4, 8, 9, 6, 11, 12, 1, 3, 7, 5, 2, 10], [3, 1, 6, 10, 2, 4, 9, 11, 8, 7, 12, 5], [12, 5, 4, 7, 8, 3, 10, 1, 6, 11, 9, 2], [8, 9, 2, 11, 6, 5, 12, 7, 4, 3, 10, 1], [6, 3, 8, 2, 12, 7, 5, 9, 10, 4, 1, 11], [7, 12, 5, 1, 4, 11, 8, 10, 2, 9, 6, 3], [11, 4, 10, 9, 3, 1, 6, 2, 12, 8, 5, 7]]

**backtrackingMRVcp:**

Execution Time: 0.135899051038

Consistency Checks: 144

Solution: [[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4], [9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8], [2, 7, 1, 3, 10, 9, 4, 5, 11, 12, 8, 6], [10, 11, 12, 5, 7, 8, 2, 6, 3, 1, 4, 9], [4, 8, 9, 6, 11, 12, 1, 3, 7, 5, 2, 10], [3, 1, 6, 10, 2, 4, 9, 11, 8, 7, 12, 5], [12, 5, 4, 7, 8, 3, 10, 1, 6, 11, 9, 2], [8, 9, 2, 11, 6, 5, 12, 7, 4, 3, 10, 1], [6, 3, 8, 2, 12, 7, 5, 9, 10, 4, 1, 11], [7, 12, 5, 1, 4, 11, 8, 10, 2, 9, 6, 3], [11, 4, 10, 9, 3, 1, 6, 2, 12, 8, 5, 7]]

**minConflict:**

Execution Time: 5.15154959548

Consistency Checks: 4920

Solution: [[7, 10, 3, 8, 11, 2, 4, 12, 5, 6, 1, 9], [6, 11, 9, 12, 1, 3, 8, 5, 10, 4, 7, 2], [5, 2, 4, 1, 10, 9, 7, 6, 3, 11, 12, 8], [1, 7, 8, 4, 6, 12, 3, 9, 2, 10, 11, 5], [9, 6, 2, 10, 4, 5, 1, 11, 12, 7, 8, 3], [11, 12, 5, 3, 8, 7, 10, 2, 6, 1, 9, 4], [12, 1, 7, 5, 9, 6, 11, 3, 4, 8, 2, 10], [2, 4, 6, 9, 7, 8, 12, 10, 11, 5, 3, 1], [8, 3, 10, 11, 2, 1, 5, 4, 7, 9, 6, 12], [10, 5, 12, 6, 3, 11, 9, 8, 1, 2, 4, 7], [3, 9, 11, 7, 5, 4, 2, 1, 8, 12, 10, 6], [4, 8, 1, 2, 12, 10, 6, 7, 9, 3, 5, 11]]

**INPUT # 2**

12,3,4;

1,-,-,-,-,-,-,-,-,-,-,11;

-,2,-,-,-,-,-,-,-,-,12,-;

-,-,3,-,-,-,-,-,-,1,-,-;

-,-,-,4,-,-,-,-,2,-,-,-;

-,-,-,-,9,-,-,3,-,-,-,-;

-,-,-,-,-,6,4,-,-,-,-,-;

-,-,-,-,-,5,7,-,-,-,-,-;

-,-,-,-,11,-,-,8,-,-,-,-;

-,-,-,7,-,-,-,-,9,-,-,-;

-,-,8,-,-,-,1,-,-,10,-,-;

-,9,-,-,-,-,10,-,-,-,11,-;

10,-,-,-,-,-,6,-,-,-,-,12;

**Backtracking:**

Execution Time: 151.990319652

Consistency Checks: 5199583

Solution: [[1, 4, 5, 6, 2, 3, 8, 12, 7, 9, 10, 11], [7, 2, 9, 8, 1, 4, 11, 10, 3, 5, 12, 6], [11, 10, 3, 12, 5, 7, 9, 6, 4, 1, 2, 8], [3, 1, 6, 4, 7, 8, 5, 11, 2, 12, 9, 10], [2, 5, 7, 10, 9, 1, 12, 3, 11, 6, 8, 4], [8, 11, 12, 9, 10, 6, 4, 2, 1, 3, 5, 7], [4, 3, 10, 11, 6, 5, 7, 9, 12, 8, 1, 2], [9, 6, 1, 5, 11, 12, 2, 8, 10, 4, 7, 3], [12, 8, 2, 7, 4, 10, 3, 1, 9, 11, 6, 5], [5, 12, 8, 2, 3, 11, 1, 7, 6, 10, 4, 9], [6, 9, 4, 3, 12, 2, 10, 5, 8, 7, 11, 1], [10, 7, 11, 1, 8, 9, 6, 4, 5, 2, 3, 12]]

**backtrackingMRV:**

Execution Time: 0.157214033746

Consistency Checks: 201

Solution: [[1, 8, 10, 12, 2, 7, 5, 6, 3, 4, 9, 11], [4, 2, 11, 9, 3, 10, 8, 1, 6, 5, 12, 7], [6, 7, 3, 5, 4, 11, 9, 12, 8, 1, 2, 10], [3, 6, 9, 4, 7, 1, 11, 5, 2, 12, 10, 8], [7, 1, 2, 10, 9, 8, 12, 3, 11, 6, 5, 4], [11, 12, 5, 8, 10, 6, 4, 2, 1, 9, 7, 3], [2, 10, 4, 11, 1, 5, 7, 9, 12, 3, 8, 6], [9, 5, 1, 6, 11, 12, 3, 8, 10, 7, 4, 2], [8, 3, 12, 7, 6, 4, 2, 10, 9, 11, 1, 5], [12, 11, 8, 2, 5, 3, 1, 7, 4, 10, 6, 9], [5, 9, 6, 3, 12, 2, 10, 4, 7, 8, 11, 1], [10, 4, 7, 1, 8, 9, 6, 11, 5, 2, 3, 12]]

**backtrackingMRVfwd:**

Execution Time: 0.156967289002

Consistency Checks: 196

Solution: [[1, 8, 10, 12, 2, 7, 5, 6, 3, 4, 9, 11], [4, 2, 11, 9, 3, 10, 8, 1, 6, 5, 12, 7], [6, 7, 3, 5, 4, 11, 9, 12, 8, 1, 2, 10], [3, 6, 9, 4, 7, 1, 11, 5, 2, 12, 10, 8], [7, 1, 2, 10, 9, 8, 12, 3, 11, 6, 5, 4], [11, 12, 5, 8, 10, 6, 4, 2, 1, 9, 7, 3], [2, 10, 4, 11, 1, 5, 7, 9, 12, 3, 8, 6], [9, 5, 1, 6, 11, 12, 3, 8, 10, 7, 4, 2], [8, 3, 12, 7, 6, 4, 2, 10, 9, 11, 1, 5], [12, 11, 8, 2, 5, 3, 1, 7, 4, 10, 6, 9], [5, 9, 6, 3, 12, 2, 10, 4, 7, 8, 11, 1], [10, 4, 7, 1, 8, 9, 6, 11, 5, 2, 3, 12]]

**backtrackingMRVcp:**

Execution Time: 0.17053653939

Consistency Checks: 196

Solution: [[1, 8, 10, 12, 2, 7, 5, 6, 3, 4, 9, 11], [4, 2, 11, 9, 3, 10, 8, 1, 6, 5, 12, 7], [6, 7, 3, 5, 4, 11, 9, 12, 8, 1, 2, 10], [3, 6, 9, 4, 7, 1, 11, 5, 2, 12, 10, 8], [7, 1, 2, 10, 9, 8, 12, 3, 11, 6, 5, 4], [11, 12, 5, 8, 10, 6, 4, 2, 1, 9, 7, 3], [2, 10, 4, 11, 1, 5, 7, 9, 12, 3, 8, 6], [9, 5, 1, 6, 11, 12, 3, 8, 10, 7, 4, 2], [8, 3, 12, 7, 6, 4, 2, 10, 9, 11, 1, 5], [12, 11, 8, 2, 5, 3, 1, 7, 4, 10, 6, 9], [5, 9, 6, 3, 12, 2, 10, 4, 7, 8, 11, 1], [10, 4, 7, 1, 8, 9, 6, 11, 5, 2, 3, 12]]

**minConflict:**

Execution Time: 6.53418911424

Consistency Checks: 6292

Solution: [[1, 5, 6, 8, 7, 12, 2, 10, 3, 4, 9, 11], [4, 2, 11, 10, 3, 1, 9, 6, 8, 5, 12, 7], [7, 12, 3, 9, 8, 11, 5, 4, 10, 1, 2, 6], [8, 6, 1, 4, 12, 10, 11, 7, 2, 9, 3, 5], [5, 10, 12, 11, 9, 2, 8, 3, 6, 7, 1, 4], [9, 7, 2, 3, 1, 6, 4, 5, 12, 11, 10, 8], [3, 8, 9, 12, 6, 5, 7, 1, 11, 2, 4, 10], [2, 4, 10, 6, 11, 9, 3, 8, 7, 12, 5, 1], [11, 1, 5, 7, 10, 4, 12, 2, 9, 6, 8, 3], [12, 3, 8, 2, 4, 7, 1, 11, 5, 10, 6, 9], [6, 9, 7, 1, 5, 8, 10, 12, 4, 3, 11, 2], [10, 11, 4, 5, 2, 3, 6, 9, 1, 8, 7, 12]]

**INPUT # 3**

12,3,4;

-,3,-,-,11,-,-,-,10,-,-,12;

-,11,-,-,-,-,-,-,-,-,7,-;

7,-,-,2,-,-,-,9,-,-,6,-;

-,4,-,5,-,-,-,-,-,6,-,-;

-,-,7,-,4,10,9,12,-,5,-,-;

-,-,11,-,-,-,-,-,4,-,10,-;

-,8,-,6,-,-,-,-,-,11,-,-;

-,-,3,-,10,6,12,4,-,7,-,-;

-,-,5,-,-,-,-,-,12,-,9,-;

-,5,-,-,9,-,-,-,8,-,-,11;

-,9,-,-,-,-,-,-,-,-,2,-;

3,-,-,8,-,-,-,7,-,-,5,-;

**Backtracking:**

Execution Time: 57.5117960665

Consistency Checks: 2136172

Solution: [[1, 3, 4, 9, 11, 5, 7, 6, 10, 2, 8, 12], [5, 11, 6, 10, 1, 12, 2, 8, 9, 3, 7, 4], [7, 12, 8, 2, 3, 4, 10, 9, 11, 1, 6, 5], [10, 4, 9, 5, 2, 1, 8, 11, 3, 6, 12, 7], [8, 6, 7, 3, 4, 10, 9, 12, 1, 5, 11, 2], [2, 1, 11, 12, 5, 7, 6, 3, 4, 8, 10, 9], [12, 8, 10, 6, 7, 9, 5, 1, 2, 11, 4, 3], [9, 2, 3, 11, 10, 6, 12, 4, 5, 7, 1, 8], [4, 7, 5, 1, 8, 3, 11, 2, 12, 10, 9, 6], [6, 5, 12, 7, 9, 2, 1, 10, 8, 4, 3, 11], [11, 9, 1, 4, 6, 8, 3, 5, 7, 12, 2, 10], [3, 10, 2, 8, 12, 11, 4, 7, 6, 9, 5, 1]]

**backtrackingMRV:**

Execution Time: 0.0994796123413

Consistency Checks: 113

Solution: [[5, 3, 6, 4, 11, 7, 8, 2, 10, 9, 1, 12], [1, 11, 8, 9, 12, 5, 6, 10, 2, 4, 7, 3], [7, 10, 12, 2, 1, 3, 4, 9, 11, 8, 6, 5], [10, 4, 9, 5, 3, 8, 2, 11, 7, 6, 12, 1], [6, 2, 7, 1, 4, 10, 9, 12, 3, 5, 11, 8], [8, 12, 11, 3, 6, 1, 7, 5, 4, 2, 10, 9], [12, 8, 2, 6, 7, 9, 5, 3, 1, 11, 4, 10], [9, 1, 3, 11, 10, 6, 12, 4, 5, 7, 8, 2], [4, 7, 5, 10, 8, 2, 11, 1, 12, 3, 9, 6], [2, 5, 4, 7, 9, 12, 1, 6, 8, 10, 3, 11], [11, 9, 10, 12, 5, 4, 3, 8, 6, 1, 2, 7], [3, 6, 1, 8, 2, 11, 10, 7, 9, 12, 5, 4]]

**backtrackingMRVfwd:**

Execution Time: 0.0975039438534

Consistency Checks: 111

Solution: [[5, 3, 6, 4, 11, 7, 8, 2, 10, 9, 1, 12], [1, 11, 8, 9, 12, 5, 6, 10, 2, 4, 7, 3], [7, 10, 12, 2, 1, 3, 4, 9, 11, 8, 6, 5], [10, 4, 9, 5, 3, 8, 2, 11, 7, 6, 12, 1], [6, 2, 7, 1, 4, 10, 9, 12, 3, 5, 11, 8], [8, 12, 11, 3, 6, 1, 7, 5, 4, 2, 10, 9], [12, 8, 2, 6, 7, 9, 5, 3, 1, 11, 4, 10], [9, 1, 3, 11, 10, 6, 12, 4, 5, 7, 8, 2], [4, 7, 5, 10, 8, 2, 11, 1, 12, 3, 9, 6], [2, 5, 4, 7, 9, 12, 1, 6, 8, 10, 3, 11], [11, 9, 10, 12, 5, 4, 3, 8, 6, 1, 2, 7], [3, 6, 1, 8, 2, 11, 10, 7, 9, 12, 5, 4]]

**backtrackingMRVcp:**

Execution Time: 0.10523043265

Consistency Checks: 111

Solution: [[5, 3, 6, 4, 11, 7, 8, 2, 10, 9, 1, 12], [1, 11, 8, 9, 12, 5, 6, 10, 2, 4, 7, 3], [7, 10, 12, 2, 1, 3, 4, 9, 11, 8, 6, 5], [10, 4, 9, 5, 3, 8, 2, 11, 7, 6, 12, 1], [6, 2, 7, 1, 4, 10, 9, 12, 3, 5, 11, 8], [8, 12, 11, 3, 6, 1, 7, 5, 4, 2, 10, 9], [12, 8, 2, 6, 7, 9, 5, 3, 1, 11, 4, 10], [9, 1, 3, 11, 10, 6, 12, 4, 5, 7, 8, 2], [4, 7, 5, 10, 8, 2, 11, 1, 12, 3, 9, 6], [2, 5, 4, 7, 9, 12, 1, 6, 8, 10, 3, 11], [11, 9, 10, 12, 5, 4, 3, 8, 6, 1, 2, 7], [3, 6, 1, 8, 2, 11, 10, 7, 9, 12, 5, 4]]

**minConflict:**

Execution Time: 7.83769370211

Consistency Checks: 10620

Solution: [[5, 3, 8, 9, 11, 2, 7, 6, 10, 4, 1, 12], [6, 11, 4, 12, 1, 5, 10, 8, 3, 9, 7, 2], [7, 1, 10, 2, 3, 12, 4, 9, 11, 8, 6, 5], [10, 4, 9, 5, 8, 1, 11, 2, 7, 6, 12, 3], [2, 6, 7, 3, 4, 10, 9, 12, 1, 5, 11, 8], [8, 12, 11, 1, 6, 7, 3, 5, 4, 2, 10, 9], [1, 8, 12, 6, 7, 9, 5, 3, 2, 11, 4, 10], [9, 2, 3, 11, 10, 6, 12, 4, 5, 7, 8, 1], [4, 7, 5, 10, 2, 8, 1, 11, 12, 3, 9, 6], [12, 5, 6, 7, 9, 4, 2, 1, 8, 10, 3, 11], [11, 9, 1, 4, 5, 3, 8, 10, 6, 12, 2, 7], [3, 10, 2, 8, 12, 11, 6, 7, 9, 1, 5, 4]]

**INPUT # 4**

12,3,4;

-,2,12,-,-,-,1,4,-,8,-,-;

-,9,-,-,-,2,-,-,-,4,1,10;

10,1,-,-,5,-,6,9,-,-,-,12;

-,-,-,-,1,7,-,-,-,-,-,-;

9,-,6,-,2,-,-,12,5,7,-,-;

4,-,3,-,-,-,-,-,11,-,2,-;

-,5,-,12,-,-,-,-,-,1,-,4;

-,-,1,2,11,-,-,5,-,10,-,6;

-,-,-,-,-,-,3,7,-,-,-,-;

12,-,-,-,10,3,-,8,-,-,6,11;

1,10,5,-,-,-,9,-,-,-,12,-;

-,-,7,-,12,5,-,-,-,9,10,-;

**Backtracking:**

Execution Time: 1.12871180855

Consistency Checks: 46018

Solution: [[3, 2, 12, 11, 7, 10, 1, 4, 6, 8, 9, 5], [5, 9, 8, 6, 3, 2, 12, 11, 7, 4, 1, 10], [10, 1, 4, 7, 5, 8, 6, 9, 2, 3, 11, 12], [2, 12, 11, 5, 1, 7, 8, 3, 10, 6, 4, 9], [9, 8, 6, 10, 2, 4, 11, 12, 5, 7, 3, 1], [4, 7, 3, 1, 6, 9, 5, 10, 11, 12, 2, 8], [11, 5, 9, 12, 8, 6, 10, 2, 3, 1, 7, 4], [7, 3, 1, 2, 11, 12, 4, 5, 9, 10, 8, 6], [8, 6, 10, 4, 9, 1, 3, 7, 12, 11, 5, 2], [12, 4, 2, 9, 10, 3, 7, 8, 1, 5, 6, 11], [1, 10, 5, 3, 4, 11, 9, 6, 8, 2, 12, 7], [6, 11, 7, 8, 12, 5, 2, 1, 4, 9, 10, 3]]

**backtrackingMRV:**

Execution Time: 4.38378087356

Consistency Checks: 5032

Solution: [[3, 2, 12, 11, 7, 10, 1, 4, 6, 8, 9, 5], [5, 9, 8, 6, 3, 2, 12, 11, 7, 4, 1, 10], [10, 1, 4, 7, 5, 8, 6, 9, 2, 3, 11, 12], [2, 12, 11, 5, 1, 7, 8, 3, 10, 6, 4, 9], [9, 8, 6, 10, 2, 4, 11, 12, 5, 7, 3, 1], [4, 7, 3, 1, 6, 9, 5, 10, 11, 12, 2, 8], [11, 5, 9, 12, 8, 6, 10, 2, 3, 1, 7, 4], [7, 3, 1, 2, 11, 12, 4, 5, 9, 10, 8, 6], [8, 6, 10, 4, 9, 1, 3, 7, 12, 11, 5, 2], [12, 4, 2, 9, 10, 3, 7, 8, 1, 5, 6, 11], [1, 10, 5, 3, 4, 11, 9, 6, 8, 2, 12, 7], [6, 11, 7, 8, 12, 5, 2, 1, 4, 9, 10, 3]]

**backtrackingMRVfwd:**

Execution Time: 4.06660483081

Consistency Checks: 4602

Solution: [[3, 2, 12, 11, 7, 10, 1, 4, 6, 8, 9, 5], [5, 9, 8, 6, 3, 2, 12, 11, 7, 4, 1, 10], [10, 1, 4, 7, 5, 8, 6, 9, 2, 3, 11, 12], [2, 12, 11, 5, 1, 7, 8, 3, 10, 6, 4, 9], [9, 8, 6, 10, 2, 4, 11, 12, 5, 7, 3, 1], [4, 7, 3, 1, 6, 9, 5, 10, 11, 12, 2, 8], [11, 5, 9, 12, 8, 6, 10, 2, 3, 1, 7, 4], [7, 3, 1, 2, 11, 12, 4, 5, 9, 10, 8, 6], [8, 6, 10, 4, 9, 1, 3, 7, 12, 11, 5, 2], [12, 4, 2, 9, 10, 3, 7, 8, 1, 5, 6, 11], [1, 10, 5, 3, 4, 11, 9, 6, 8, 2, 12, 7], [6, 11, 7, 8, 12, 5, 2, 1, 4, 9, 10, 3]]

**backtrackingMRVcp:**

Execution Time: 4.56434542725

Consistency Checks: 4608

Solution: [[3, 2, 12, 11, 7, 10, 1, 4, 6, 8, 9, 5], [5, 9, 8, 6, 3, 2, 12, 11, 7, 4, 1, 10], [10, 1, 4, 7, 5, 8, 6, 9, 2, 3, 11, 12], [2, 12, 11, 5, 1, 7, 8, 3, 10, 6, 4, 9], [9, 8, 6, 10, 2, 4, 11, 12, 5, 7, 3, 1], [4, 7, 3, 1, 6, 9, 5, 10, 11, 12, 2, 8], [11, 5, 9, 12, 8, 6, 10, 2, 3, 1, 7, 4], [7, 3, 1, 2, 11, 12, 4, 5, 9, 10, 8, 6], [8, 6, 10, 4, 9, 1, 3, 7, 12, 11, 5, 2], [12, 4, 2, 9, 10, 3, 7, 8, 1, 5, 6, 11], [1, 10, 5, 3, 4, 11, 9, 6, 8, 2, 12, 7], [6, 11, 7, 8, 12, 5, 2, 1, 4, 9, 10, 3]]

**INPUT # 5**

12,3,4;

1,9,-,-,11,8,-,-,-,10,-,-;

5,2,10,-,-,12,1,-,-,8,-,-;

-,8,3,4,-,7,-,9,-,1,6,5;

-,-,9,-,4,-,-,5,12,6,3,-;

11,12,8,-,3,1,-,-,-,9,7,-;

3,5,4,-,9,11,-,-,-,-,10,-;

9,-,2,12,-,-,-,-,-,-,-,7;

-,3,-,8,-,5,-,7,10,-,-,2;

-,-,-,-,1,6,-,10,3,-,9,-;

10,1,-,2,-,-,11,8,9,-,-,-;

-,-,11,-,-,9,12,6,-,-,2,-;

-,-,5,-,10,-,-,-,4,-,-,12;

**Backtracking:**

Execution Time: 0.00710385387922

Consistency Checks: 209

Solution: [[1, 9, 6, 7, 11, 8, 5, 4, 2, 10, 12, 3], [5, 2, 10, 11, 6, 12, 1, 3, 7, 8, 4, 9], [12, 8, 3, 4, 2, 7, 10, 9, 11, 1, 6, 5], [2, 7, 9, 1, 4, 10, 8, 5, 12, 6, 3, 11], [11, 12, 8, 10, 3, 1, 6, 2, 5, 9, 7, 4], [3, 5, 4, 6, 9, 11, 7, 12, 8, 2, 10, 1], [9, 10, 2, 12, 8, 3, 4, 11, 6, 5, 1, 7], [6, 3, 1, 8, 12, 5, 9, 7, 10, 4, 11, 2], [4, 11, 7, 5, 1, 6, 2, 10, 3, 12, 9, 8], [10, 1, 12, 2, 7, 4, 11, 8, 9, 3, 5, 6], [8, 4, 11, 3, 5, 9, 12, 6, 1, 7, 2, 10], [7, 6, 5, 9, 10, 2, 3, 1, 4, 11, 8, 12]]

**backtrackingMRV**:

Execution Time: 0.0739233568917

Consistency Checks: 76

Solution: [[1, 9, 6, 7, 11, 8, 5, 4, 2, 10, 12, 3], [5, 2, 10, 11, 6, 12, 1, 3, 7, 8, 4, 9], [12, 8, 3, 4, 2, 7, 10, 9, 11, 1, 6, 5], [2, 7, 9, 1, 4, 10, 8, 5, 12, 6, 3, 11], [11, 12, 8, 10, 3, 1, 6, 2, 5, 9, 7, 4], [3, 5, 4, 6, 9, 11, 7, 12, 8, 2, 10, 1], [9, 10, 2, 12, 8, 3, 4, 11, 6, 5, 1, 7], [6, 3, 1, 8, 12, 5, 9, 7, 10, 4, 11, 2], [4, 11, 7, 5, 1, 6, 2, 10, 3, 12, 9, 8], [10, 1, 12, 2, 7, 4, 11, 8, 9, 3, 5, 6], [8, 4, 11, 3, 5, 9, 12, 6, 1, 7, 2, 10], [7, 6, 5, 9, 10, 2, 3, 1, 4, 11, 8, 12]]

**backtrackingMRVfwd:**

Execution Time: 0.0741863517229

Consistency Checks: 76

Solution: [[1, 9, 6, 7, 11, 8, 5, 4, 2, 10, 12, 3], [5, 2, 10, 11, 6, 12, 1, 3, 7, 8, 4, 9], [12, 8, 3, 4, 2, 7, 10, 9, 11, 1, 6, 5], [2, 7, 9, 1, 4, 10, 8, 5, 12, 6, 3, 11], [11, 12, 8, 10, 3, 1, 6, 2, 5, 9, 7, 4], [3, 5, 4, 6, 9, 11, 7, 12, 8, 2, 10, 1], [9, 10, 2, 12, 8, 3, 4, 11, 6, 5, 1, 7], [6, 3, 1, 8, 12, 5, 9, 7, 10, 4, 11, 2], [4, 11, 7, 5, 1, 6, 2, 10, 3, 12, 9, 8], [10, 1, 12, 2, 7, 4, 11, 8, 9, 3, 5, 6], [8, 4, 11, 3, 5, 9, 12, 6, 1, 7, 2, 10], [7, 6, 5, 9, 10, 2, 3, 1, 4, 11, 8, 12]]

**backtrackingMRVcp:**

Execution Time: 0.0761385003479

Consistency Checks: 76

Solution: [[1, 9, 6, 7, 11, 8, 5, 4, 2, 10, 12, 3], [5, 2, 10, 11, 6, 12, 1, 3, 7, 8, 4, 9], [12, 8, 3, 4, 2, 7, 10, 9, 11, 1, 6, 5], [2, 7, 9, 1, 4, 10, 8, 5, 12, 6, 3, 11], [11, 12, 8, 10, 3, 1, 6, 2, 5, 9, 7, 4], [3, 5, 4, 6, 9, 11, 7, 12, 8, 2, 10, 1], [9, 10, 2, 12, 8, 3, 4, 11, 6, 5, 1, 7], [6, 3, 1, 8, 12, 5, 9, 7, 10, 4, 11, 2], [4, 11, 7, 5, 1, 6, 2, 10, 3, 12, 9, 8], [10, 1, 12, 2, 7, 4, 11, 8, 9, 3, 5, 6], [8, 4, 11, 3, 5, 9, 12, 6, 1, 7, 2, 10], [7, 6, 5, 9, 10, 2, 3, 1, 4, 11, 8, 12]]

**minConflict:**

Execution Time: 0.884011154402

Consistency Checks: 1445

Solution: [[1, 9, 6, 7, 11, 8, 5, 4, 2, 10, 12, 3], [5, 2, 10, 11, 6, 12, 1, 3, 7, 8, 4, 9], [12, 8, 3, 4, 2, 7, 10, 9, 11, 1, 6, 5], [2, 7, 9, 1, 4, 10, 8, 5, 12, 6, 3, 11], [11, 12, 8, 10, 3, 1, 6, 2, 5, 9, 7, 4], [3, 5, 4, 6, 9, 11, 7, 12, 8, 2, 10, 1], [9, 10, 2, 12, 8, 3, 4, 11, 6, 5, 1, 7], [6, 3, 1, 8, 12, 5, 9, 7, 10, 4, 11, 2], [4, 11, 7, 5, 1, 6, 2, 10, 3, 12, 9, 8], [10, 1, 12, 2, 7, 4, 11, 8, 9, 3, 5, 6], [8, 4, 11, 3, 5, 9, 12, 6, 1, 7, 2, 10], [7, 6, 5, 9, 10, 2, 3, 1, 4, 11, 8, 12]]

**INPUT # 6 (Given in Sudoku.PDF)**

**Backtracking:**

Execution Time: 10.976893659

Consistency Checks: 404745

Solution: [[8, 11, 12, 1, 7, 10, 5, 2, 6, 4, 3, 9], [7, 10, 4, 2, 6, 8, 9, 3, 5, 12, 11, 1], [3, 6, 9, 5, 1, 12, 4, 11, 7, 2, 10, 8], [11, 4, 1, 12, 9, 5, 2, 10, 8, 7, 6, 3], [6, 8, 5, 9, 4, 3, 7, 12, 10, 1, 2, 11], [2, 3, 7, 10, 11, 6, 1, 8, 4, 9, 5, 12], [5, 12, 8, 6, 10, 2, 11, 4, 1, 3, 9, 7], [1, 9, 11, 3, 5, 7, 8, 6, 12, 10, 4, 2], [10, 7, 2, 4, 12, 1, 3, 9, 11, 6, 8, 5], [12, 5, 6, 8, 2, 9, 10, 7, 3, 11, 1, 4], [9, 1, 10, 11, 3, 4, 12, 5, 2, 8, 7, 6], [4, 2, 3, 7, 8, 11, 6, 1, 9, 5, 12, 10]]

**backtrackingMRV**:

Execution Time: 1.33694470138

Consistency Checks: 1507

Solution: [[8, 11, 12, 1, 7, 10, 5, 2, 6, 4, 3, 9], [7, 10, 4, 2, 6, 8, 9, 3, 5, 12, 11, 1], [3, 6, 9, 5, 1, 12, 4, 11, 7, 2, 10, 8], [11, 4, 1, 12, 9, 5, 2, 10, 8, 7, 6, 3], [6, 8, 5, 9, 4, 3, 7, 12, 10, 1, 2, 11], [2, 3, 7, 10, 11, 6, 1, 8, 4, 9, 5, 12], [5, 12, 8, 6, 10, 2, 11, 4, 1, 3, 9, 7], [1, 9, 11, 3, 5, 7, 8, 6, 12, 10, 4, 2], [10, 7, 2, 4, 12, 1, 3, 9, 11, 6, 8, 5], [12, 5, 6, 8, 2, 9, 10, 7, 3, 11, 1, 4], [9, 1, 10, 11, 3, 4, 12, 5, 2, 8, 7, 6], [4, 2, 3, 7, 8, 11, 6, 1, 9, 5, 12, 10]]

**backtrackingMRVfwd:**

Execution Time: 1.27041385123

Consistency Checks: 1407

Solution: [[8, 11, 12, 1, 7, 10, 5, 2, 6, 4, 3, 9], [7, 10, 4, 2, 6, 8, 9, 3, 5, 12, 11, 1], [3, 6, 9, 5, 1, 12, 4, 11, 7, 2, 10, 8], [11, 4, 1, 12, 9, 5, 2, 10, 8, 7, 6, 3], [6, 8, 5, 9, 4, 3, 7, 12, 10, 1, 2, 11], [2, 3, 7, 10, 11, 6, 1, 8, 4, 9, 5, 12], [5, 12, 8, 6, 10, 2, 11, 4, 1, 3, 9, 7], [1, 9, 11, 3, 5, 7, 8, 6, 12, 10, 4, 2], [10, 7, 2, 4, 12, 1, 3, 9, 11, 6, 8, 5], [12, 5, 6, 8, 2, 9, 10, 7, 3, 11, 1, 4], [9, 1, 10, 11, 3, 4, 12, 5, 2, 8, 7, 6], [4, 2, 3, 7, 8, 11, 6, 1, 9, 5, 12, 10]]

**backtrackingMRVcp:**

Execution Time: 1.40321982647

Consistency Checks: 1408

Solution: [[8, 11, 12, 1, 7, 10, 5, 2, 6, 4, 3, 9], [7, 10, 4, 2, 6, 8, 9, 3, 5, 12, 11, 1], [3, 6, 9, 5, 1, 12, 4, 11, 7, 2, 10, 8], [11, 4, 1, 12, 9, 5, 2, 10, 8, 7, 6, 3], [6, 8, 5, 9, 4, 3, 7, 12, 10, 1, 2, 11], [2, 3, 7, 10, 11, 6, 1, 8, 4, 9, 5, 12], [5, 12, 8, 6, 10, 2, 11, 4, 1, 3, 9, 7], [1, 9, 11, 3, 5, 7, 8, 6, 12, 10, 4, 2], [10, 7, 2, 4, 12, 1, 3, 9, 11, 6, 8, 5], [12, 5, 6, 8, 2, 9, 10, 7, 3, 11, 1, 4], [9, 1, 10, 11, 3, 4, 12, 5, 2, 8, 7, 6], [4, 2, 3, 7, 8, 11, 6, 1, 9, 5, 12, 10]]