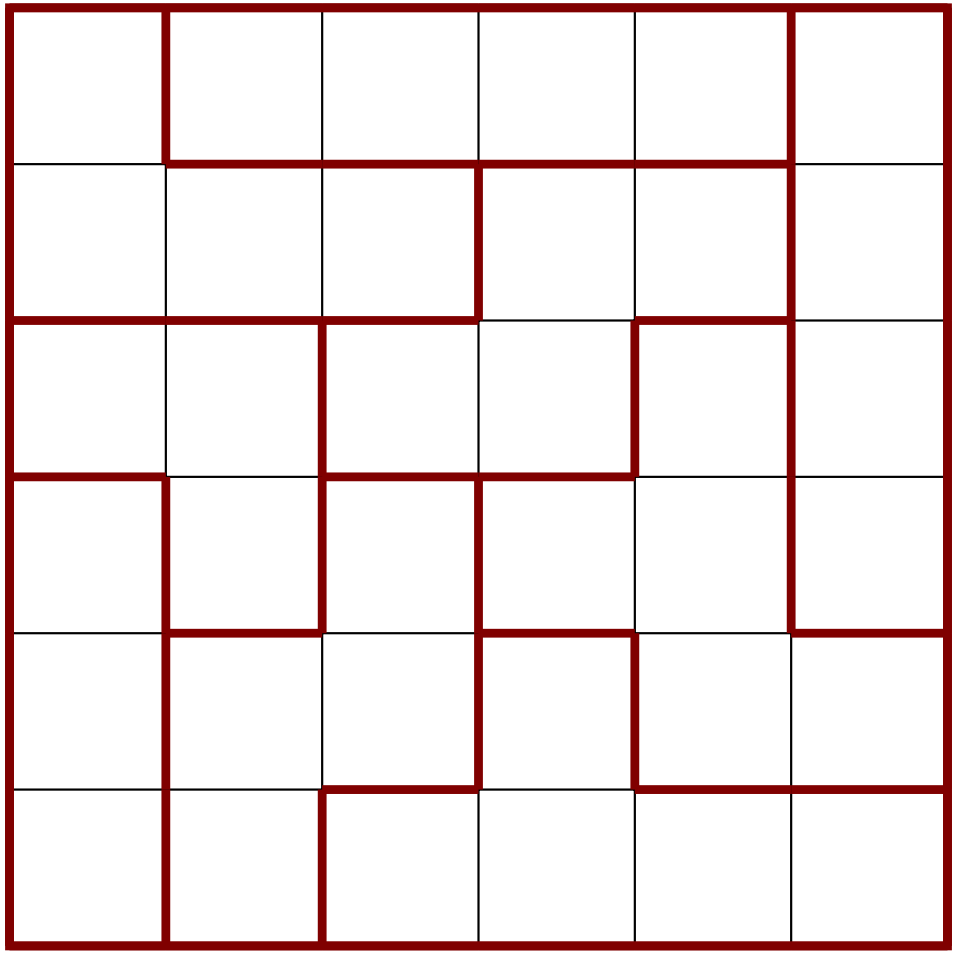
ROC'DEZ Challenge

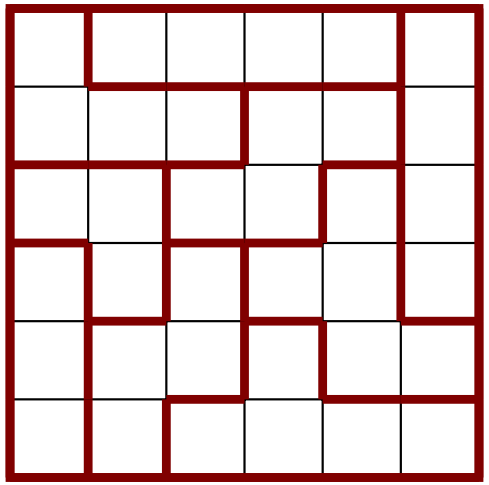
LAAS-CNRS

Toulouse

Clueless Puzzles



Rules

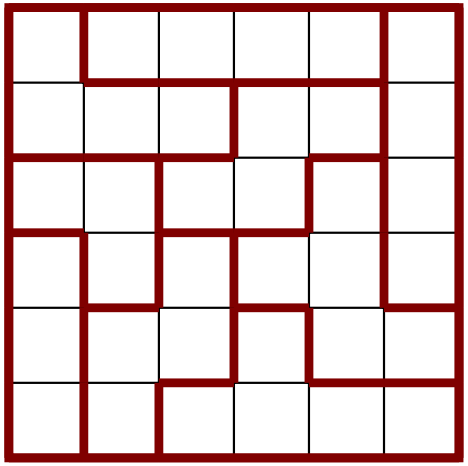


- Solve the puzzle!
- Like the name suggests... no clues!

Questions?

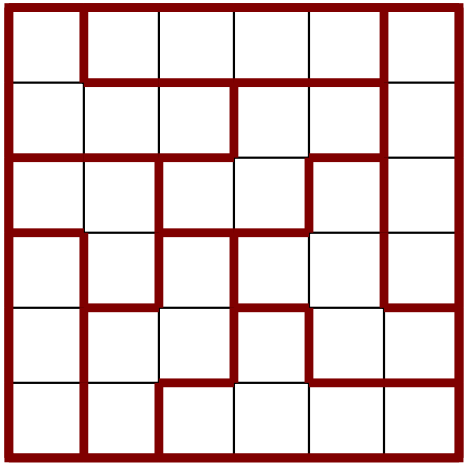
Alright, alright...

- Note by [Gerard Butters, Frederick Henle, James Henle, Colleen McGaughey (Smith College, Northampton)]
  - Sudoku puzzles are "inelegant": numeric clues
  - Can we have a puzzle without any clue?
  - Yes! (sort of)



Alright, alright...

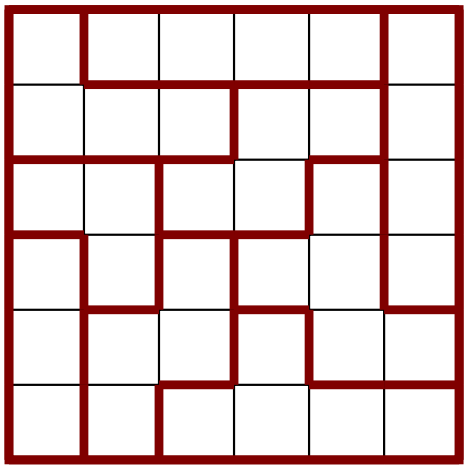
- Note by [Gerard Butters, Frederick Henle, James Henle, Colleen McGaughey (Smith College, Northampton)]
  - Sudoku puzzles are "inelegant": numeric clues
  - Can we have a puzzle without any clue?
  - Yes! (sort of)



Alright, alright...

- Note by [Gerard Butters, Frederick Henle, James Henle, Colleen McGaughey (Smith College, Northampton)]
  - Sudoku puzzles are "inelegant": numeric clues
  - Can we have a puzzle without any clue?
  - Yes! (sort of)

4	1	3	2	5	6
1	2	6	5	3	4
6	3	5	4	2	1
5	6	4	3	1	2
2	5	1	6	4	3
3	4	2	1	6	5

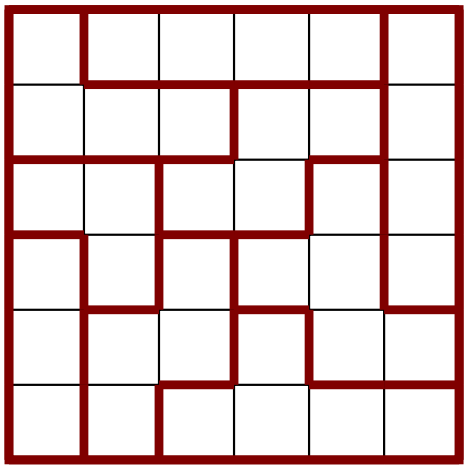


- Latin square (rows and columns are permutations of $\{1, \ldots, N\}$)

Alright, alright...

- Note by [Gerard Butters, Frederick Henle, James Henle, Colleen McGaughey (Smith College, Northampton)]
  - Sudoku puzzles are "inelegant": numeric clues
  - Can we have a puzzle without any clue?
  - Yes! (sort of)

4	1	3	2	5	6
1	2	6	5	3	4
6	3	5	4	2	1
5	6	4	3	1	2
2	5	1	6	4	3
3	4	2	1	6	5



- Latin square (rows and columns are permutations of $\{1, \ldots, N\}$)
- Specified regions all add up to the same value (14)

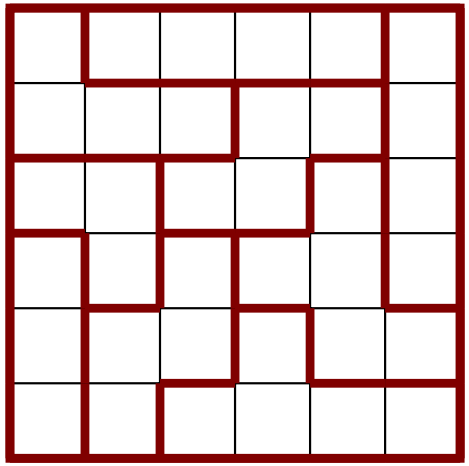
Alright, alright...

- Note by [Gerard Butters, Frederick Henle, James Henle, Colleen McGaughey (Smith College, Northampton)]
  - Sudoku puzzles are "inelegant": numeric clues
  - Can we have a puzzle without any clue?
  - Yes! (sort of)

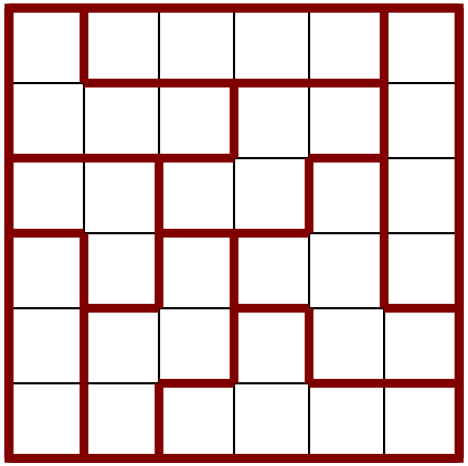
2	1	3	4	6	5
1	5	6	3	4	2
4	6	2	5	3	1
3	4	5	2	1	6
6	2	4	1	5	3
5	3	1	6	2	4

- Latin square (rows and columns are permutations of $\{1, \ldots, N\}$)
- Specified regions all add up to the same value (14)

Solving a puzzle by hand

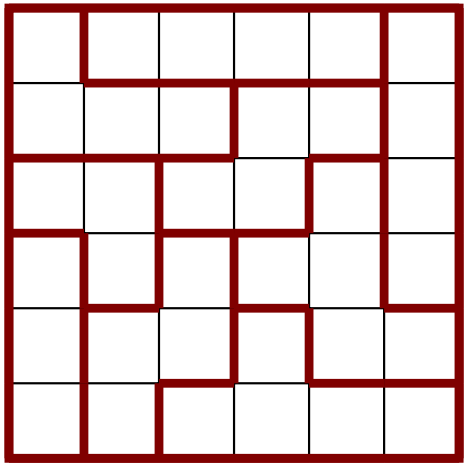


Solving a puzzle by hand



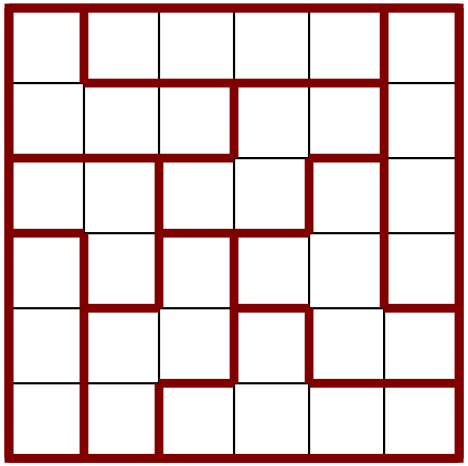
- Sum in each of the M regions $V = N * (N * (N + 1)/2)/M$

Solving a puzzle by hand



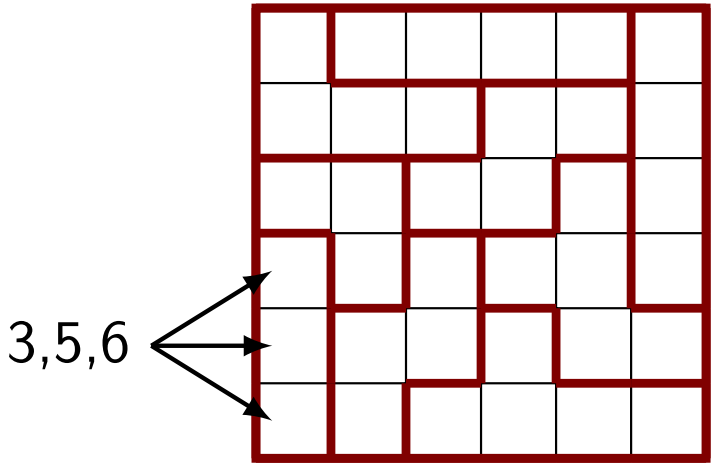
- Sum in each of the M regions $V = N * (N * (N + 1)/2)/M$
  - $N = 6, M = 9$ then $V = 14$

Solving a puzzle by hand



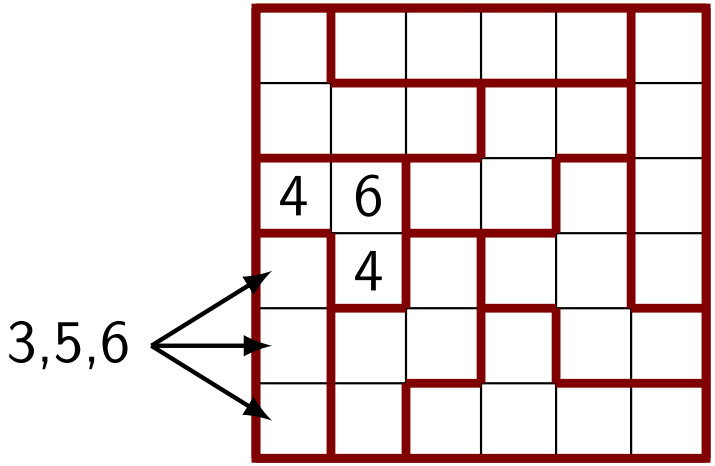
- Sum in each of the M regions $V = N * (N * (N + 1)/2)/M$
  - $N = 6, M = 9$ then $V = 14$
- 3-partitions of 14 in $\{1, \ldots, 6\}$:
  - (2, 6, 6)
  - (4, 4, 6)
  - (4, 5, 5)
  - (3, 5, 6)

Solving a puzzle by hand



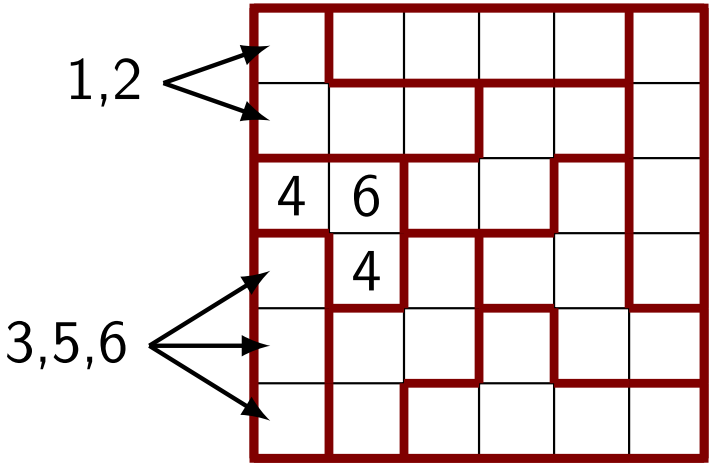
- Sum in each of the M regions $V = N * (N * (N + 1)/2)/M$
  - $N = 6, M = 9$ then $V = 14$
- 3-partitions of 14 in $\{1, \ldots, 6\}$:
  - (2, 6, 6)
  - (4, 4, 6)
  - (4, 5, 5)
  - (3, 5, 6) only partition consistant with a row/column

Solving a puzzle by hand



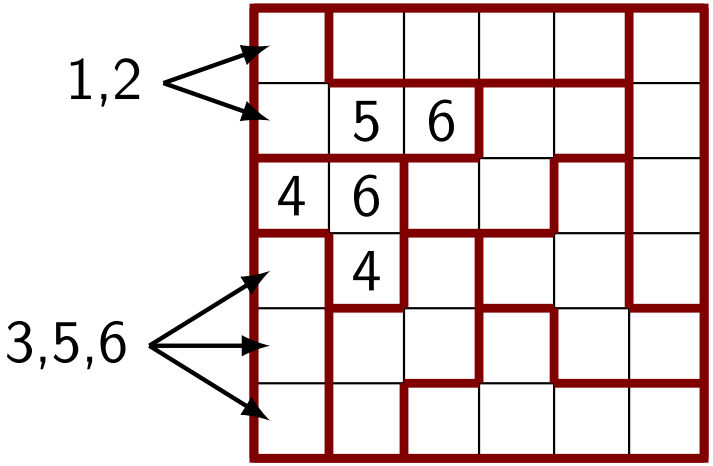
- Sum in each of the M regions $V = N * (N * (N + 1)/2)/M$
  - $N = 6, M = 9$ then $V = 14$
- 3-partitions of 14 in $\{1, \ldots, 6\}$:
  - (2, 6, 6)
  - (4, 4, 6)
  - (4, 5, 5)
  - (3, 5, 6) only partition consistant with a row/column

Solving a puzzle by hand



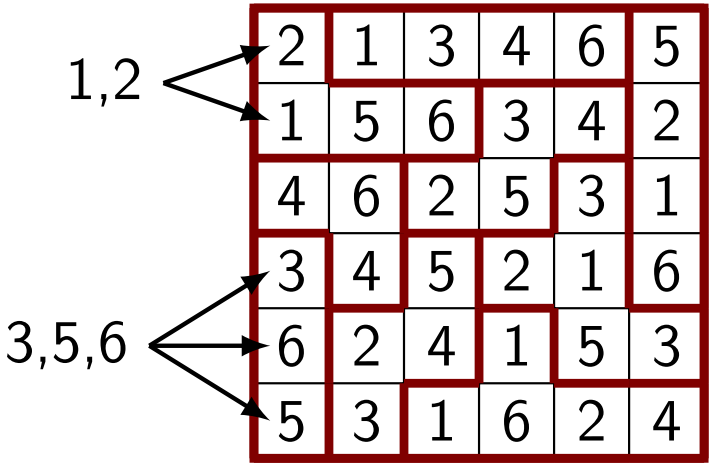
- Sum in each of the M regions $V = N * (N * (N + 1)/2)/M$
  - $N = 6, M = 9$ then $V = 14$
- 3-partitions of 14 in $\{1, \ldots, 6\}$:
  - (2, 6, 6)
  - (4, 4, 6)
  - (4, 5, 5)
  - (3, 5, 6) only partition consistant with a row/column

Solving a puzzle by hand



- Sum in each of the M regions $V = N * (N * (N + 1)/2)/M$
  - $N = 6, M = 9$ then $V = 14$
- 3-partitions of 14 in $\{1, \ldots, 6\}$:
  - (2, 6, 6)
  - (4, 4, 6)
  - (4, 5, 5)
  - (3, 5, 6) only partition consistant with a row/column

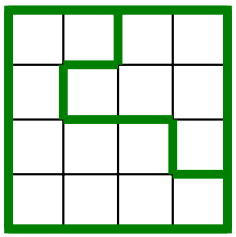
Solving a puzzle by hand



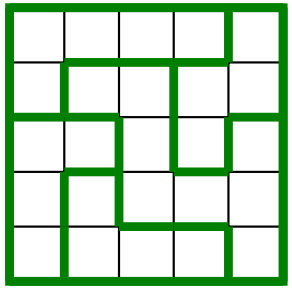
- Sum in each of the M regions $V = N * (N * (N + 1)/2)/M$
  - $N = 6, M = 9$ then $V = 14$
- 3-partitions of 14 in $\{1, \ldots, 6\}$:
  - (2, 6, 6)
  - (4, 4, 6)
  - (4, 5, 5)
  - (3, 5, 6) only partition consistant with a row/column

Challenge

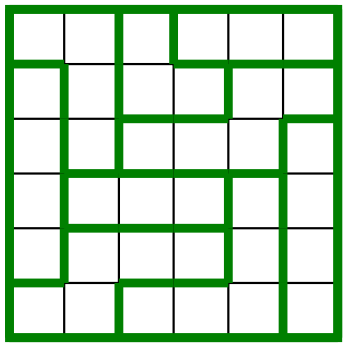
- 1-st Part: Solve these puzzles:



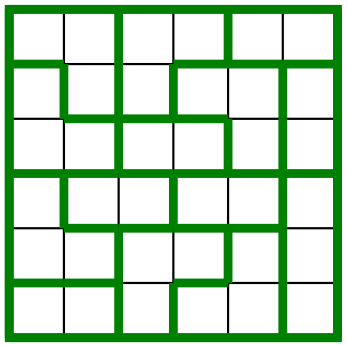
gen_4_2_1



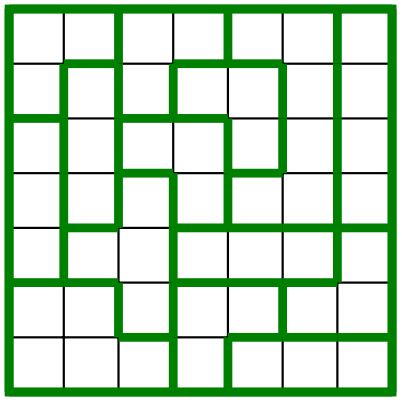
gen_5_5_1



gen_6_9_1



gen_6_14_1



gen_7_14_1

- 2-nd Part: Create new puzzles for $N \geq 4$ and $1 < M < N(N+1)/2$
  - Criterion: number of solutions (a true puzzle has only one solution)

Rules (1-st part)

- Puzzles on paper and this format:

$\langle$size of the board$\rangle$
$\langle$number of regions$\rangle$

1	1	1	1
1	2	2	2
1	1	2	2
1	1	1	2

- Solutions by USB stick in this format:

$\langle$id of the team$\rangle$
$\langle$puzzle name$\rangle$

1	2	3	4
3	1	4	2
1	4	2	3
2	3	1	4

- Score: $N * (\lfloor 10/k \rfloor)$ for the k-th team to solve puzzle of size N
  - You can run
  - But you can't tackle or trip someone up (arbitrary point deduction)
  - -1 for a wrong solution

Rules (2-nd part)

- Puzzle in the same format as in the data [file name = team id]
- For each class of puzzle (`size` $\times$ `number of regions`):
  - Count only the best (i.e., with fewest solutions) for this class
  - Score: $\lfloor 100 \cdot 2^N / k \rfloor$ for a puzzle of size N with k solutions
    - ★ -10 for a puzzle with no solution or equal to those in the data

After lunch

- Archive `rocdez.zip` with
  - The original note from Northampton folks
  - Data set (`rocdez/data`) [note + challenge instances]
  - A `Puzzle` class (in C++ and in Python)
    - ★ read/write instances and solutions + a few other things
- You can use whatever method to compute solutions
  - By hand: perhaps the best way to solve the easiest instances first
  - Standard MIP/CP models: should easily solve all puzzles
  - Generating new puzzles is much harder
    - ★ Generating feasible puzzles is hard
    - ★ Generating nice puzzles is extremely hard

- Postgrads
  - **d1** Grégoire, Jean-Thomas, Leonardo
  - **d2** Azzedine, Margaux, Mohamed
  - **d3** Clément, Letitia, Yacine
- Staff
  - **p1** Cyril, Julien, Sandra
  - **p2** Nicolas, Pierre
  - **p3** Christian, Laurent
- Interns (coached by Marie-Jo, Patrick and Nadia)
  - **s1** Marie-Jo, Alexandre, Florian, Francisco, Maria, Thomas
  - **s2** Patrick, Nadia, Mikael, Pierre, Pauline, Ulrich

Bon Appétit!

Well, for those who do not fast

Ready...

- Puzzle class (C++ and Python):
  - Read puzzle from file
  - Get the value of the sums
  - Get the list of shapes coordinates
  - Check a solution
  - Apply symmetries
    - clock-wise rotation
    - vertical mirror
    - horizontal mirror
    - flip along first/second diagonal
  - Compare two puzzles

Steady...

- Some relevant tools (`clueless_puzzle.py`):
  - `-factors` ⟨`n`⟩
    - ★ all pairs Number of regions $\times$ Sum in a n-puzzle
  - `-partitions` ⟨`s`⟩, ⟨`k`⟩, ⟨`n`⟩, [`l`]
    - ★ all k-partitions of s using numbers up to n on at most l
  - `-ascii/tikz` ⟨`file`⟩
    - ★ print the puzzle nicely
  - `-compare` ⟨`file1`⟩, ⟨`file2`⟩
    - ★ check if file1, file2 are symmetric puzzles
  - `-verify` ⟨`solfile`⟩, ⟨`puzzlefile`⟩
    - ★ check that the solution is correct

Go!