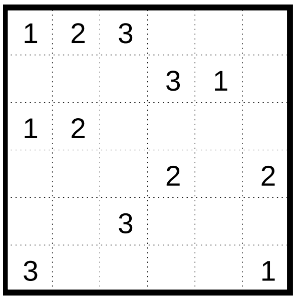
SLASH PACK

This is the penultimate puzzle in Alex Bellos' *Puzzle Ninja* book, and for me is the most remarkable invention. It has the beauty of simple rules and an uncluttered diagram, small enough to redraw when you are flummoxed, but asking you to discover new skills with a different way of thinking. This has presented me with the toughest challenge, the necessity of guaranteeing puzzles with unique solutions, but I have persevered with the knowledge that Krystal will get these to you via *Locallink*. Alex is happy for me to quote details, and I will make use of his introductory example.

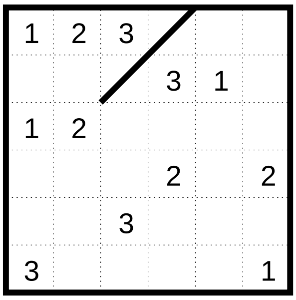
The challenge: Divide the grid into regionsthat include each of the given numbers only once, drawing dividing lines diagonally across unnumbered cells.

The rules: Each cell can contain only one diagonal line. Some cells may be left completely empty.

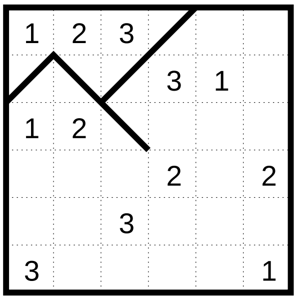
You start with a figure like A and look for some obvious slashes. As no final region can have more than one occurrence of each digit, you immediately see that two [3]s must be separated by two slashes as in figure B.



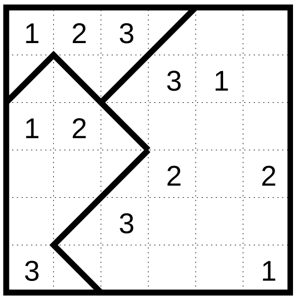
A



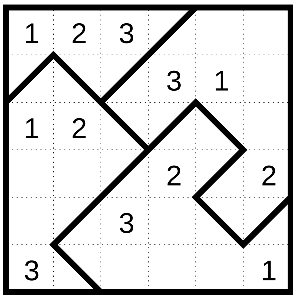
B



C



D



E

As [1], [2] and [3] on the top row cannot be separated, they must be in the same region, and as this region cannot contain any other 'givens' there must be a boundary between them and the [1] and [2] in the third row. Figure C shows the only possible such boundary, and one other diagonal has been added. This 'back-slash' divides the region which must contain the [1] and the [3] on the second row from the region which must contain the [1] and [2] on the third row, demanded by the fact that both these regions contain a [1].

Now you must decide which [1] can be in the region which will have the [3] in the bottom left corner. There are two to consider. You will immediately see that the [1] in the bottom right corner could only exist in a region with this [3] if there were a division below the [3] on the fifth row, but as all dividers are diagonals any divider attempted must intersect the boundary. Therefore this [3] must connect with the [1] in the column above, and this forces the dividers added in figure D.

The puzzle is now completed with the dividers added in figure E. There is only one way to add these dividers, so the solution must therefore be unique.

The solutions I have given have the regions represented with different colours. During the solution process I think of chromosomes and genes, and for this introductory example there are 4 chromosomes (regions) each eventually with 3 genes. When considering whether it is right to separate two growing regions, the only thing that can be definite is when the partial chromosomes have non-matching copies of the same gene. Nevertheless, when the solution is nearly achieved, gene selection often becomes Hobson's choice. Other puzzle solvers may well decide on a different mental construct here, but the need for such a construct is just one of the new challenges that this fascinating puzzle presents.

SLASH PACK

3		3		3	
					3
2			2		
1			1		2
	2				
			1		1

1

	2		3		
1				3	
		1			3
1		2			1
			2		
	3			2	

2

1		3	2		1		3
	2			1			
2						3	
		3		1			
	2				1		
							3
3		2		2	1		

3

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

4

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

5

		1			2	
	1					
3		2		3		3
				1		
1		2				3
					1	
	2			3		2

6

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

7

		2			1		3
						1	
		2					1
3			2		2		
							2
			3				
1		1		3		3	

8

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

9

	3		3		2	
2				1		
	1					
3				2		3
		2				
2					1	3
		1		1		

10

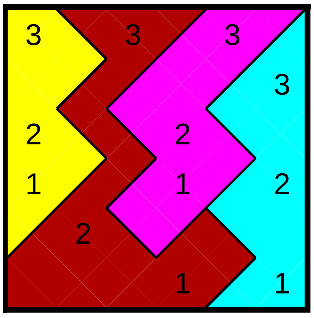
	1		2		2		1	
		3						
4						1		1
		4		1				
4		3		2				2
						2		
2		3		3		4		4
1		4		3		3		

11

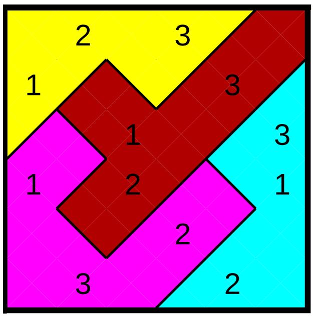
1		1		3	
	1		2		
	3		3		2
				2	

12

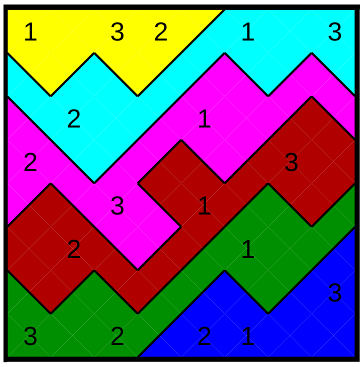
SLASH PACK



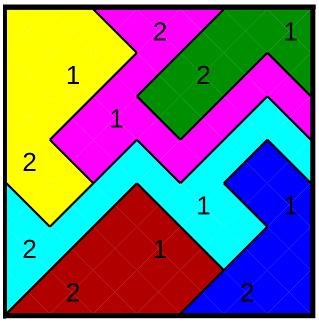
1



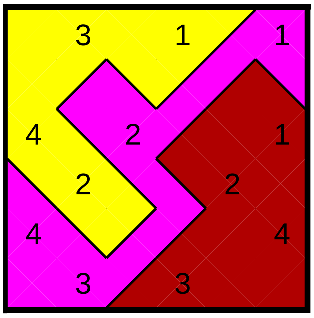
2



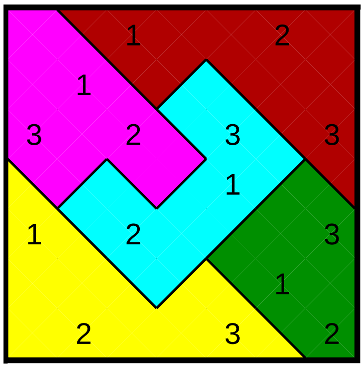
3



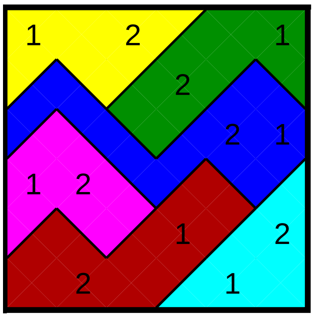
4



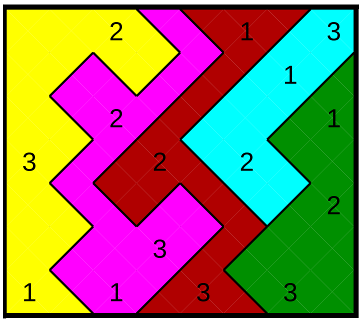
5



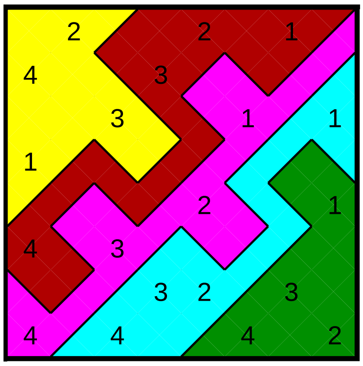
6



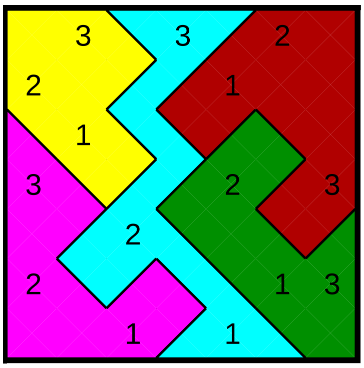
7



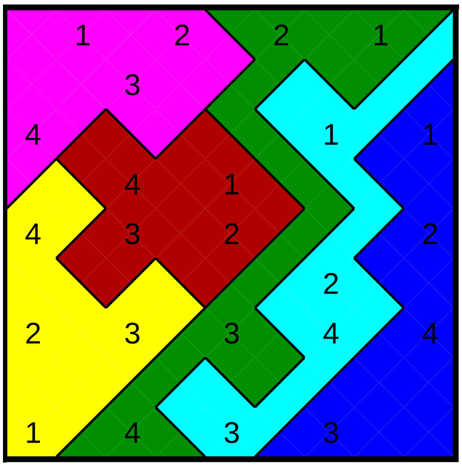
8



9

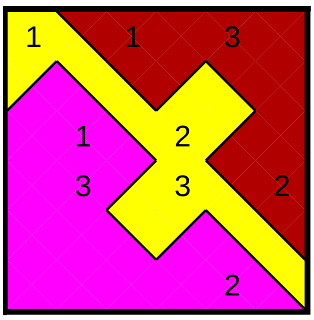


10



11

12



Solutions

7×7

	3			1		2
		3				
			1		2	
3						
	1	2			2	1
1			2		3	
			3			

1

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

2

				2		2
		1		3		
2	3					
		2		1		3
3	1					
			2	3		
1					1	

3

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

4

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

5

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

6

7×7

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

7

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

8

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

9

1			1		3	
	2					1
3		2		2		
						3
3		3				
1		1		1		2
	2		2		3	

10

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

11

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

12

8×8

	1				3		
1				3			
		2					
				2		1	
2	3		2				
				1		1	3
2		3					
	1		3		2		

1

	1		1		1		3
3							
		2			2		3
3			2			2	
3		3		2	1		
			2		1		1

2

					3		
				3			
		1					
			1		3		
							3
	2		2			1	
				2			
1						2	

3

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

4

	2		2				
					2		3
3		1					
					1	3	
	1						3
		1			1		
3							
			2		2		

5

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

6

8×8

				3		3	
		2					
1				2		1	
		1					2
2			2		3		
	1						3
			2		1		
	3		3				1

7

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

8

3		2		2			
						1	
			1		3		3
1	2						
			3		1		2
1		3		2		3	
				2			1

9

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

10

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

11

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

12

9×9

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

1

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

2

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

3

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

4

10×10

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

1

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

2

10×10

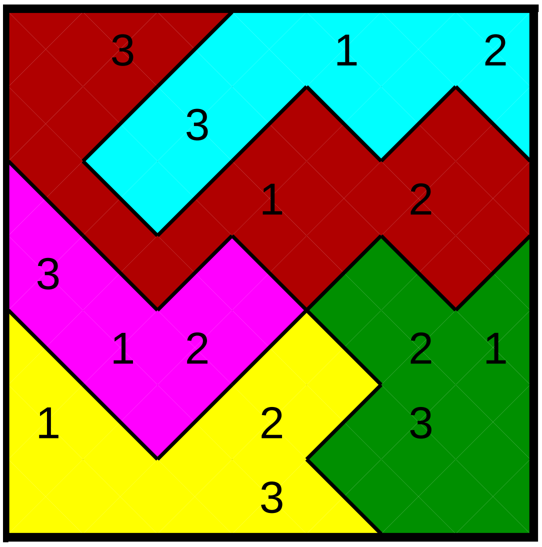
2		2							
									1
4		4	3		4		1		
						3			
		3		4					3
3						2			
		2		2					
								4	
			1						
					1		1		

3

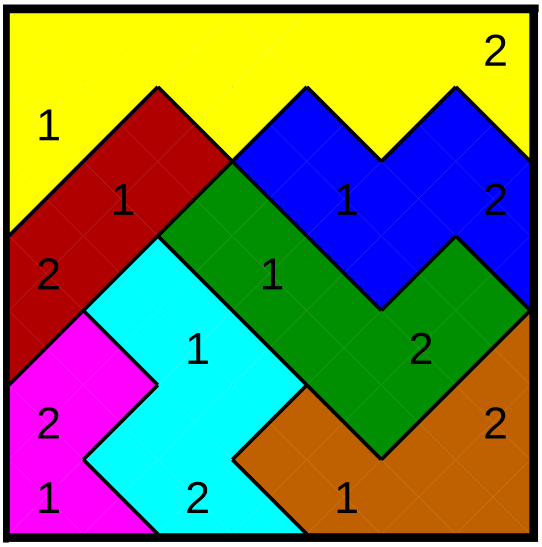
1				2		4			4
		3							
	2					4		2	
					1				
3			2		3				
		4					3		
4						2		3	
	2		3						
				4					1
				1		1		1	

4

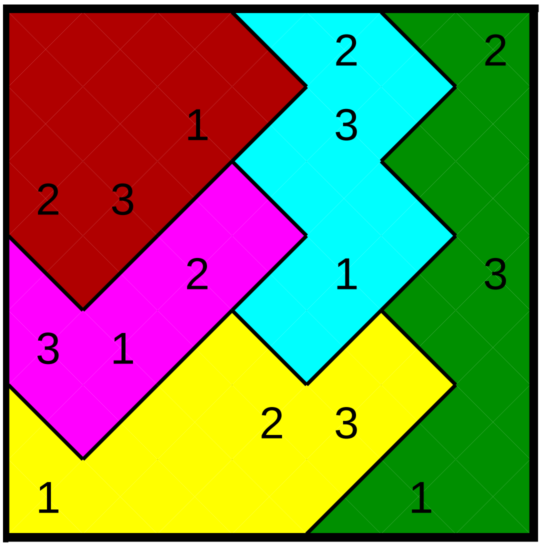
7×7 **solutions**



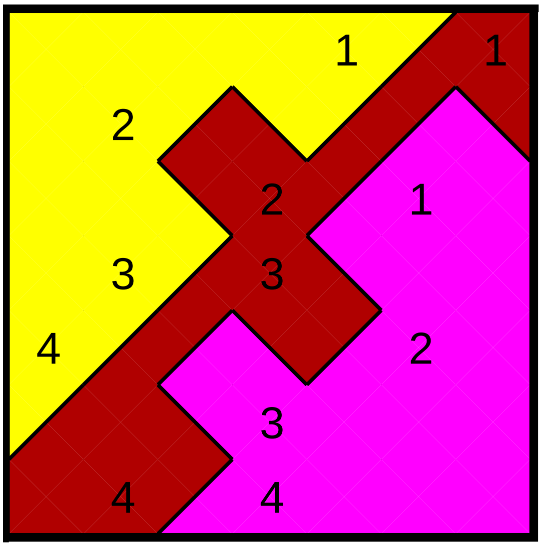
1



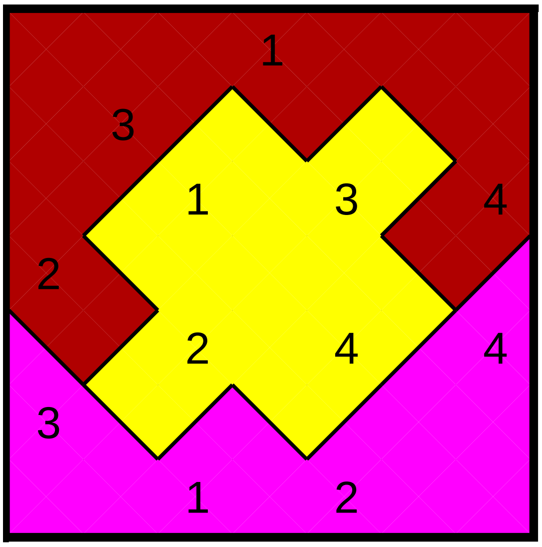
2



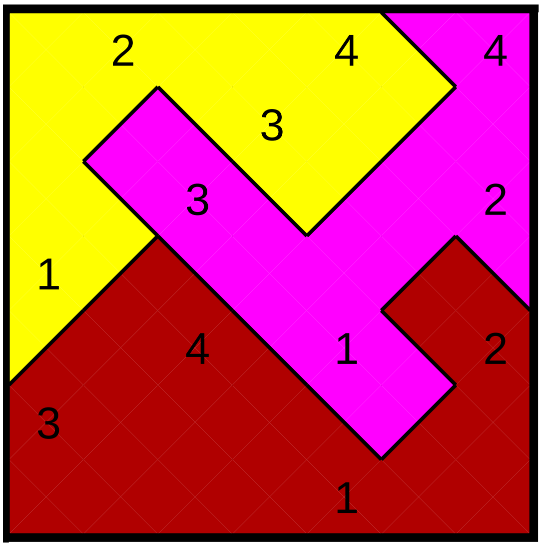
3



4

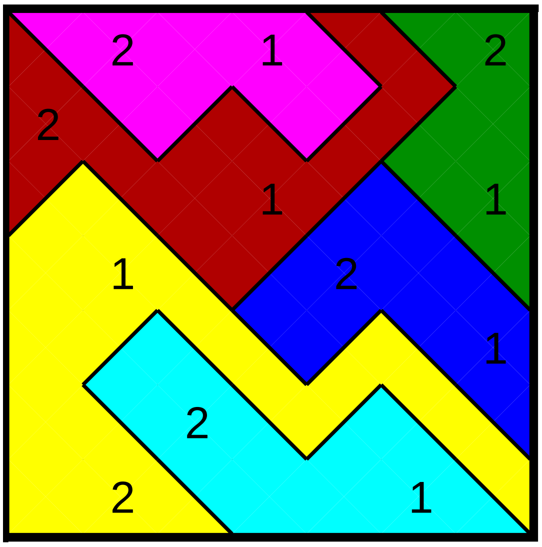


5

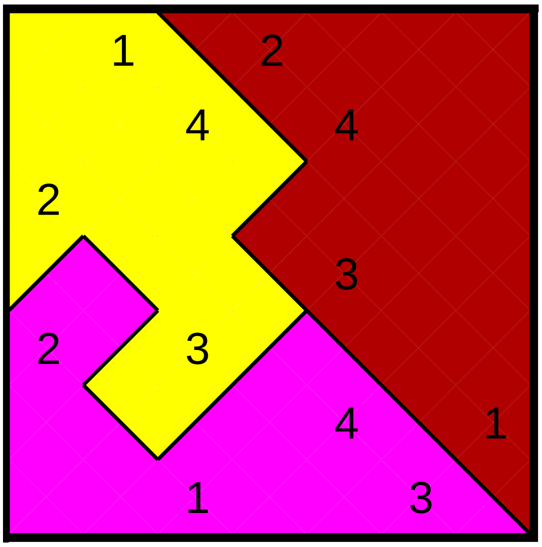


6

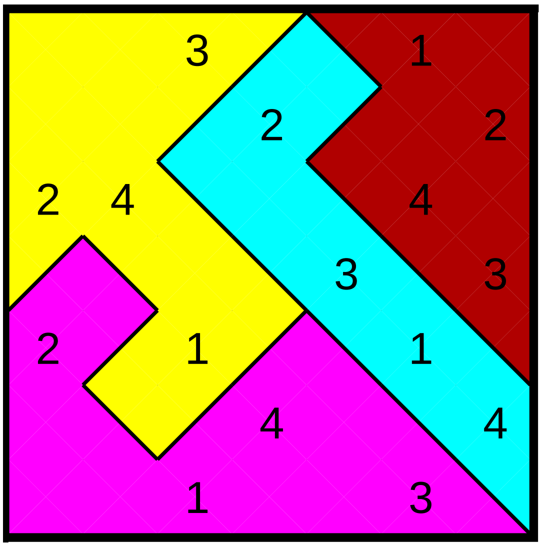
7×7 **solutions**



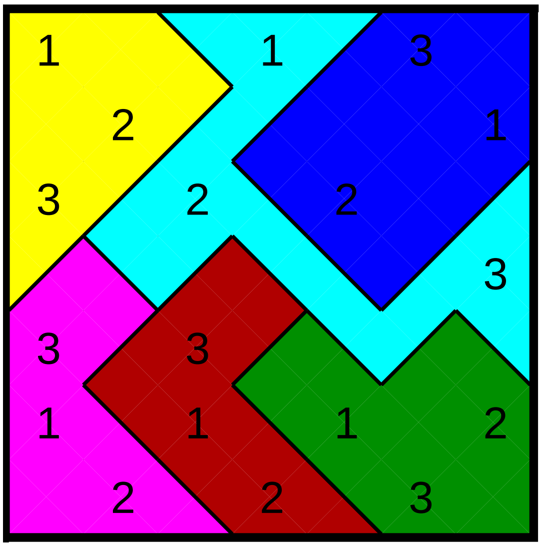
7



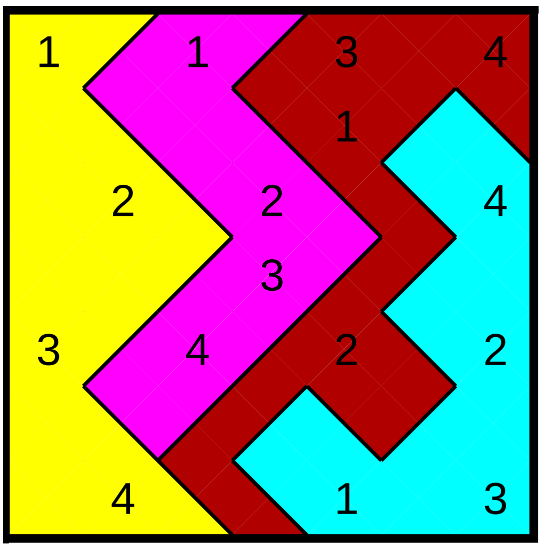
8



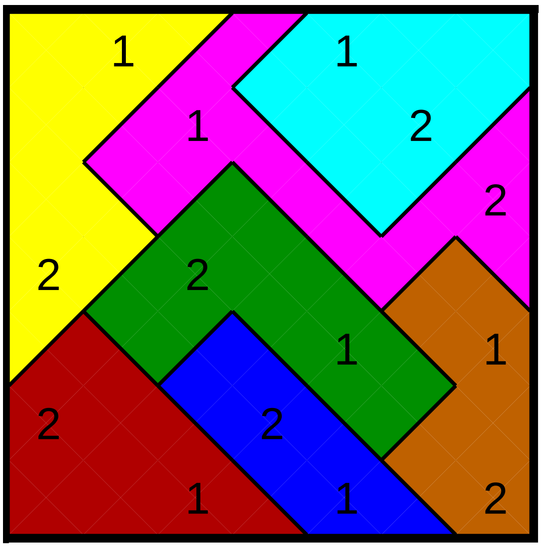
9



10

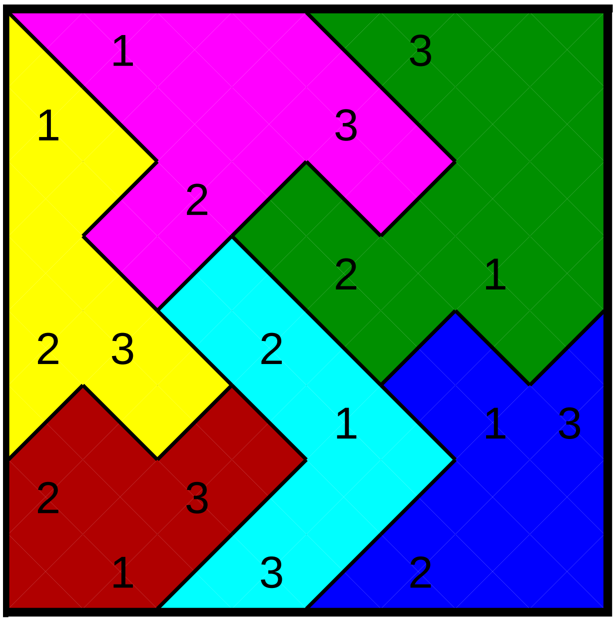


11

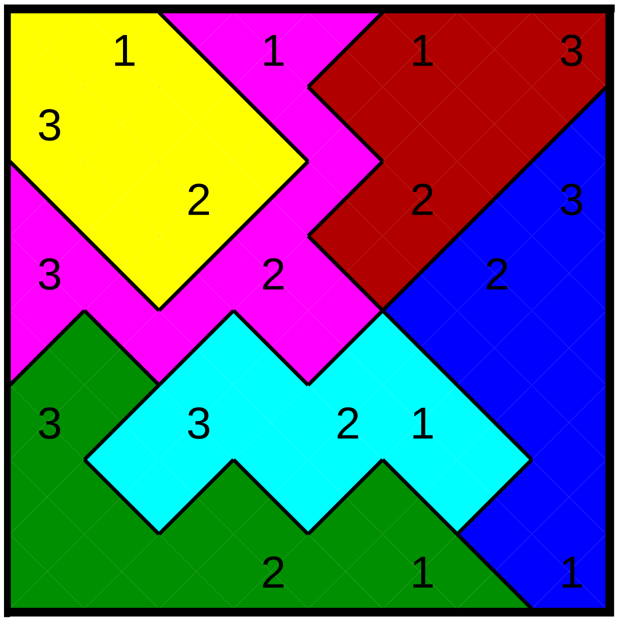


12

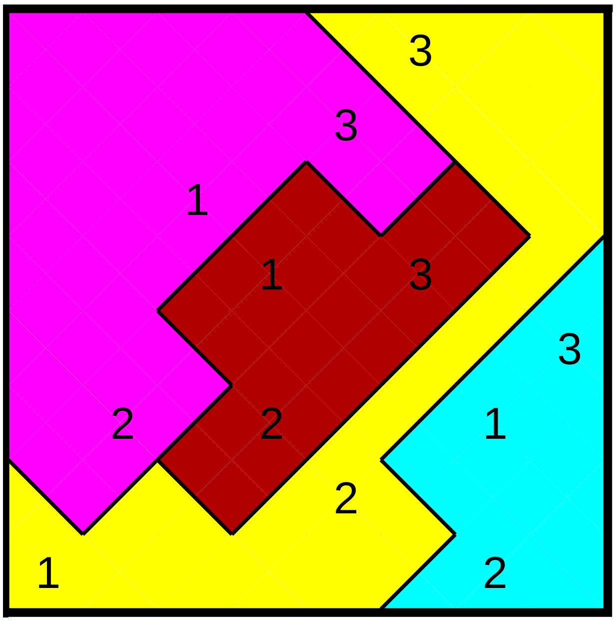
8×8 **solutions**



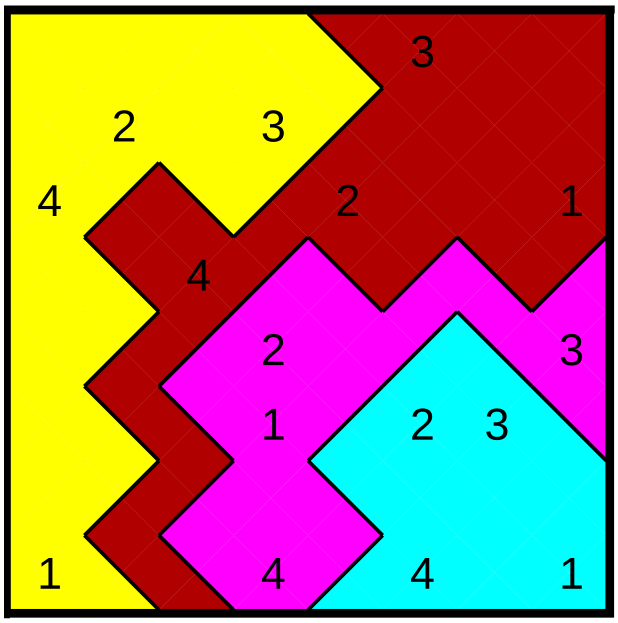
1



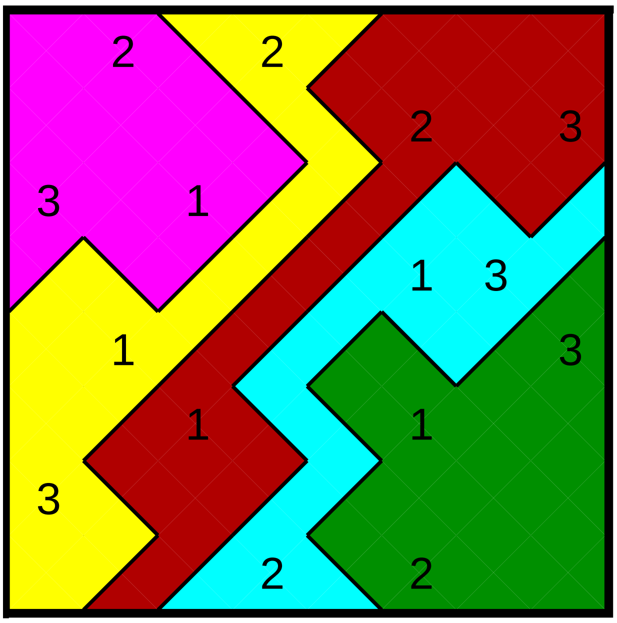
2



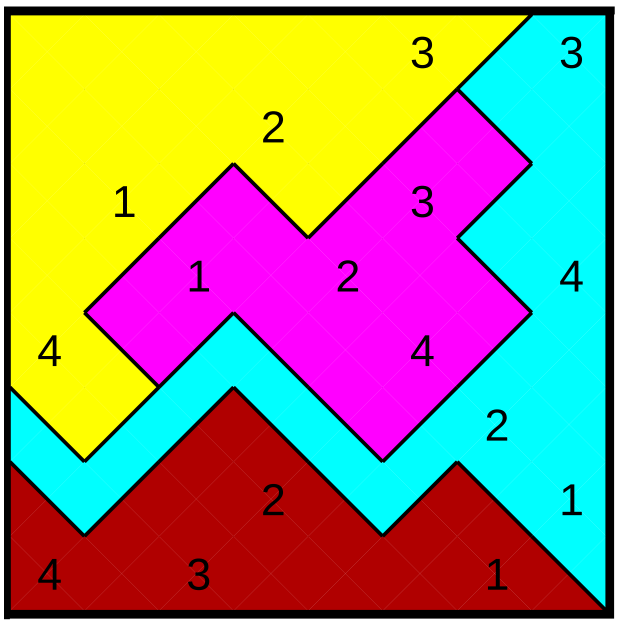
3



4

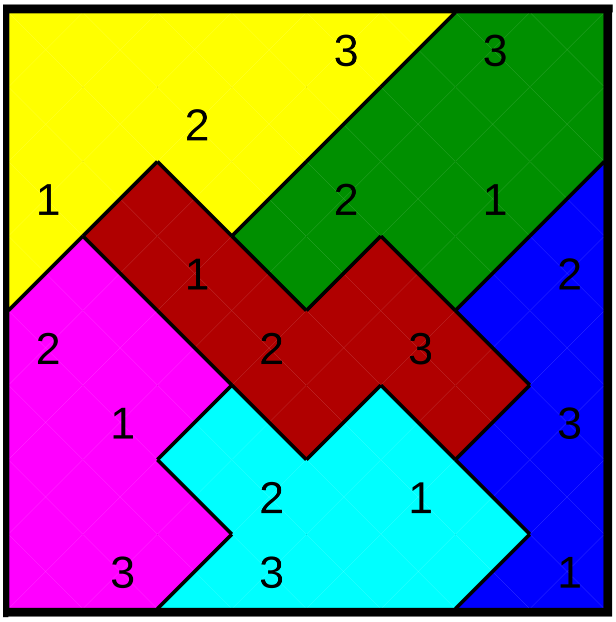


5

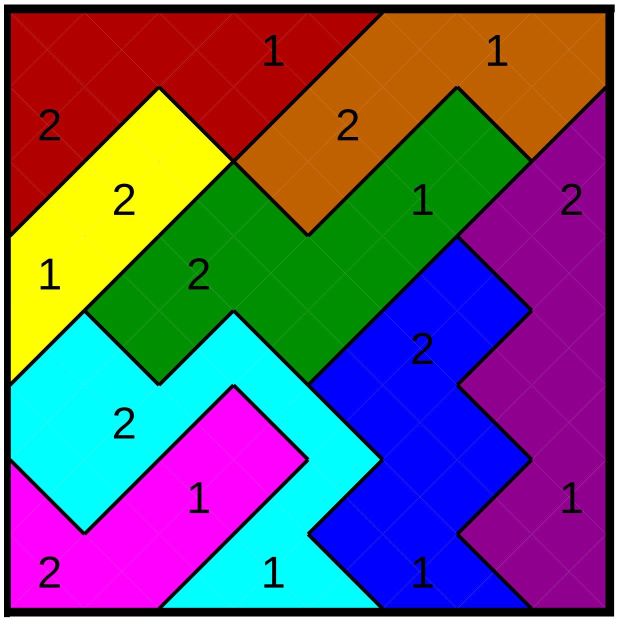


6

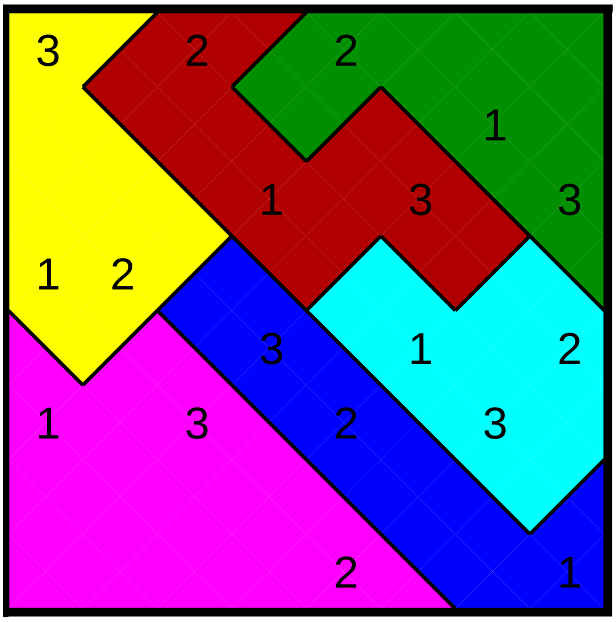
8×8 **solutions**



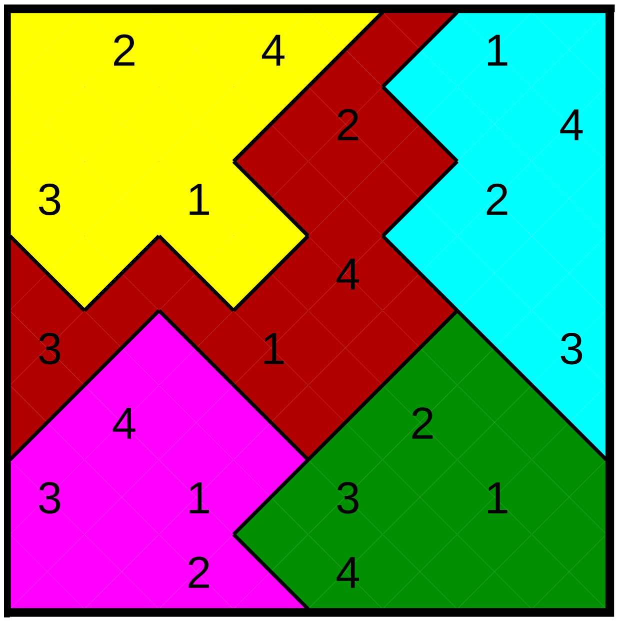
7



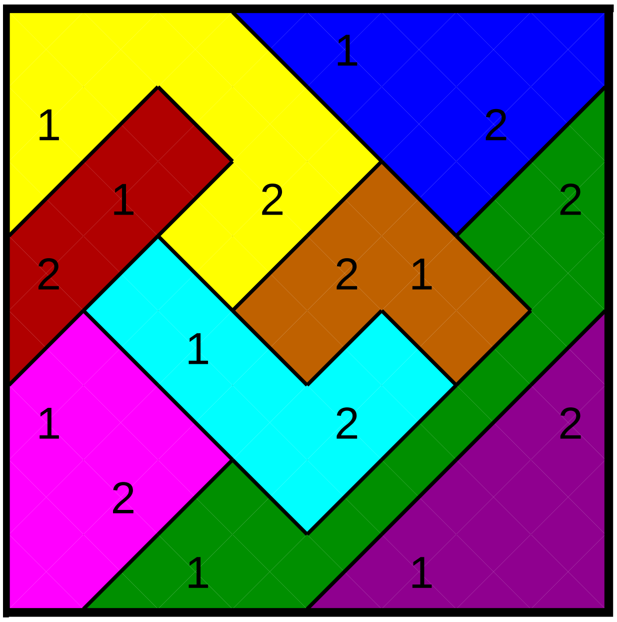
8



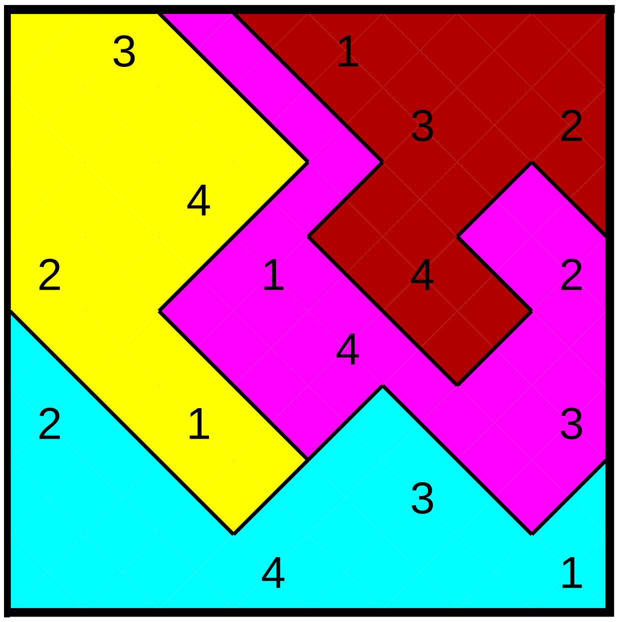
9



10

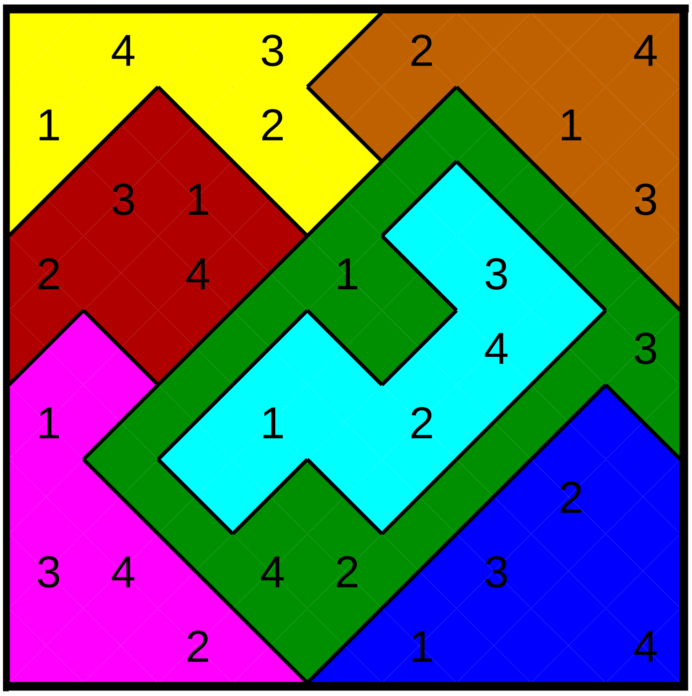


11

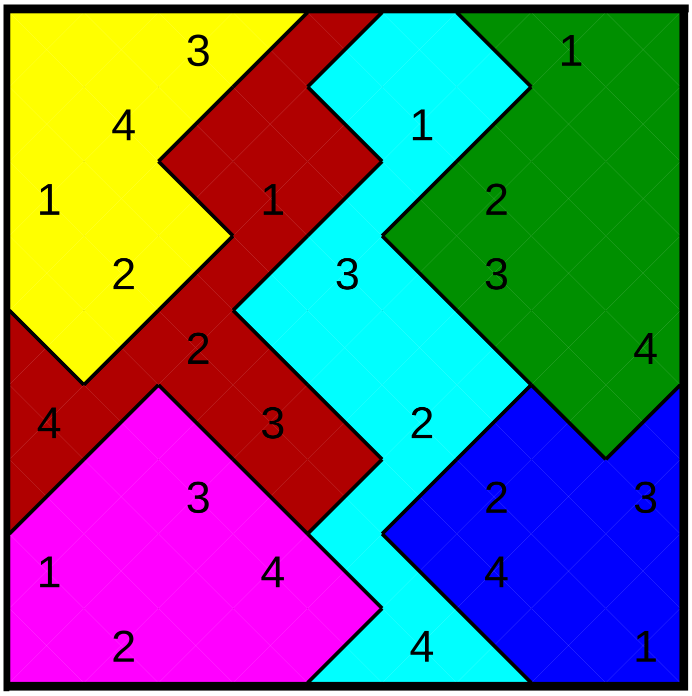


12

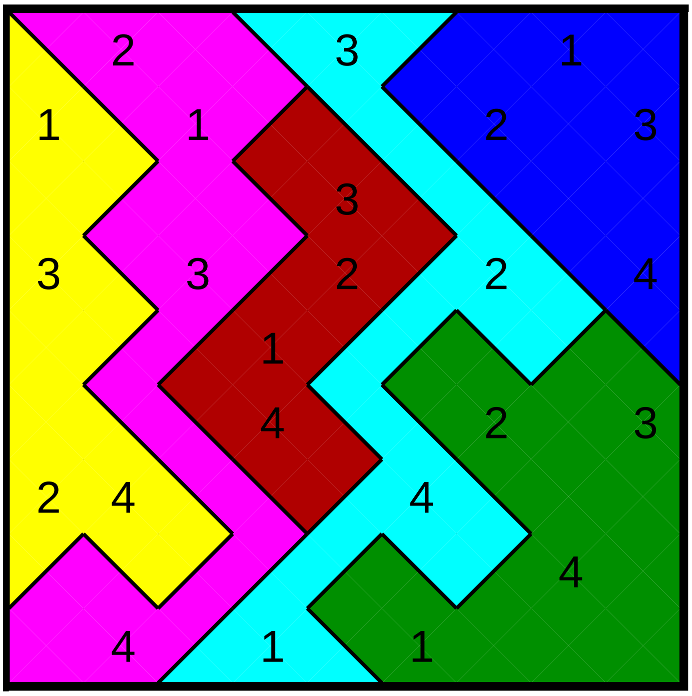
9×9 **solutions**



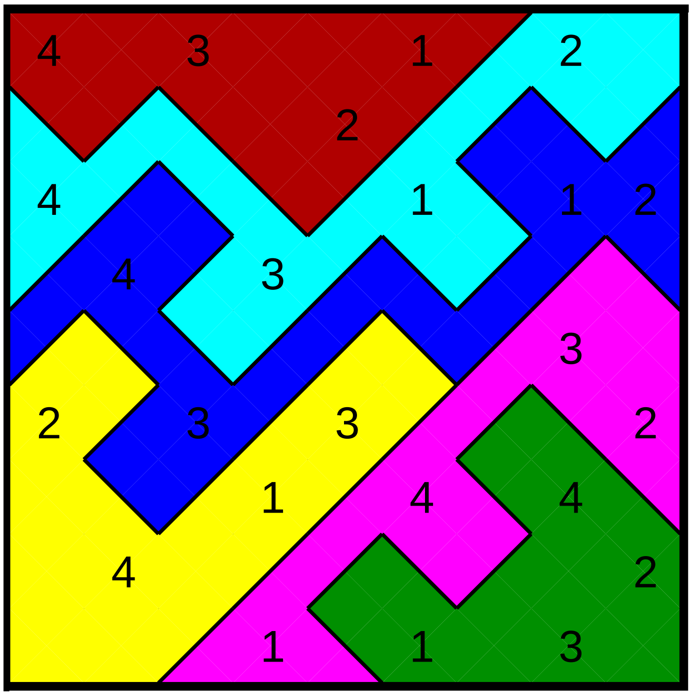
1



2

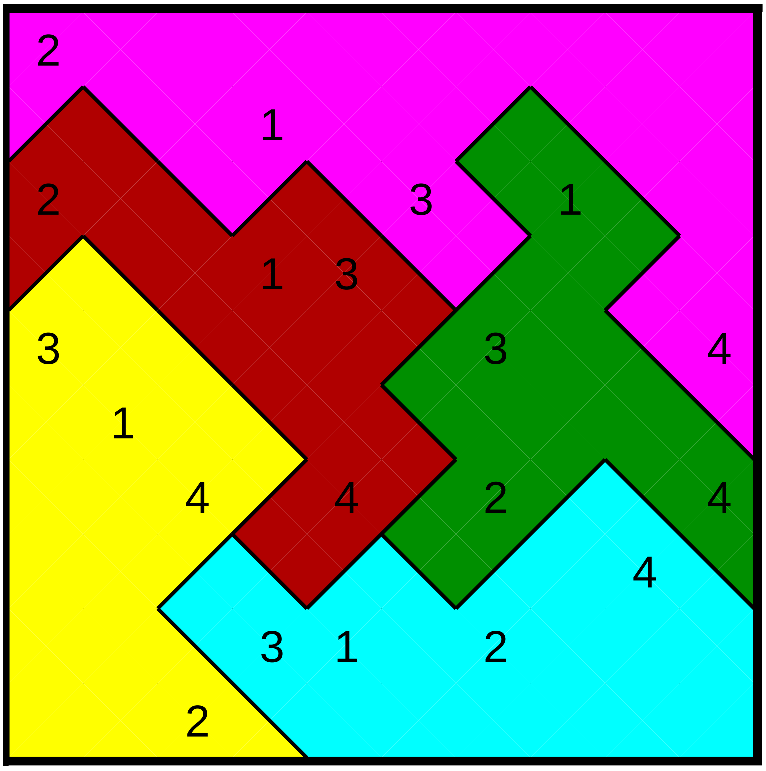


3

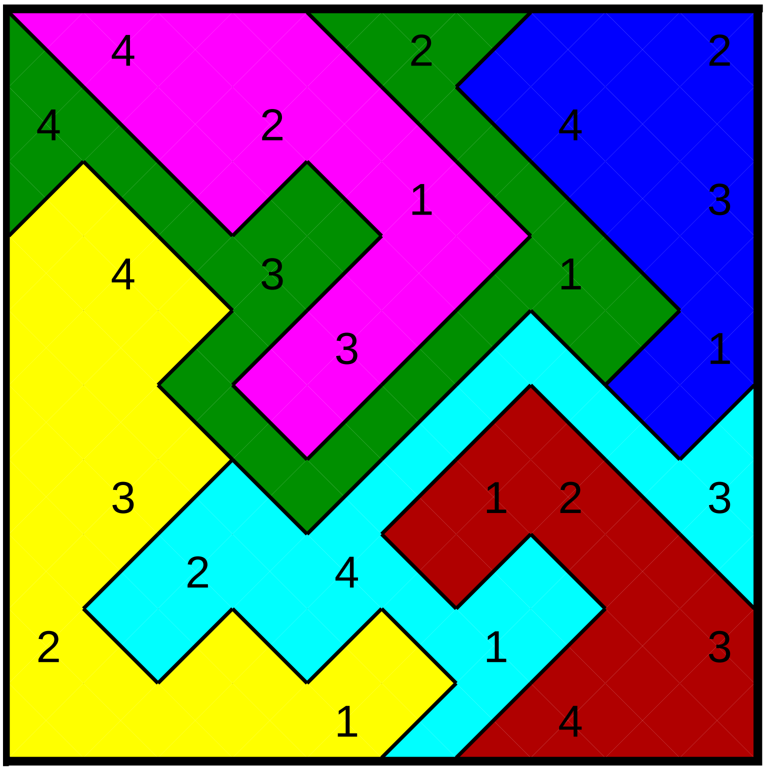


4

10×10 **solutions**

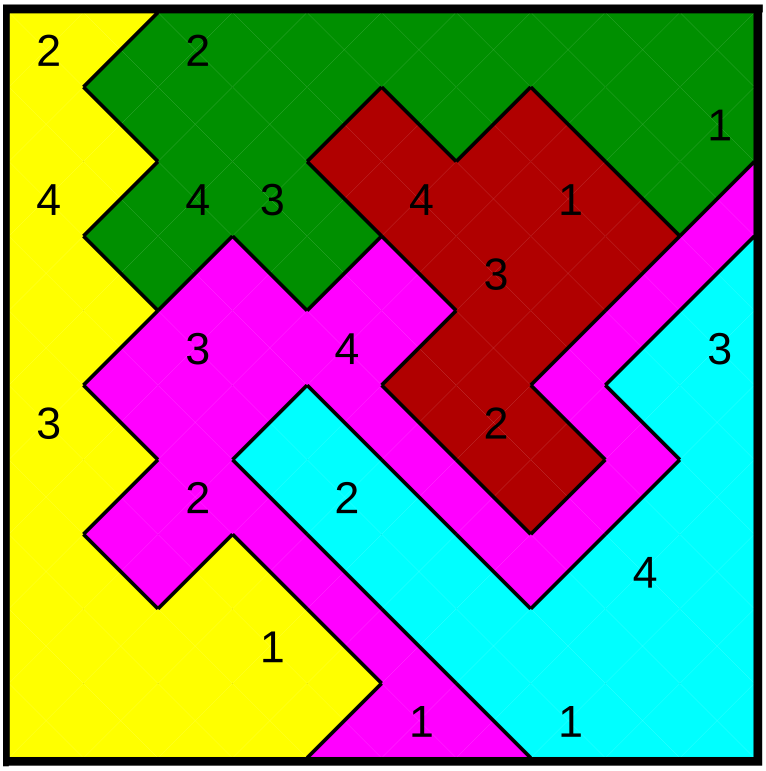


1

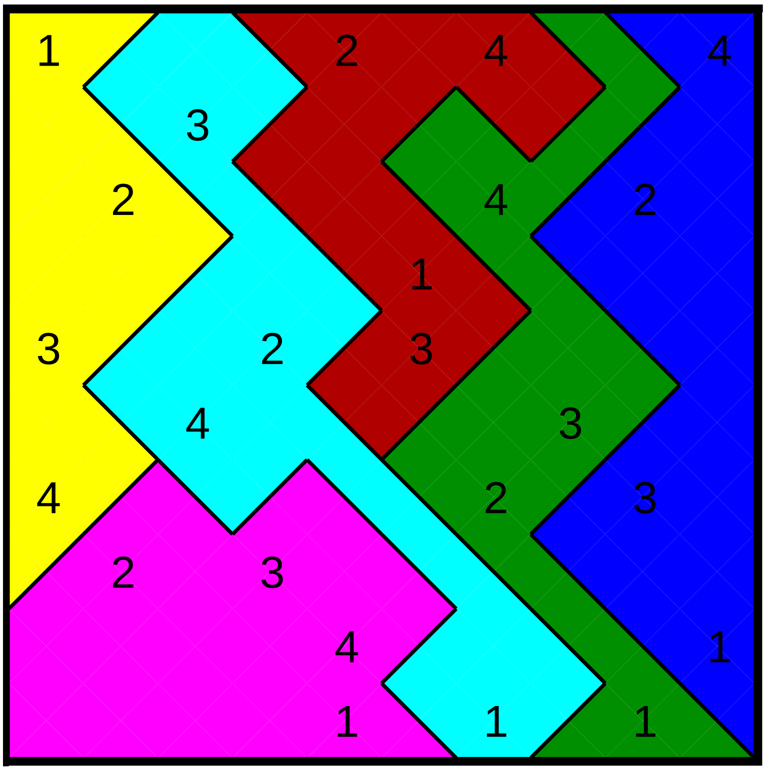


2

10×10 **solutions**



3



4