Daily Sloop Returns

by Scor

Here is the collection of simple loop puzzles and guides from the Daily Sloop Returns series, which ran for 25 days on the CTC discord server.

All puzzles follow regular simple loop rules:

Draw a non-intersecting loop through the centers of all empty cells.

Some variant puzzles have additional rules, which are clarified alongside the puzzle.

Please note that parity is not intended to be used to solve any puzzle in this series, and its usage is thus discouraged.

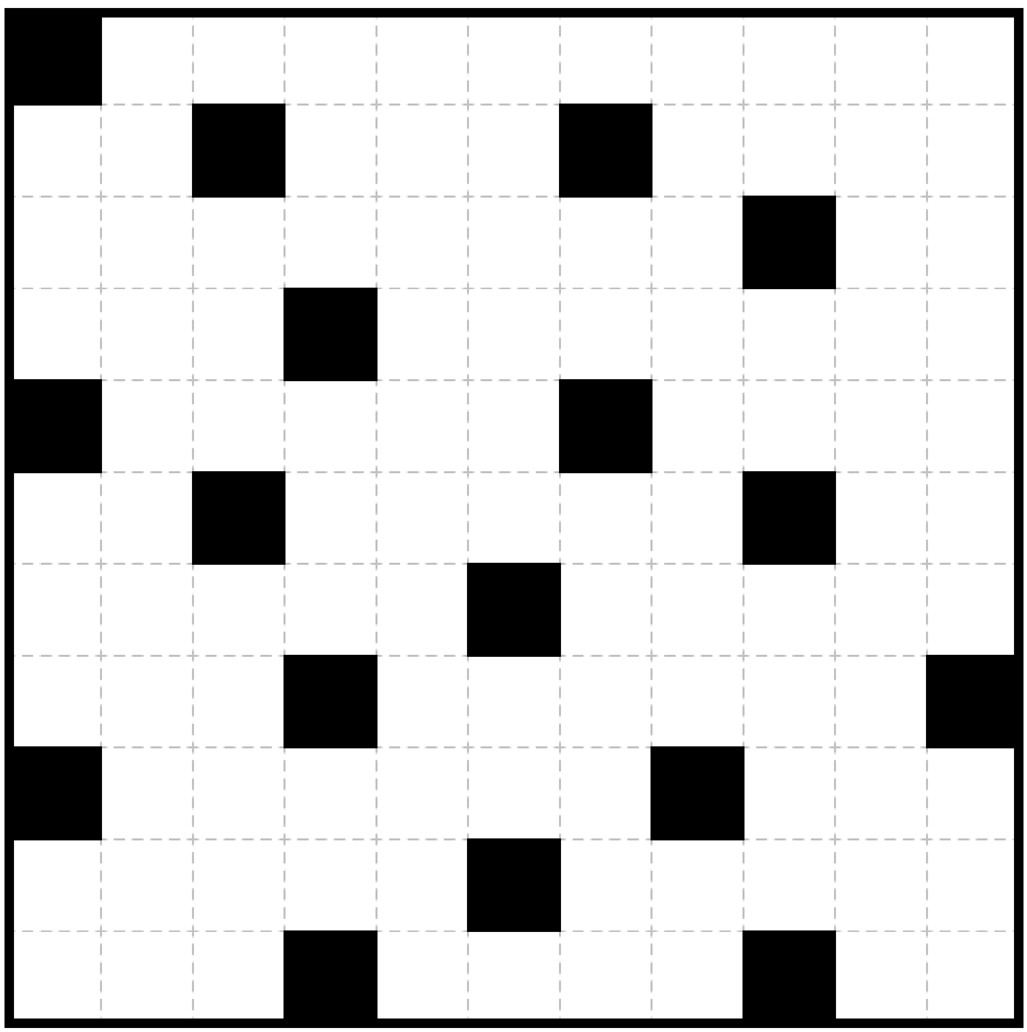
Enjoy!

DSR #1 - Recap

Solve it online:

[Puzz.link]

[Penpa+]

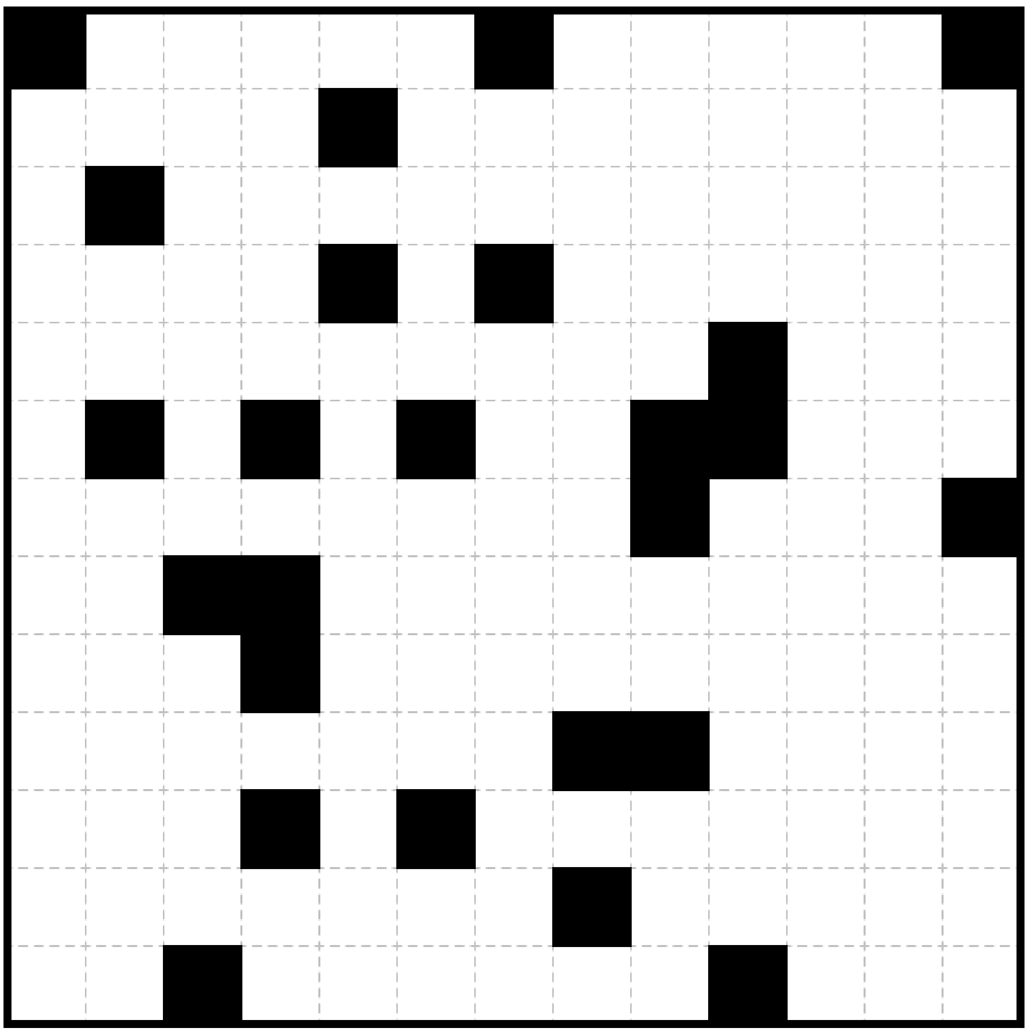


DSR #2 - Recap

Solve it online:

[Puzz.link]

[Penpa+]

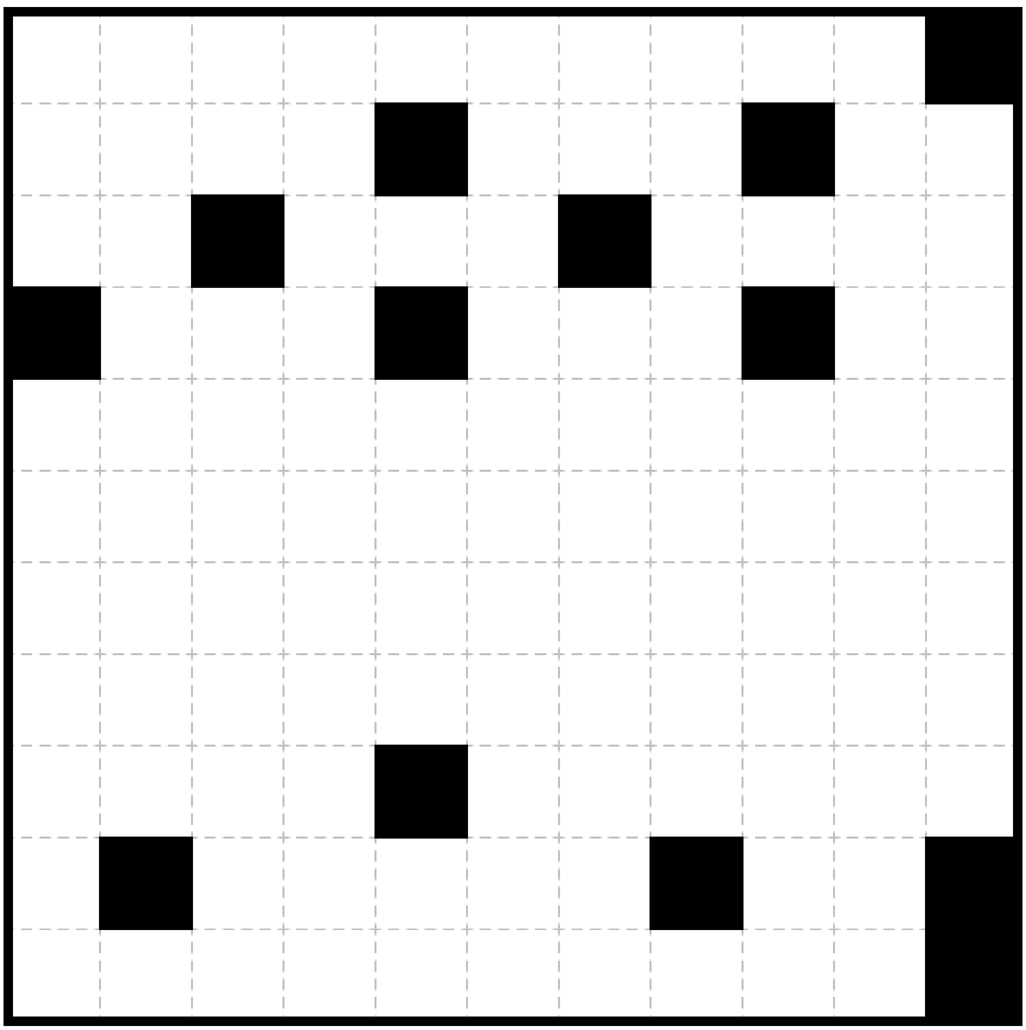


DSR #3 - Recap

Solve it online:

Puzz.link

Penpa+

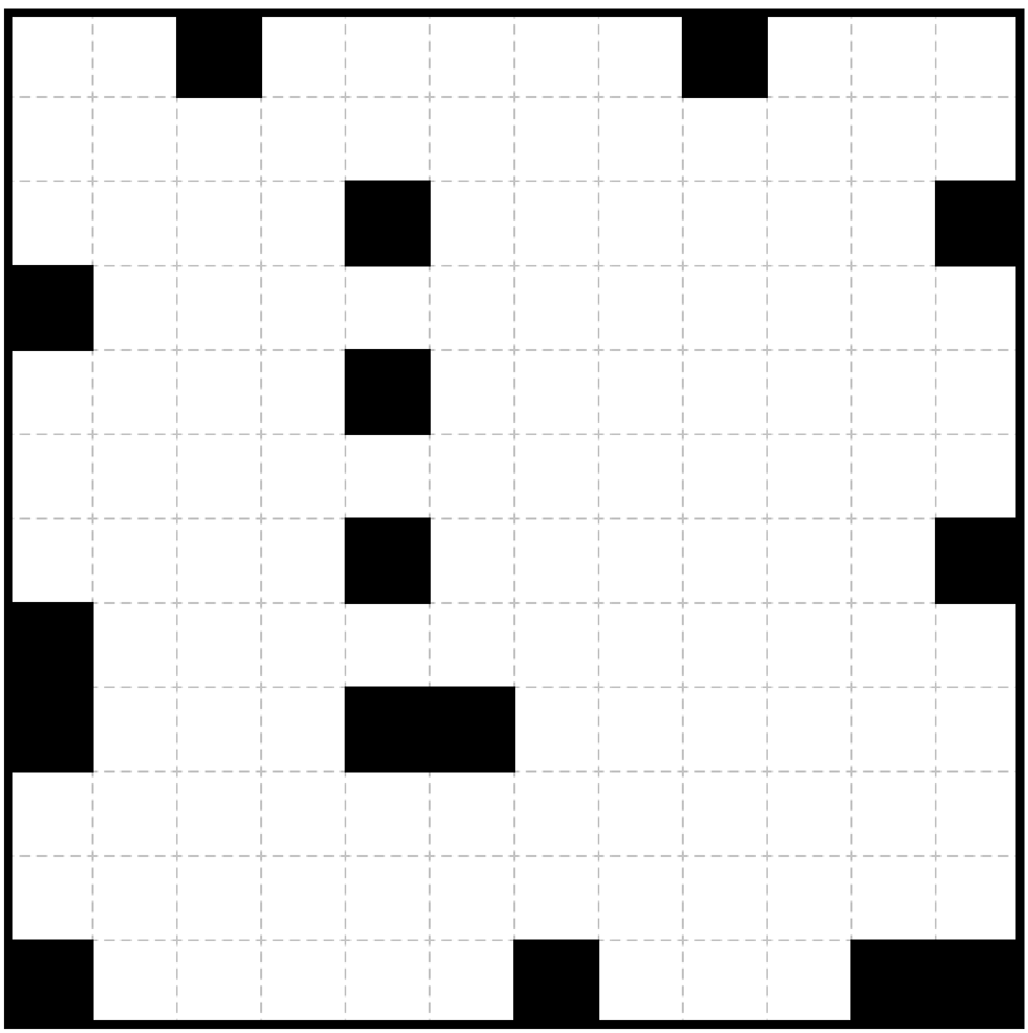


DSR #4 - Recap

Solve it online:

[Puzz.link]

[Penpa+]

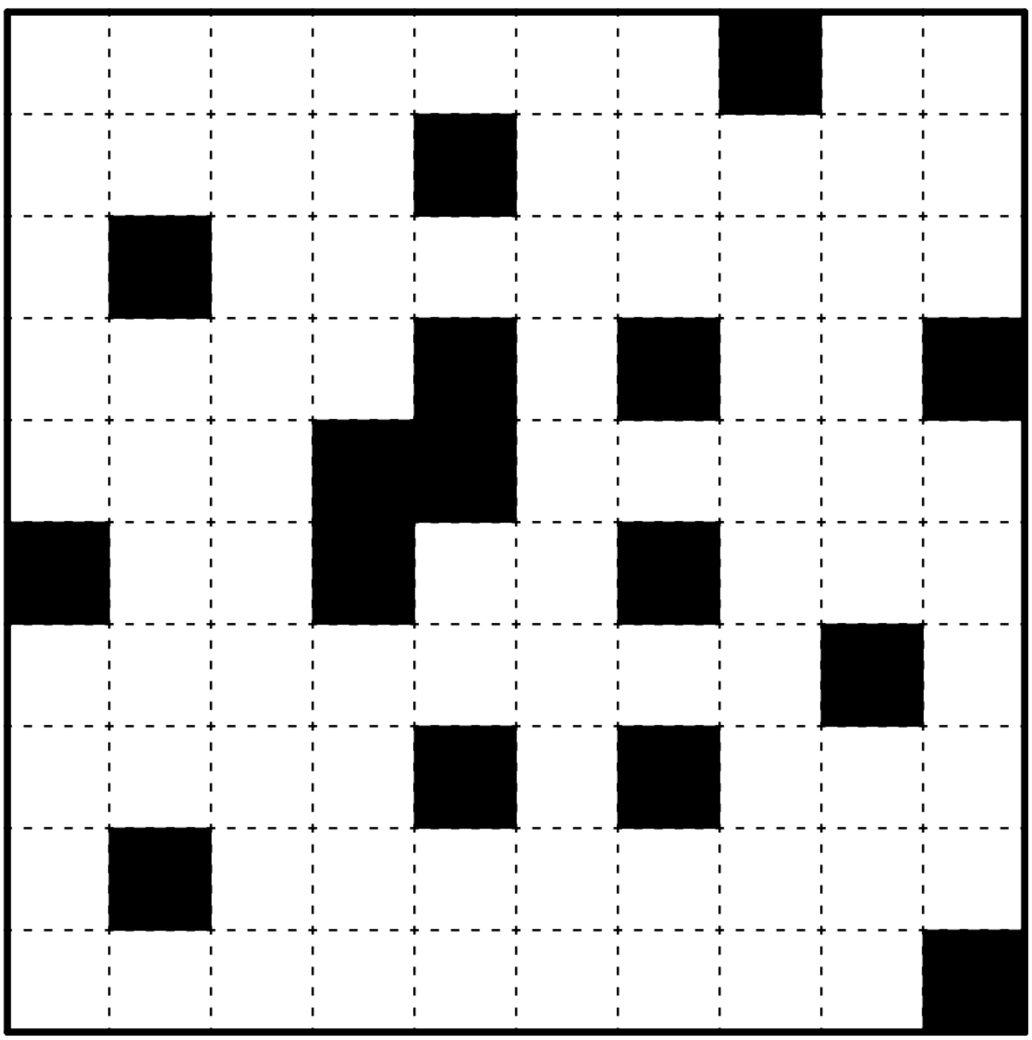


DSR #5 - Variant (Crossing)

Variant rule: Two perpendicular line segments may intersect each other, but not turn at their intersection or otherwise overlap.

Solve it online:

Penpa+

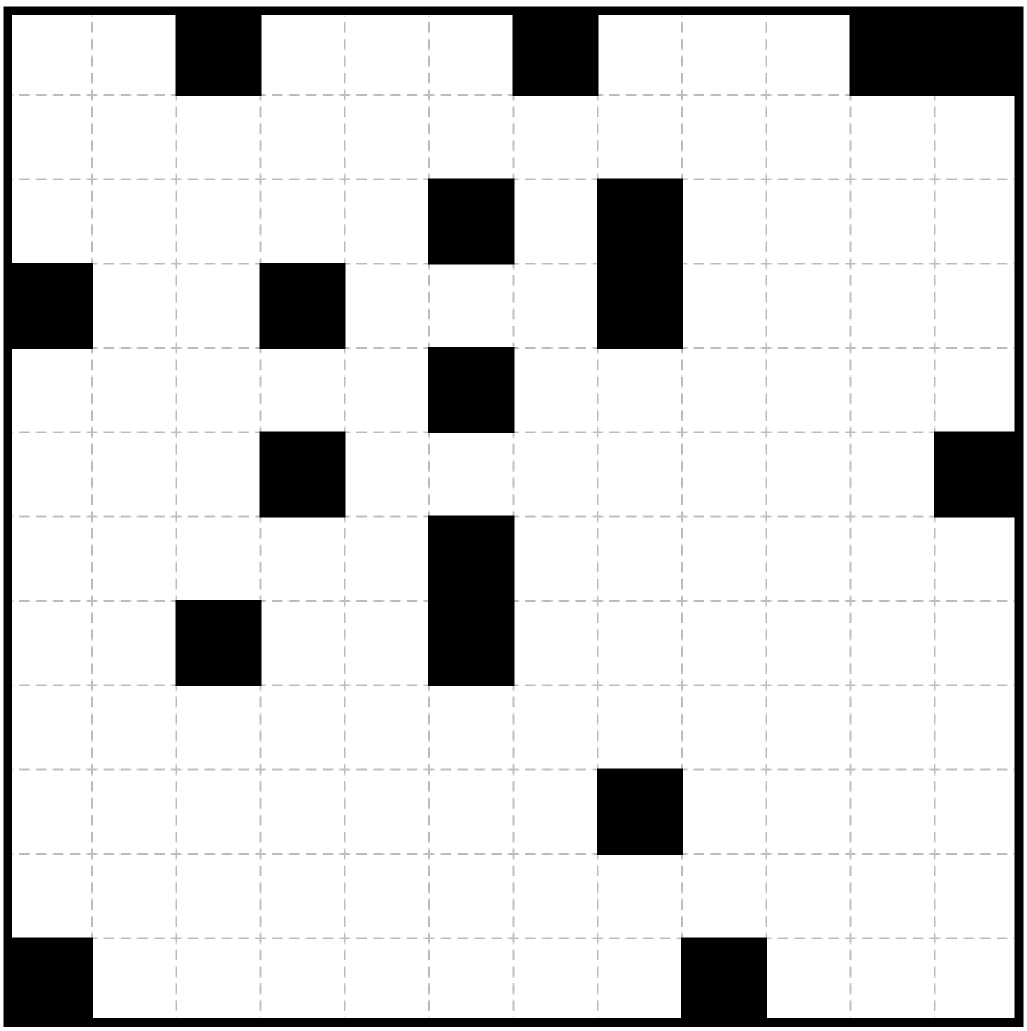


DSR #6 - Recap

Solve it online:

Puzz.link

Penpa+

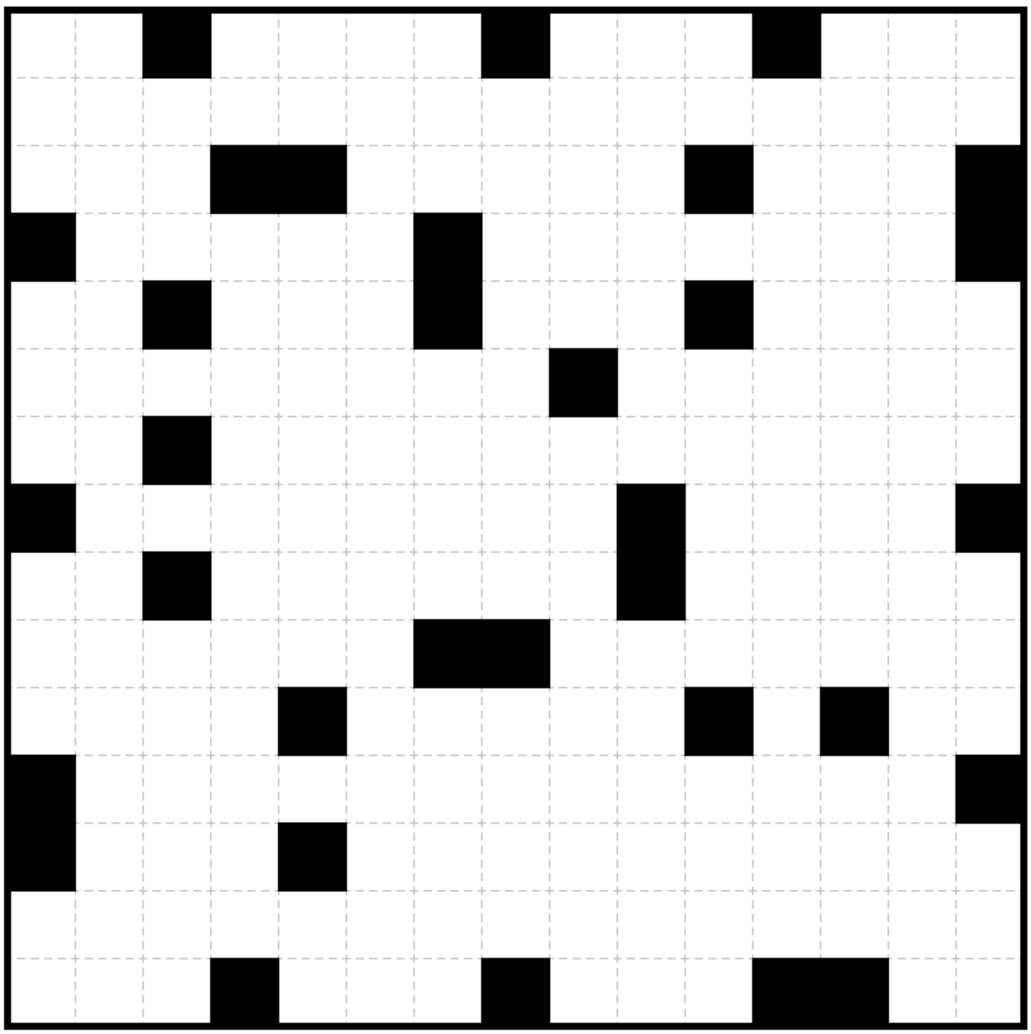


DSR #7 - Recap

Solve it online:

Puzz.link

Penpa+



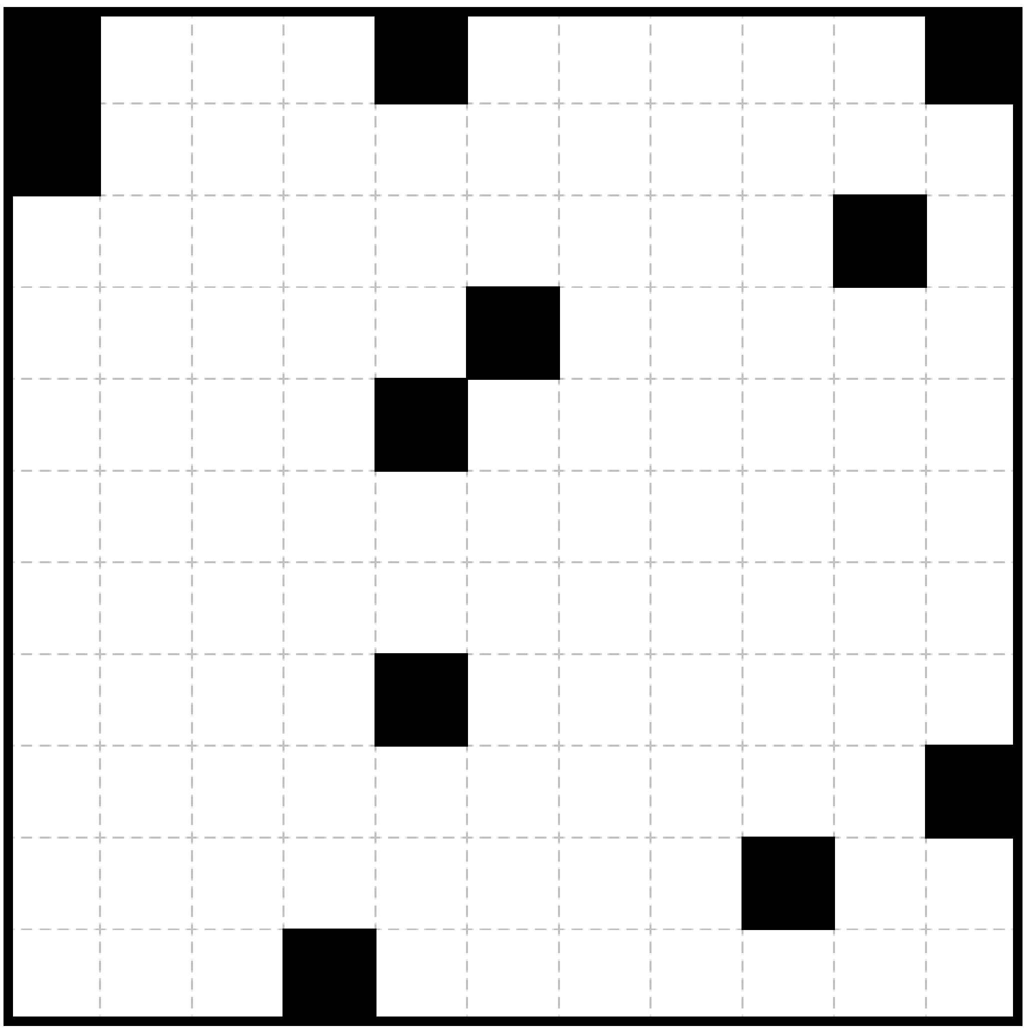
DSR #8 - Link Theory

Guide provided: Check out the guide to learn the theory required to solve the puzzles in this section.

Solve it online:

Puzz.link

Penpa+

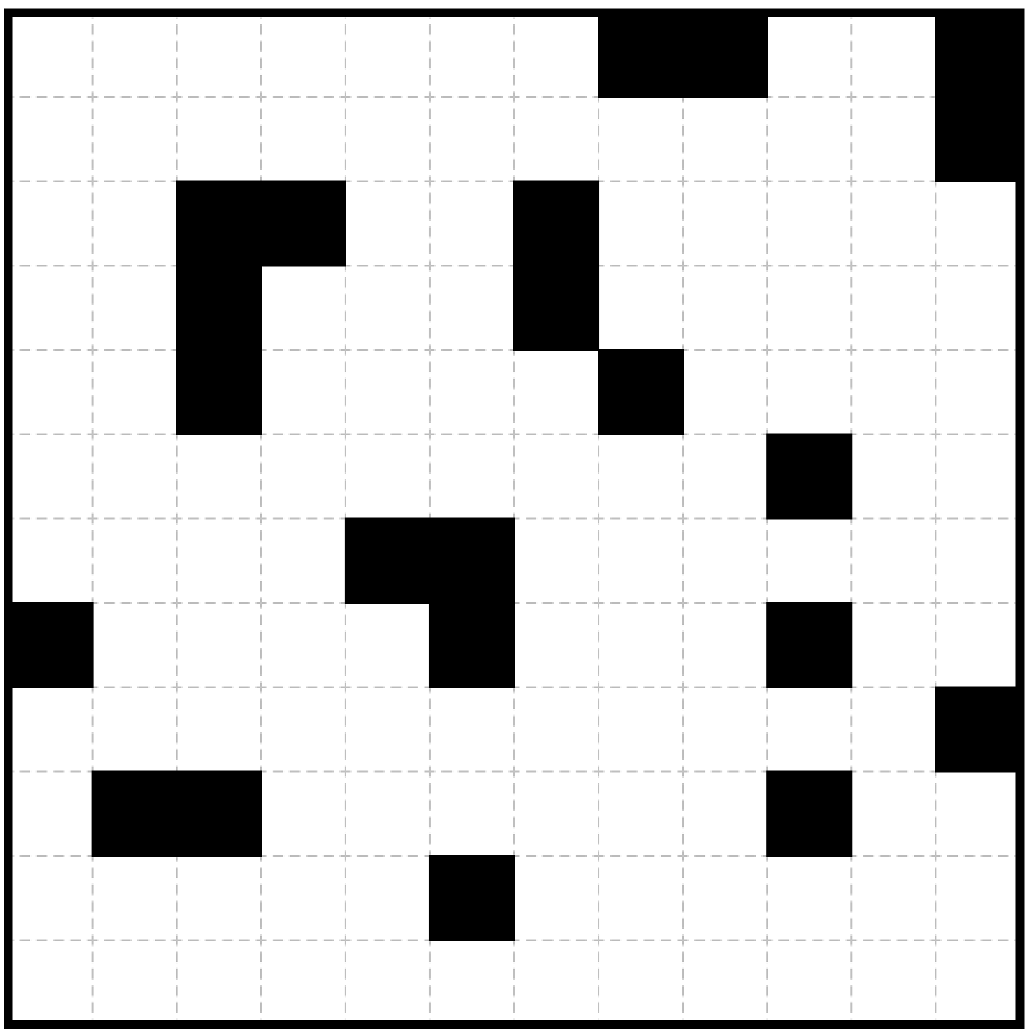


DSR #9 - Link Theory

Solve it online:

[Puzz.link]

[Penpa+]



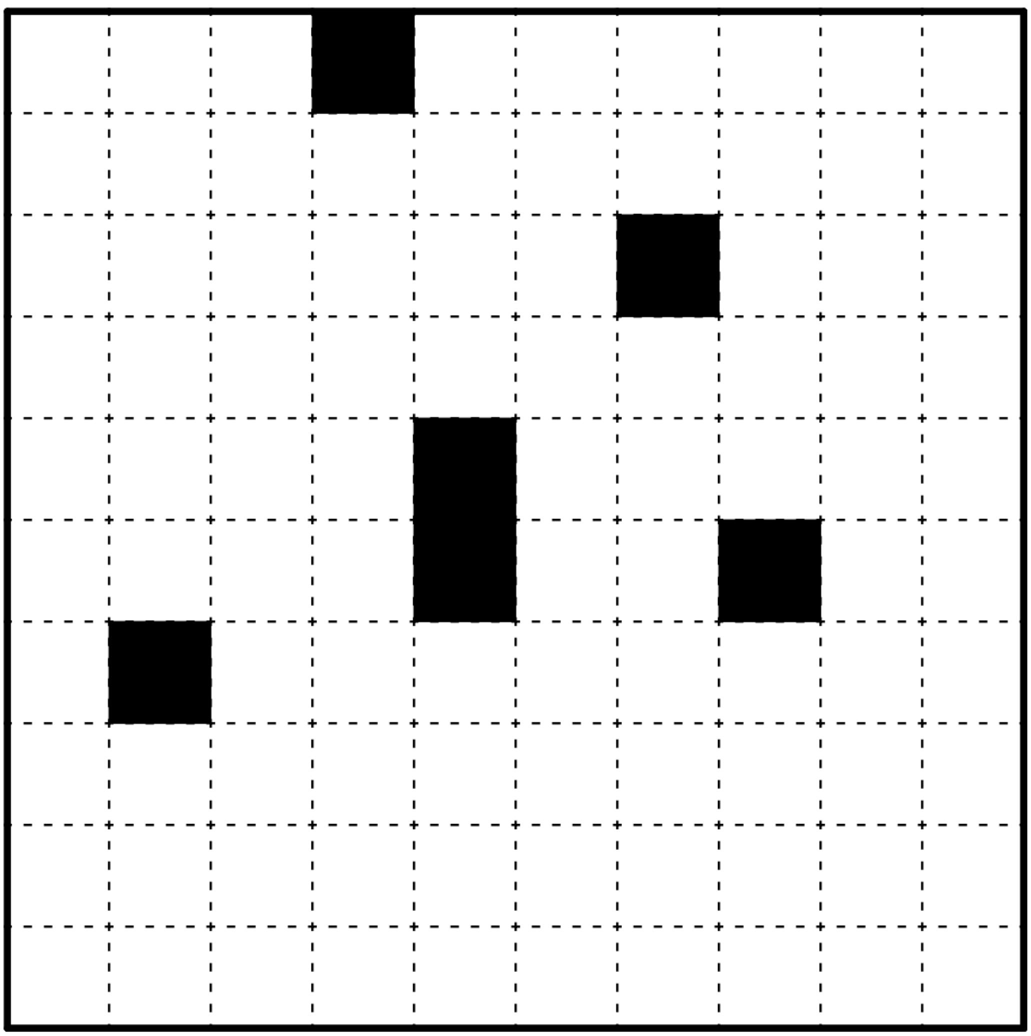
DSR #10 - Variant (Short)

<u>Variant rule</u>: No straight line segment may be more than two cells in length.

Solve it online:

[Puzzlink (no precise answer check)]

[Penpa+]

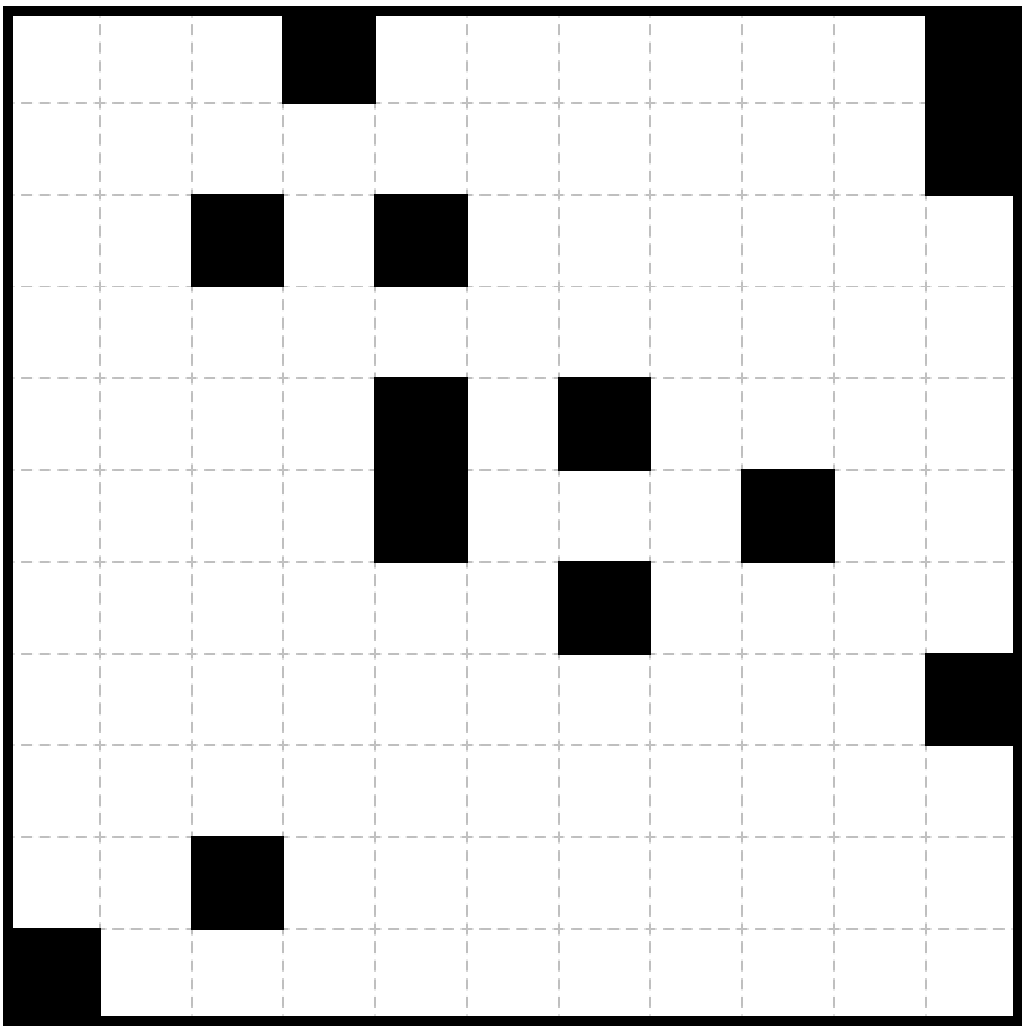


DSR #11 - Link Theory

Solve it online:

Puzz.link

Penpa+

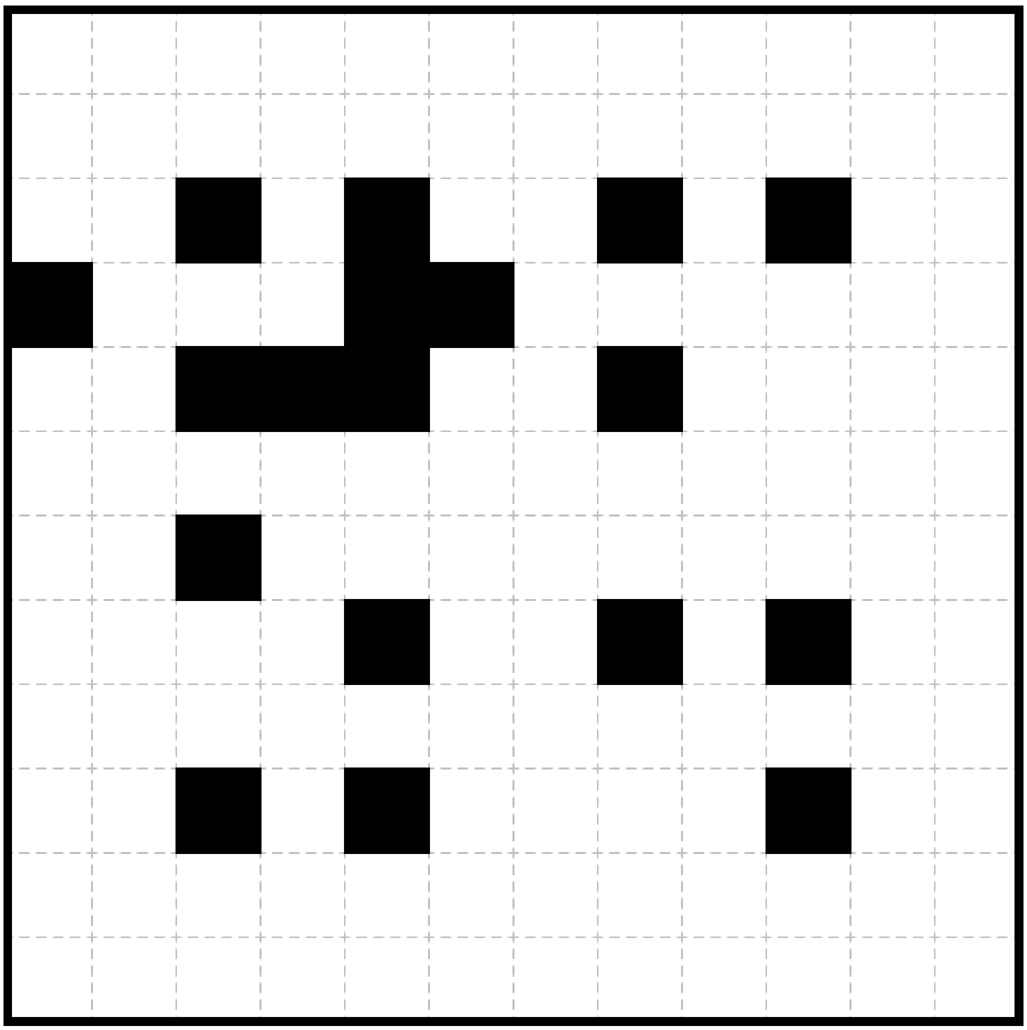


DSR #12 - Link Theory

Solve it online:

Puzz.link

Penpa+

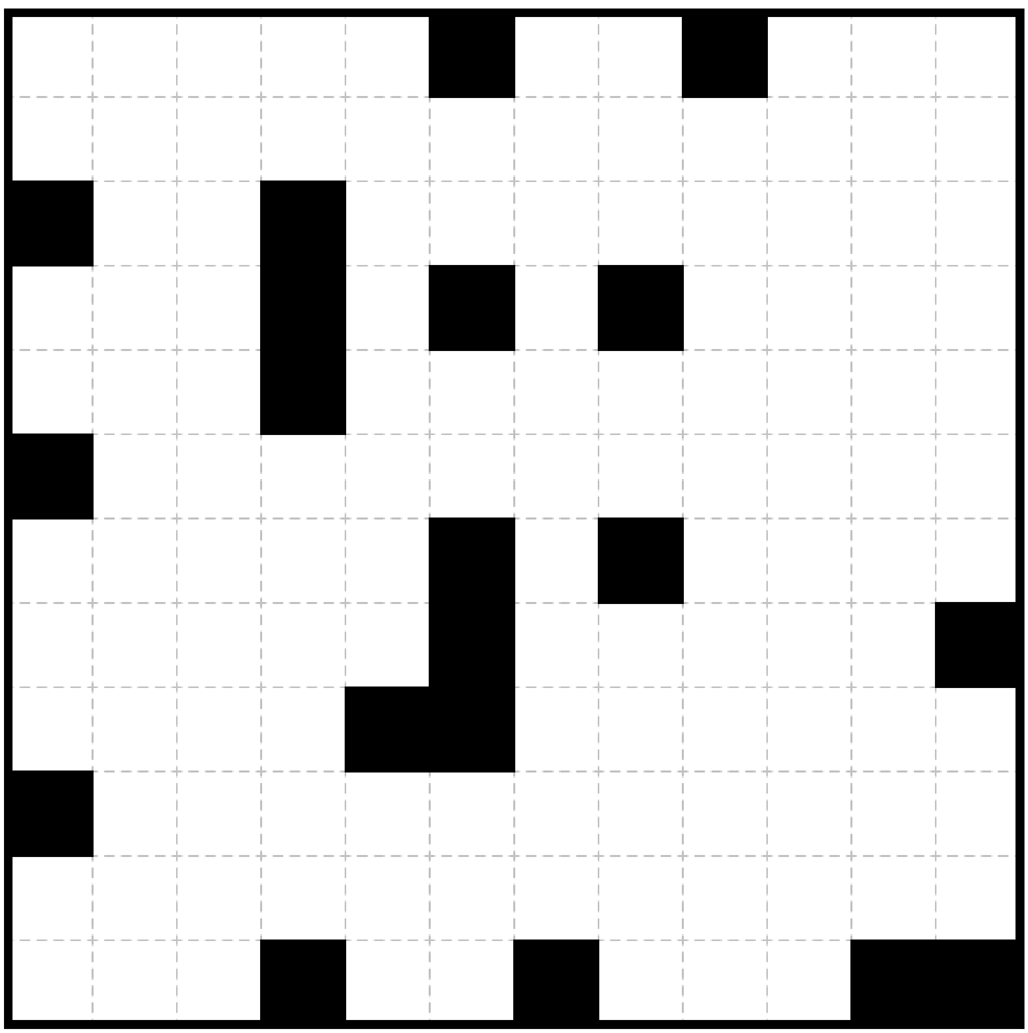


DSR #13 - Link Theory

Solve it online:

[Puzz.link]

[Penpa+]

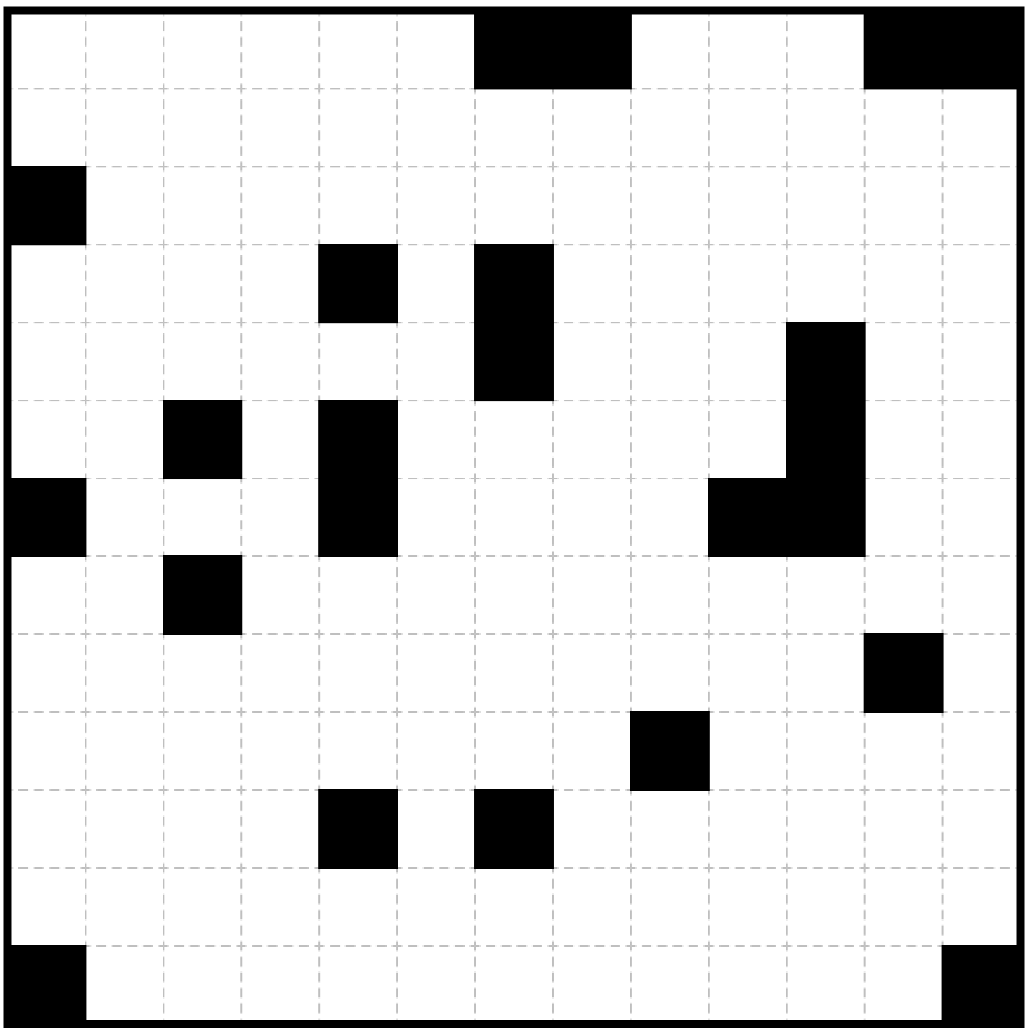


DSR #14 - Link Theory

Solve it online:

[Puzz.link]

[Penpa+]



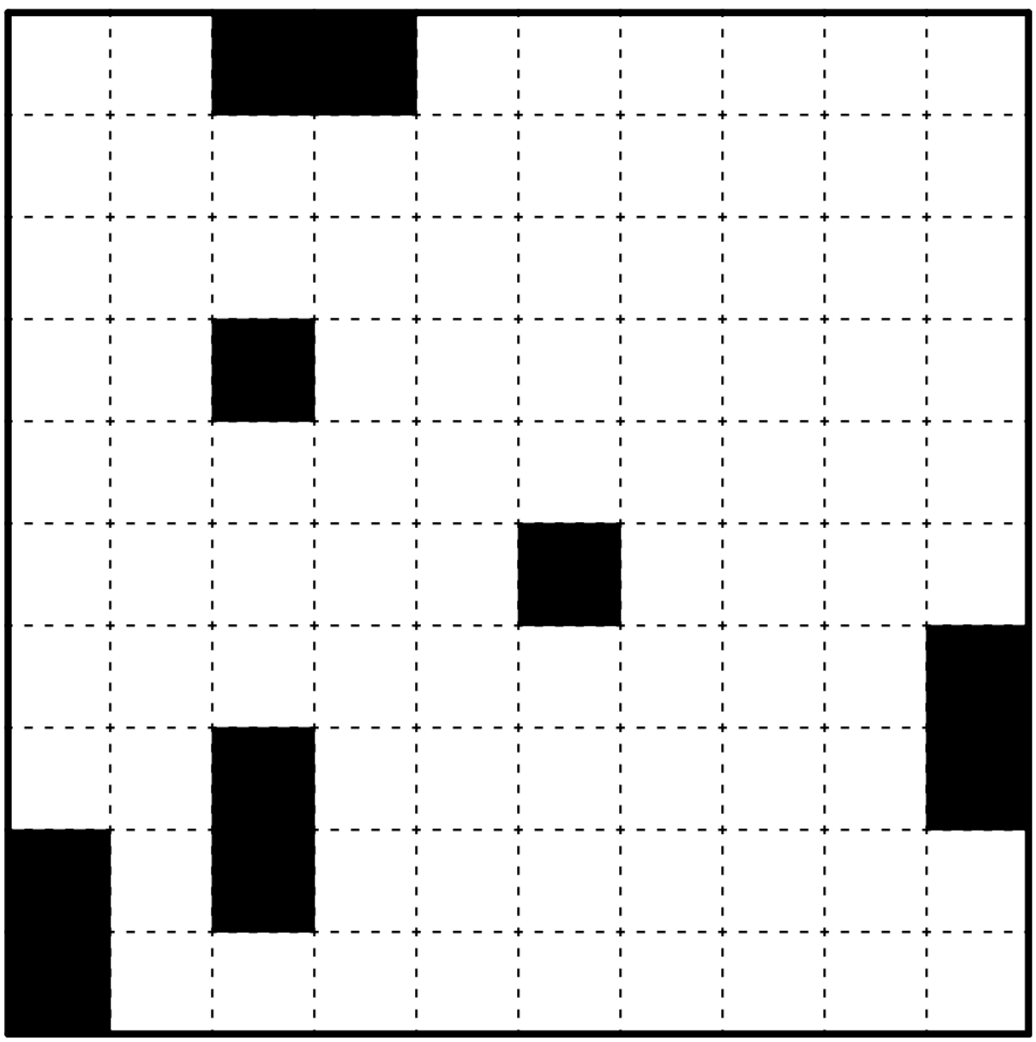
DSR #15 - Variant (Unequal Lengths)

Variant rule: No two consecutive straight line segments may be the same length.

Solve it online:

Puzzlink (no precise answer check)

Penpa+

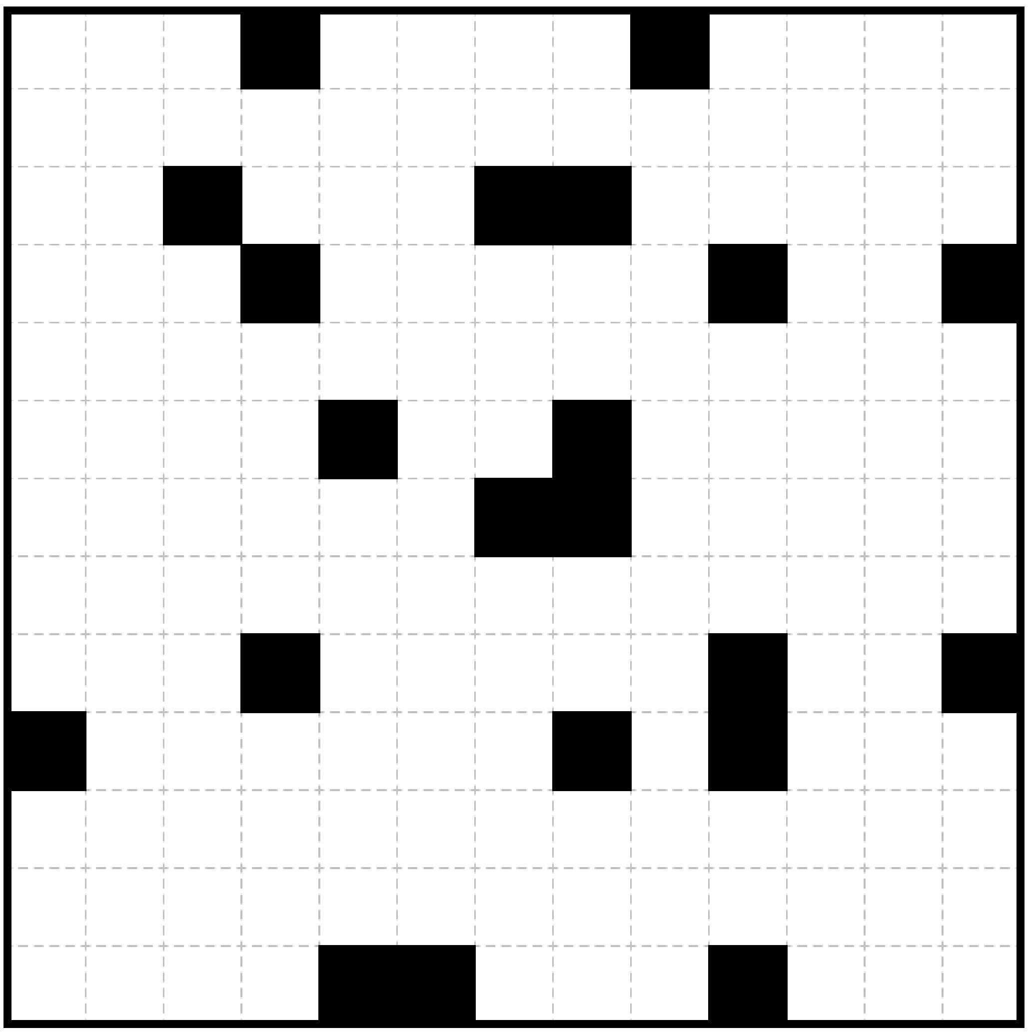


DSR #16 - Link Theory

Solve it online:

[Puzz.link]

[Penpa+]

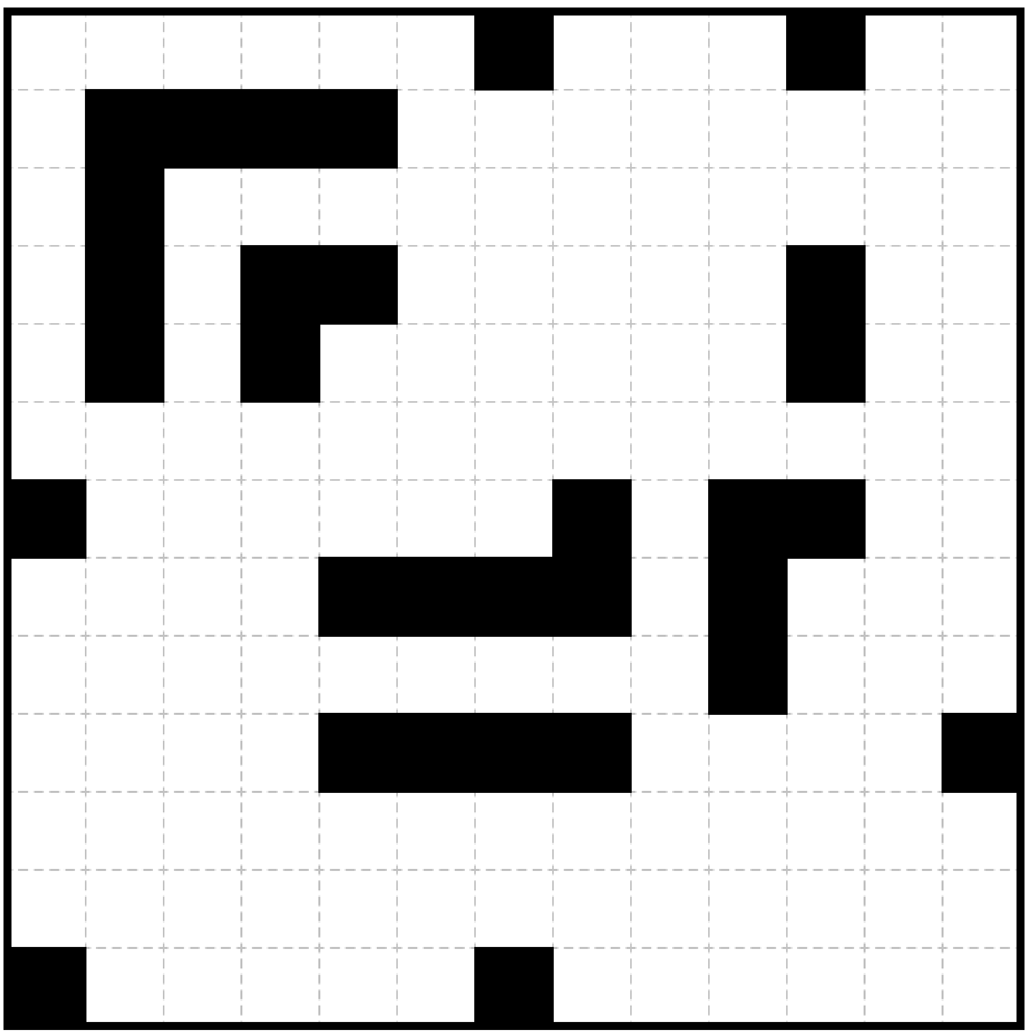


DSR #17 - Link Theory

Solve it online:

Puzz.link

Penpa+

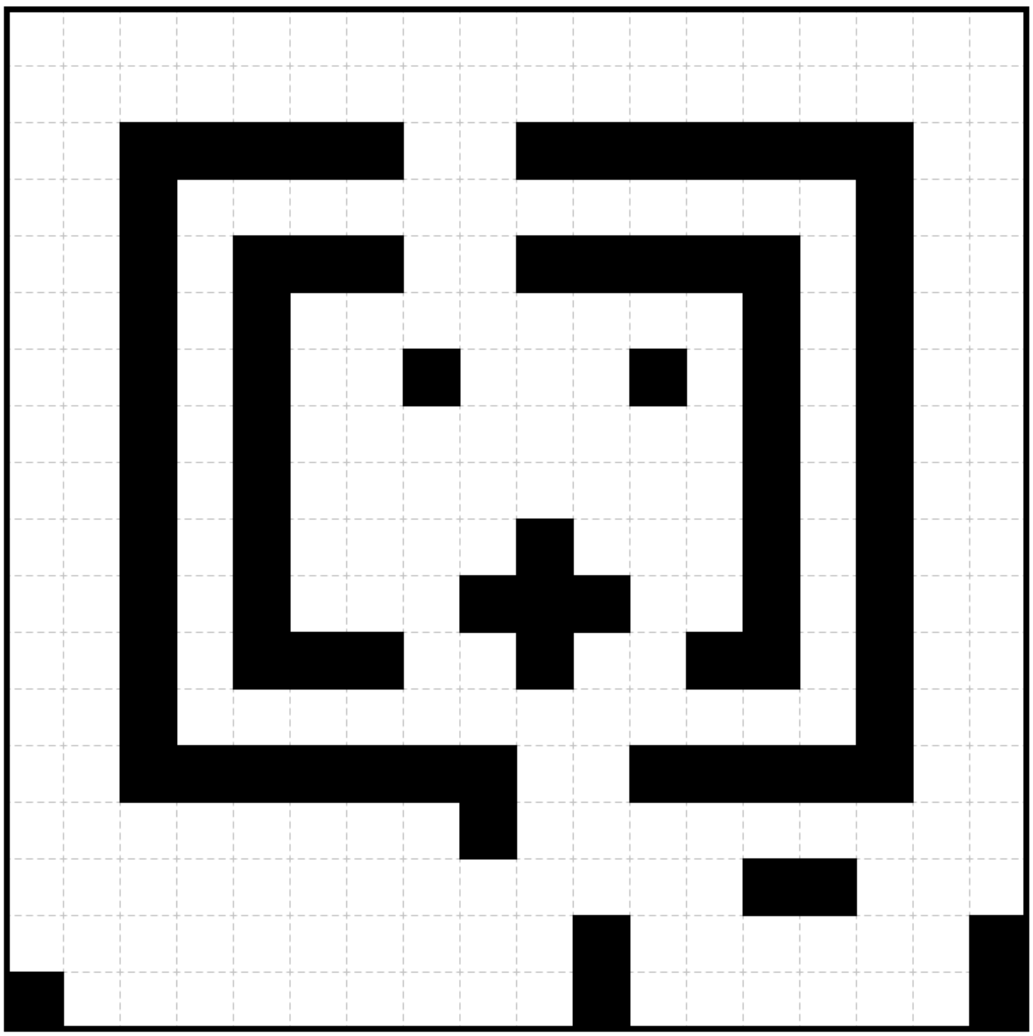


DSR #18 - Link Theory

Solve it online:

Puzz.link

Penpa+

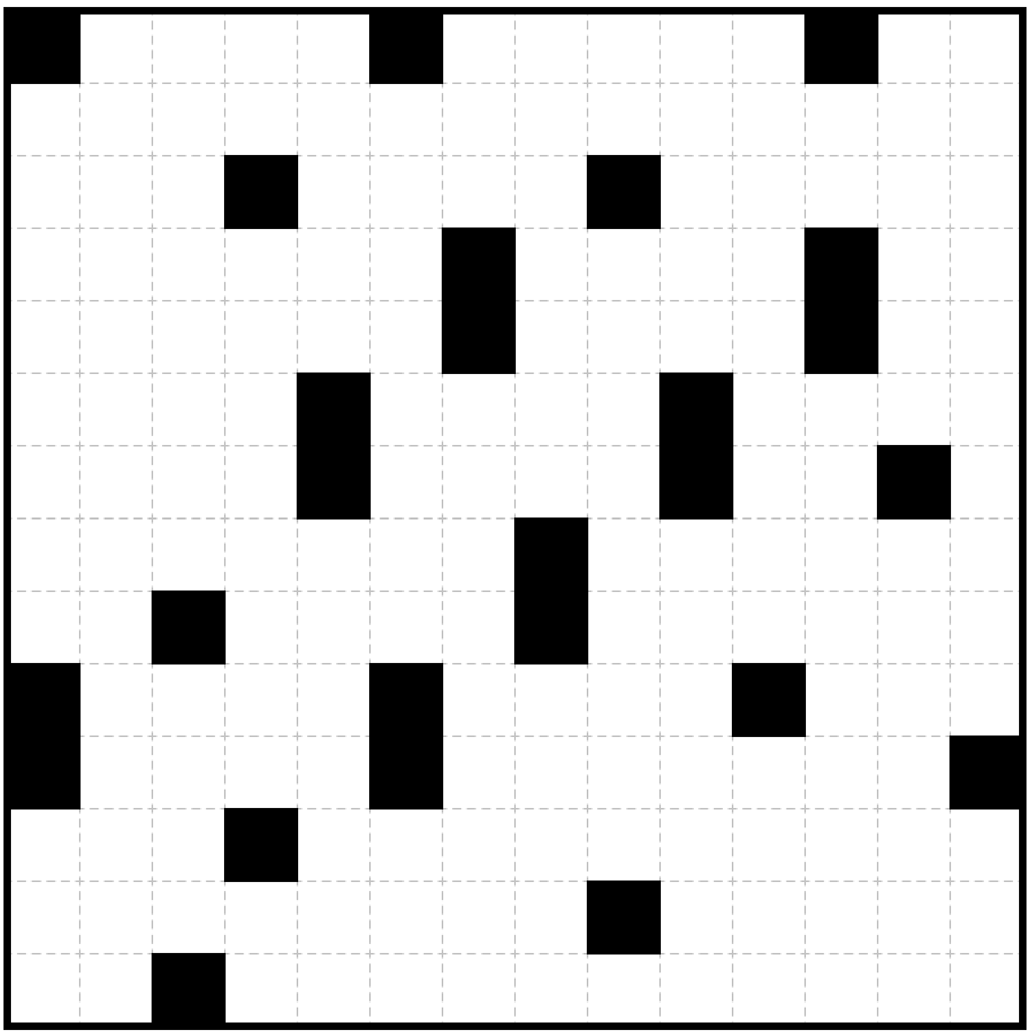


DSR #19 - Link Theory

Solve it online:

Puzz.link

Penpa+

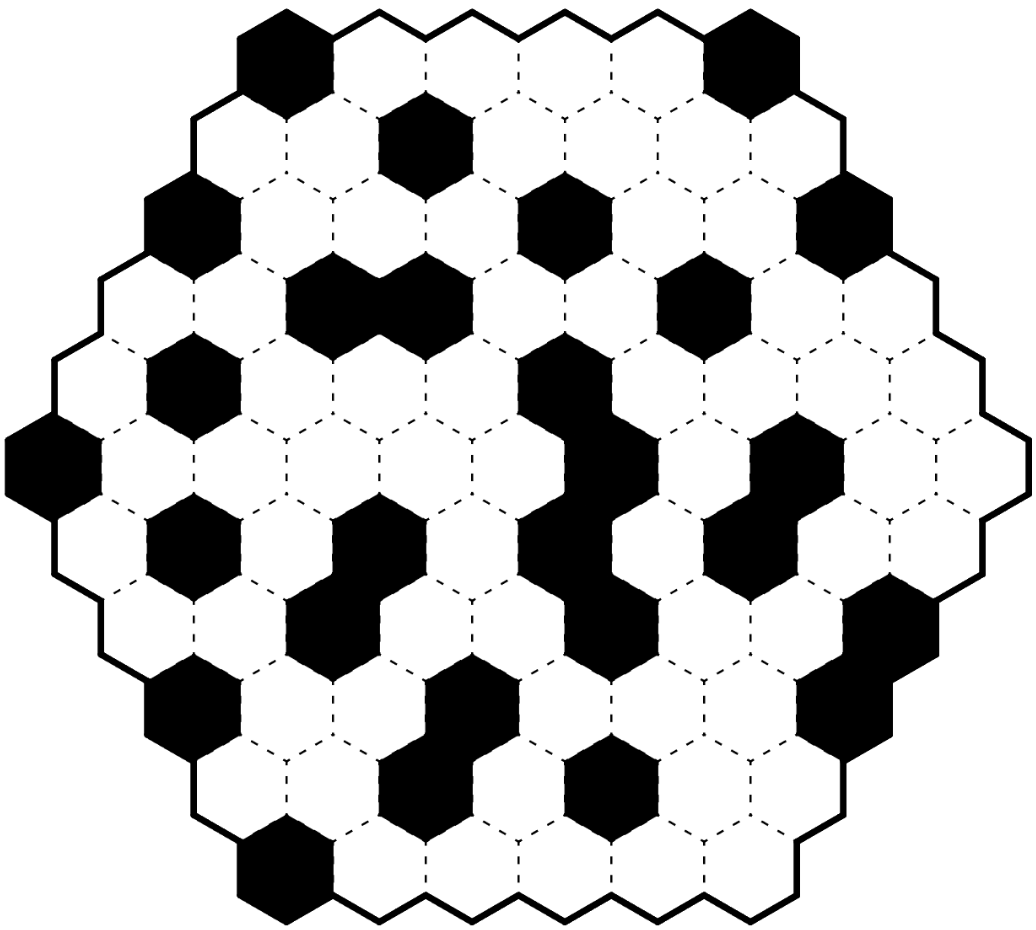


DSR #20 - Variant (Hex Grid)

No additional rules apply.

Solve it online:

Penpa+



DSR #21 - Uniqueness

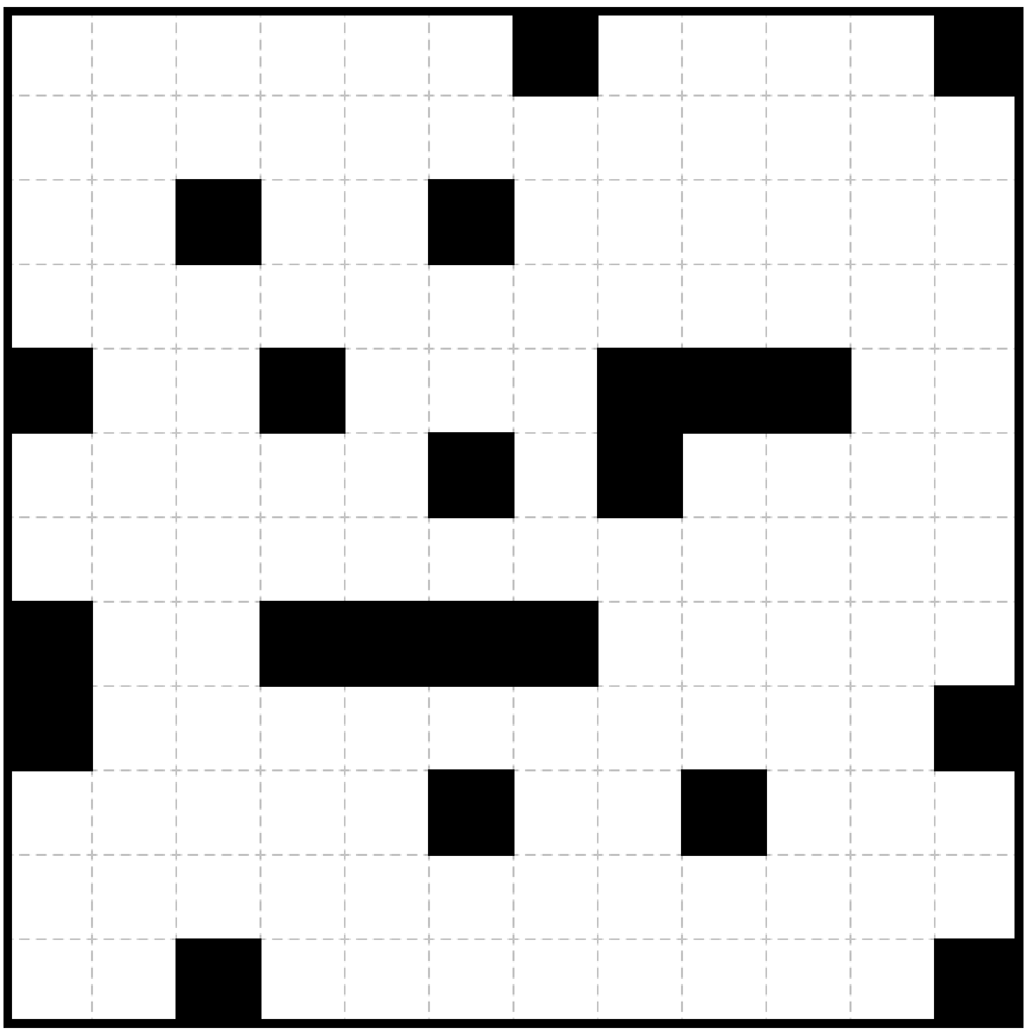
Guide provided: Check out the guide to learn the theory required to solve the puzzles in this section.

This puzzle is known to have exactly one solution.

Solve it online:

Puzz.link

Penpa+



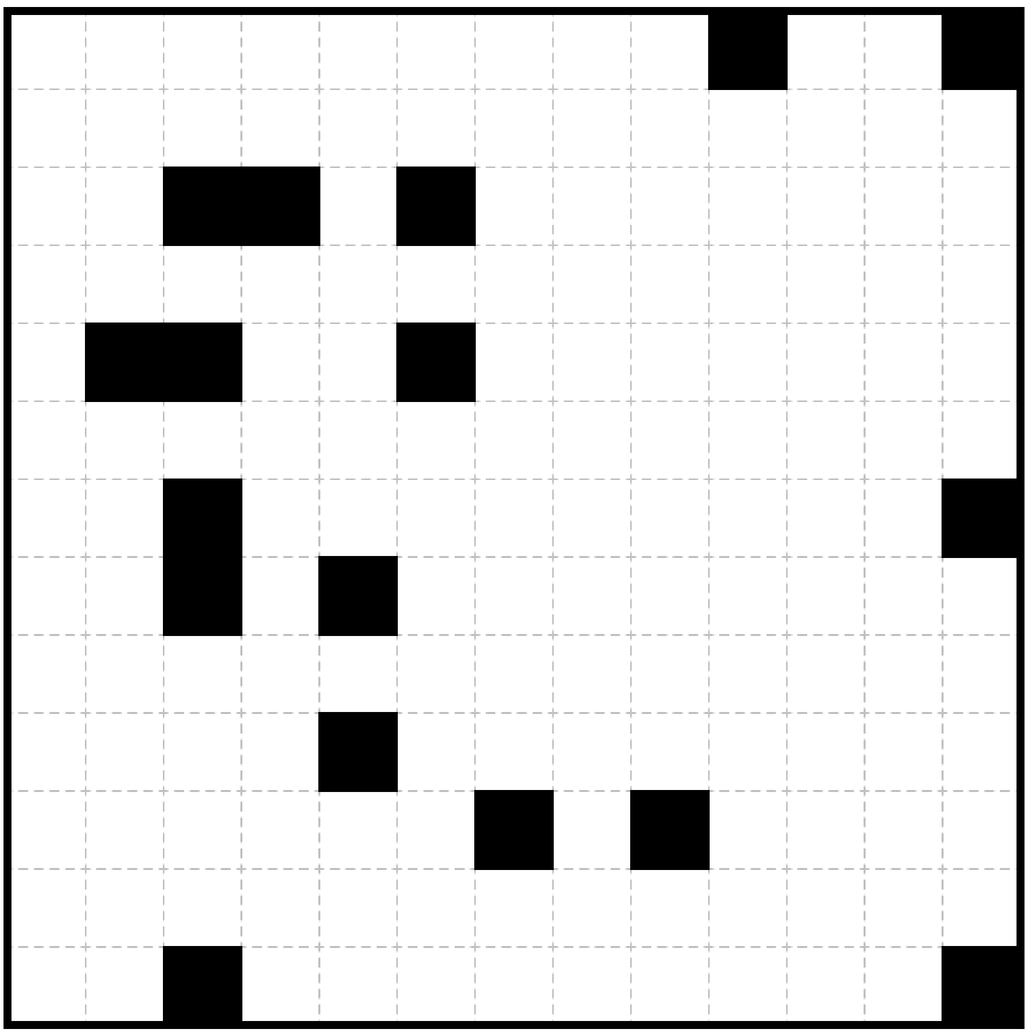
DSR #22 - Uniqueness

This puzzle is known to have exactly one solution.

Solve it online:

[Puzz.link]

[Penpa+]



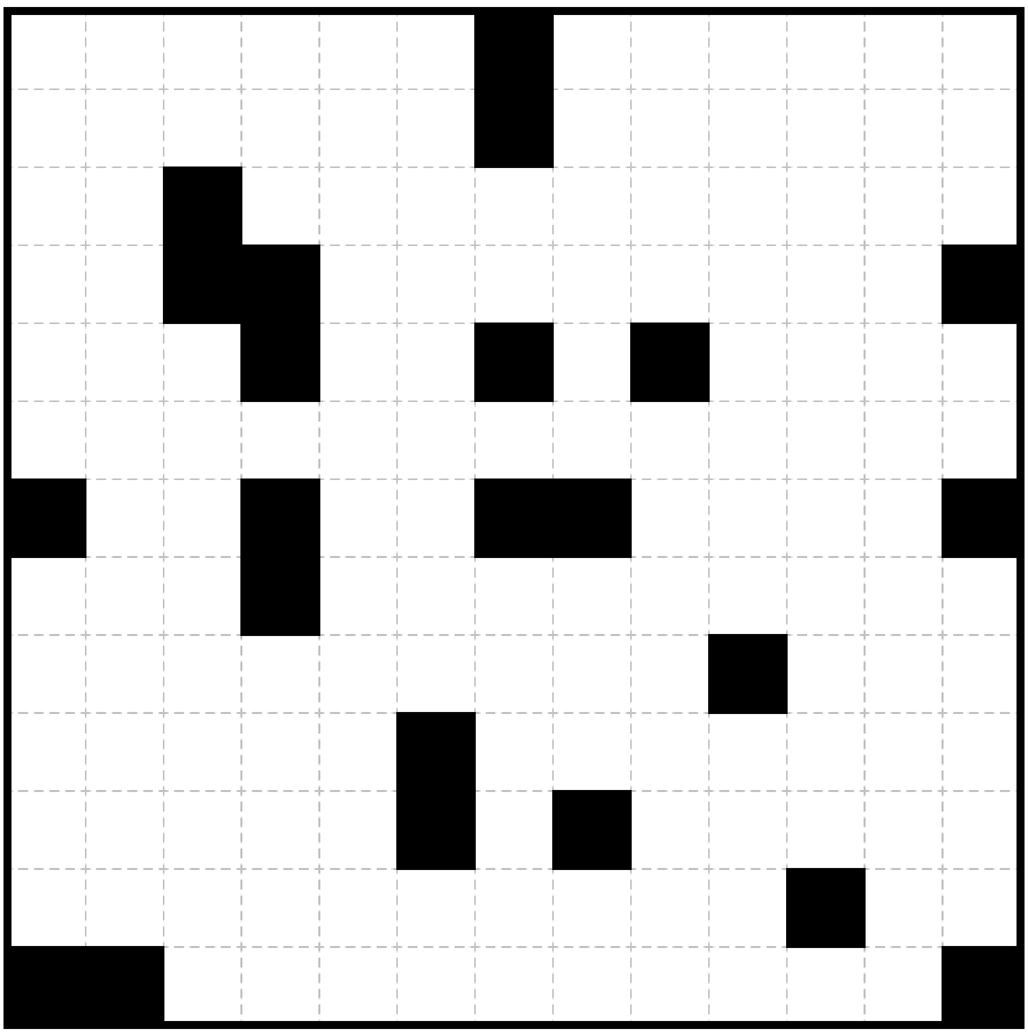
DSR #23 - Uniqueness

This puzzle is known to have exactly one solution.

Solve it online:

Puzz.link

Penpa+



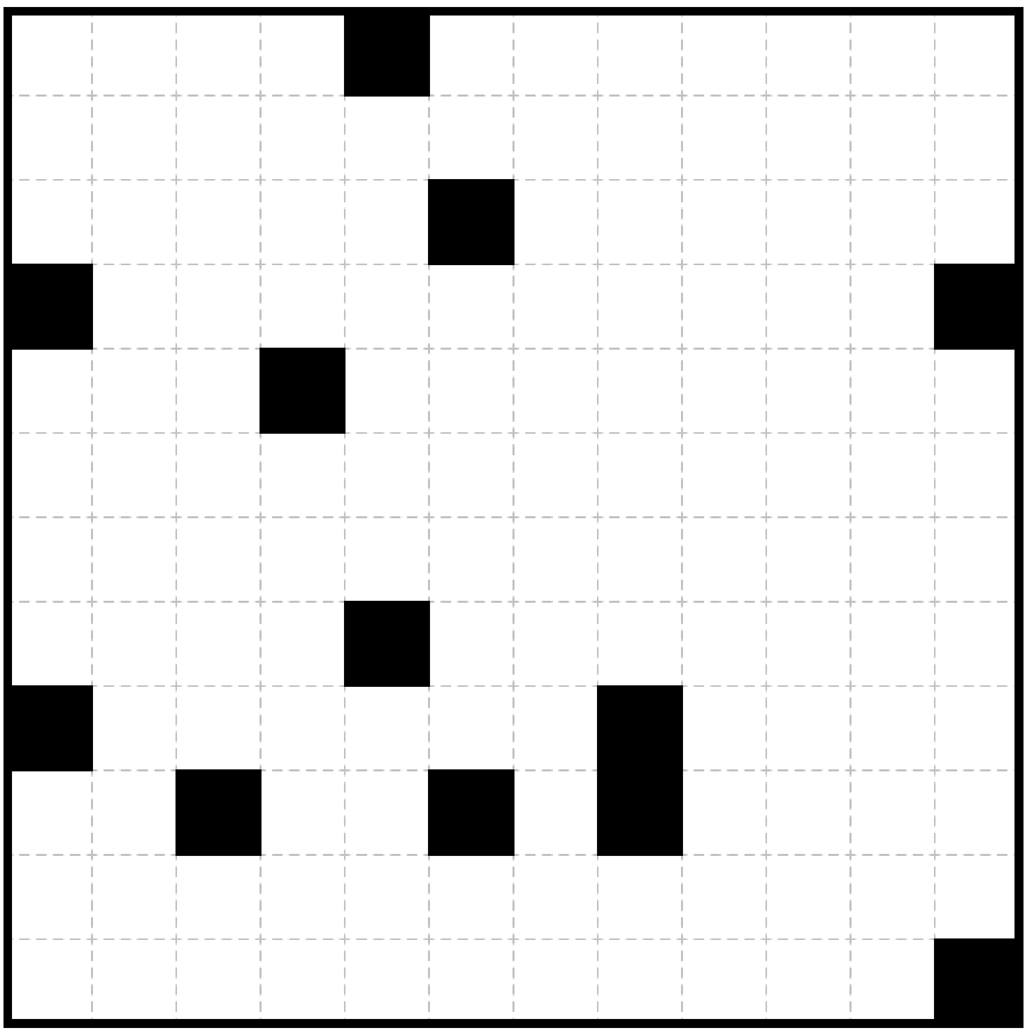
DSR #24 - Uniqueness

This puzzle is known to have exactly one solution.

Solve it online:

[Puzz.link]

[Penpa+]

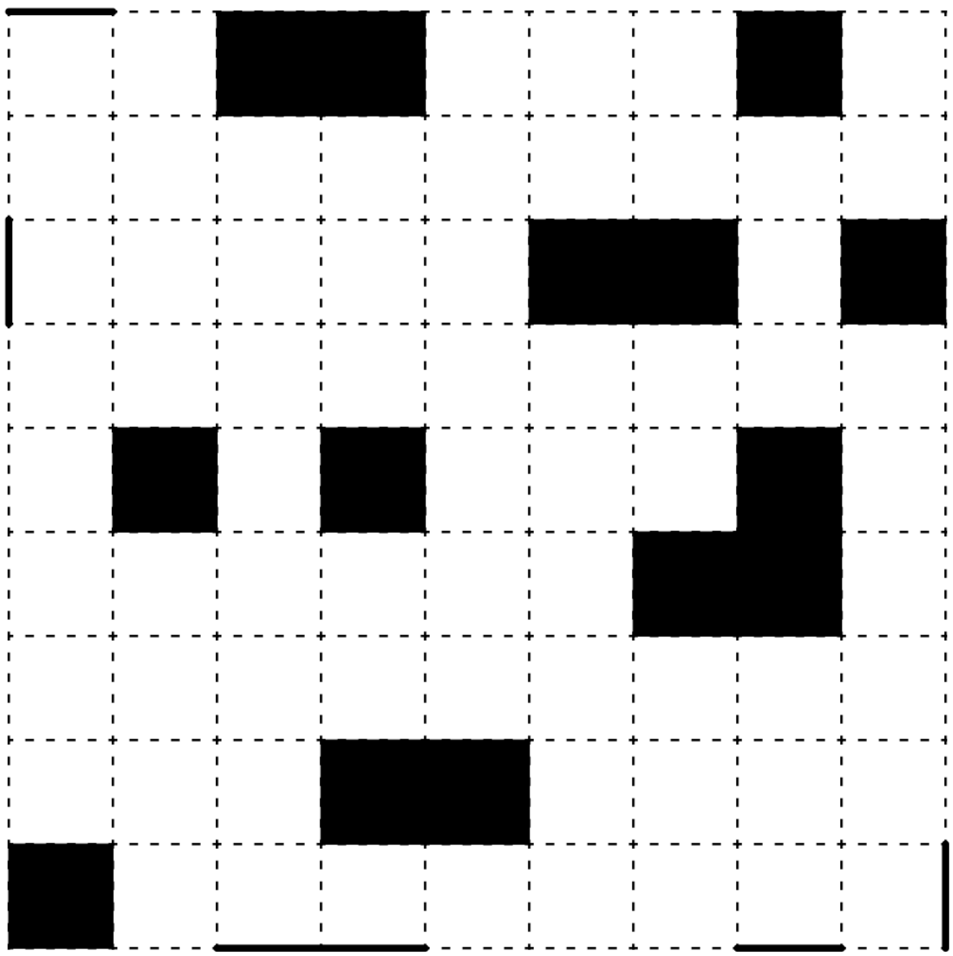


DSR #25 - Variant (Toroidal)

<u>Variant rule</u>: The loop may exit one side of the grid and reenter on the opposite side.

Solve it online:

[Penpa+]()



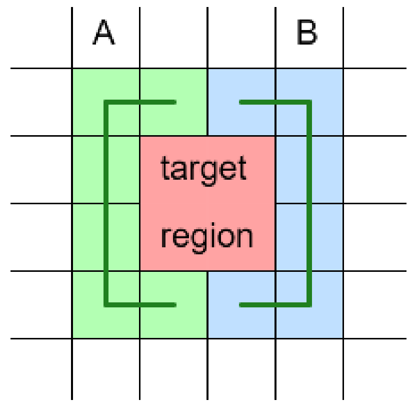
Link Theory Guide

Link Theory Guide

Link theory describes a set of deductions that can be made in simple loop puzzles. It is a generalisation of the bouncing technique, and can be applied to a greater variety of scenarios. This guide will walk through the theory and some example applications of it.

Previously, to use bouncing, we needed to identify a “target cell” that has a forced exit to avoid closing the loop early.

For link theory, however, we will instead identify two regions, “A” and “B”, of which all cells within each region are known to be immediately connected, implying that they have exactly two exits each. We will also identify a “target region”, whose only exits enter into A or B.



Also for the setup to work, we need the “complementary region” - all cells that are not part of A, B, or the target region - to be non-empty.

With this setup, there are four key rules that we can infer via link theory.

1. The target region has exactly two exits - one into A, and one into B.

If the target region exits twice into either A or B, then neither segment “escapes” since all the exits of the entered region have been used up. Because the target region requires at least two exits to escape to reach the complementary region, one must be into A, and the other into B.

2. Region A may not enter into B (and vice versa).

If regions A and B were directly connected to each other, then the target region's exits into A and B would cause the loop to close before it could reach the complementary region.

3. The region formed by combining A, B, and the target region is immediately connected, and thus can be combined into a new region A.

The target region acts as a "link" between A and B, connecting these three regions together. This combined region has exactly two exits (one from A and one from B), so moving forward in the solve, it can be treated as a new region A.

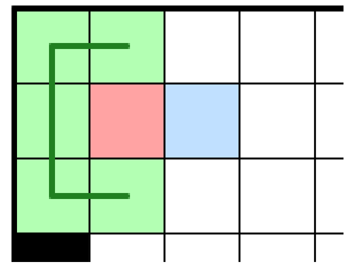
4. The complementary region also acts as a valid target region.

Like the target region, it also can only exit into A or B, and therefore it can be substituted instead of the current target region. So, rules 1 and 3 can also be applied to it as well.

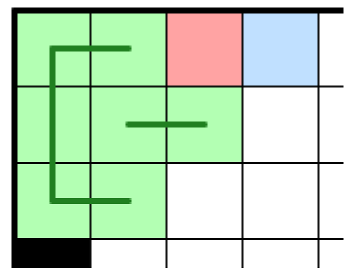
Let's go over some example applications of these rules.

For each of the following examples, green cells will represent region A, blue cells will represent region B, and red cells will represent the target region. All other white cells will make up the complementary region.

Example 1: Regular bouncing



Region A is our connected segment, the target region is the target cell of the bouncing, and region B is the cell of the remaining exit for the target region.

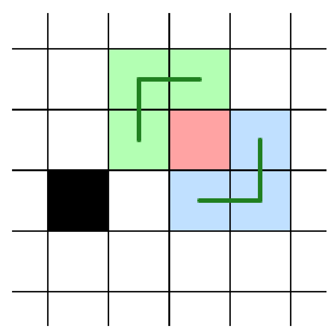


By rule 1, the target region must have an exit into B, giving a line segment from red to blue.

Then, by rule 3, we can combine these regions into our new region A, and find another setup to continue the bouncing chain.

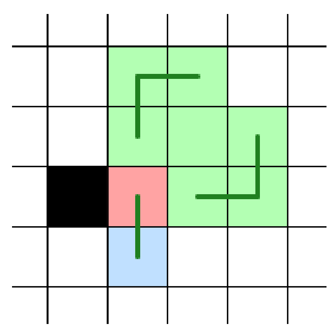
This implies that any bouncing chain can be treated as a region A or B.

Example 2: AB-link



In this scenario, we need to apply concurrent link theory rules to get deductions. Assume the initial deductions from the example images are known.

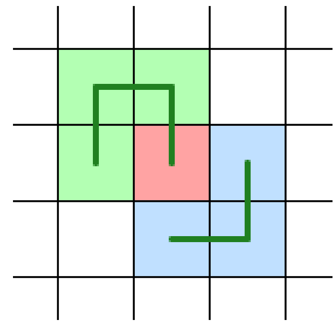
Since we have a valid setup, by rule 3 we can combine these three regions into a new region A.



We can then create another setup by using the shaded cell on the left. By rule 1, the target region must have an exit into B, giving a line segment from red to blue as shown.

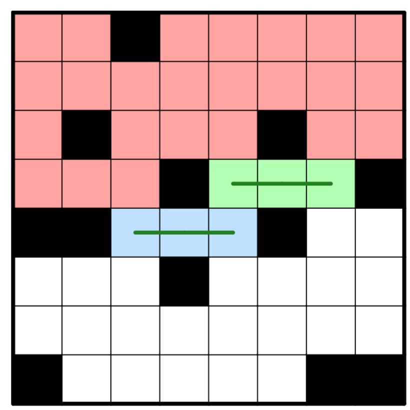
Example 3: Partially completed AB-link

When an AB-link is partially completed, it is often not as easy to spot because the target region is not as clear. However, the same logic as Example 2 can be applied in this scenario.



Example 4: Across the grid

Here's an interesting case that uses a large target region in the setup.



By rule 1, the target region must exit into B, giving a line segment from r4c3 to r5c3.

By rule 2, A and B cannot connect to each other, giving a cross between r4c5 and r5c5.

And by rule 4, combined with rule 1, the complementary region must exit into A, giving a line segment from r4c7 to r5c7.

Link theory outside of simple loop puzzles

Link theory can be applied to almost all loop genres. Whenever it is known that the loop must pass through the target region and the complementary region, the same set of rules can be applied.

Uniqueness Guide

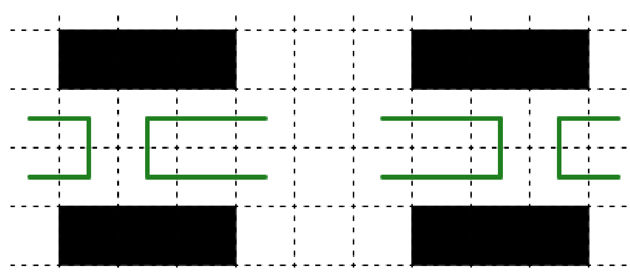
Uniqueness patterns in simple loop puzzles

Uniqueness is an interesting category of deductions because it assumes beforehand that a puzzle has exactly one solution. This guide will go over three common scenarios where such uniqueness deductions can be applied in simple loop puzzles.

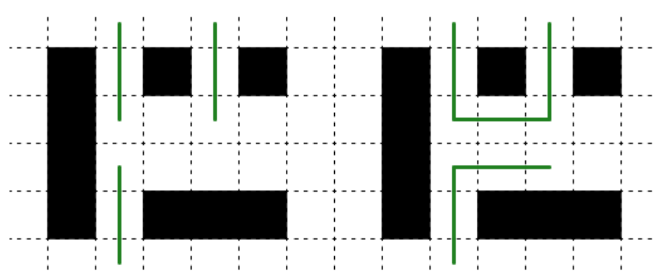
Two cell wide corridor uniqueness:

There are two ways to fill a two-cell-wide corridor. Either a single loop segment zig-zags through it, or two loop segments pass through parallel to each other.

However, we can apply uniqueness whenever we know we have the second case. If the parallel segments were to loop back around and connect to each other, there are multiple ways it can do so and still fill the space, leading to a non-unique puzzle.

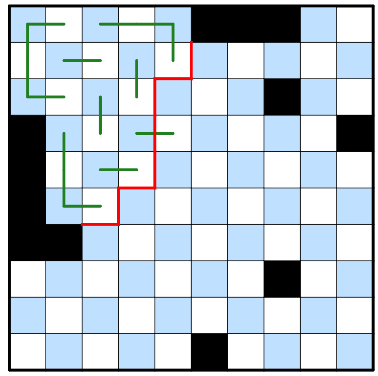


Therefore, if our puzzle is unique, then any two cell wide corridor with two entrances at one end must continue parallel with each other down the corridor until the end of the corridor or until they hit another line segment, giving the uniqueness deduction.

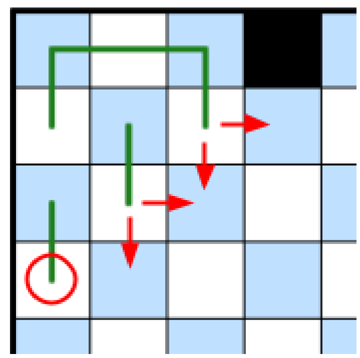


Bouncing uniqueness:

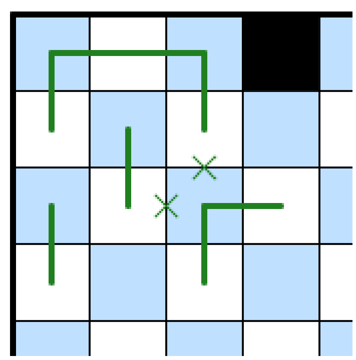
To understand bouncing uniqueness, it helps to view bouncing as a parity deduction. Any bouncing chain can be visualised as a region with an equal number of cells of each parity, and only one possible exit of one parity. This exit is therefore forced, giving the next line segment. However, this also implies that the region can only have one exit of the other parity.



This “other” exit is the key for the uniqueness deduction. Consider the example on the right. The circled cell is the first exit to the region found via bouncing. Therefore there are two other white cells that may be the second exit.

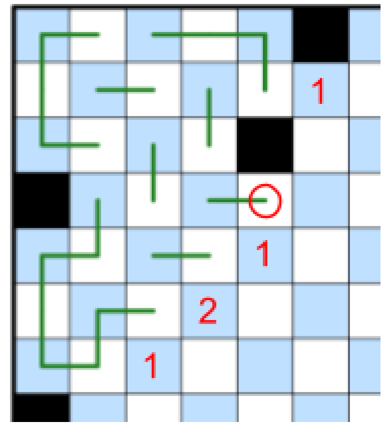


If the second exit was into r3c3, we run into a uniqueness issue: either r2c3 or r3c2 may be the cell that exits. So, by uniqueness, we must not exit into this cell, giving some crosses and line segments.

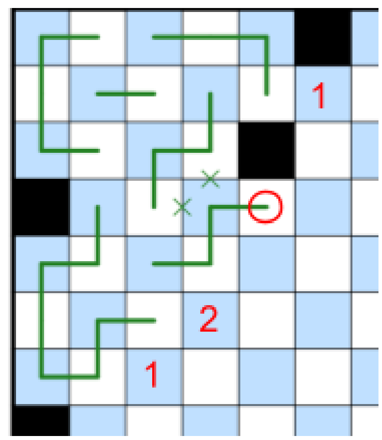


And that’s bouncing uniqueness! However, applying it can be more difficult than it seems, and there’s also a slip-up that can be easy to make. Let’s go over an example case of this.

Here’s a deceptive situation that occurs whenever a bouncing chain is “interrupted” - in this example, by the shaded cell in r3c5. Applying bouncing uniqueness suggests there are 3 second exits that will resolve the chain uniquely. However, one of these exits is fake - r5c5 will not resolve uniquely.

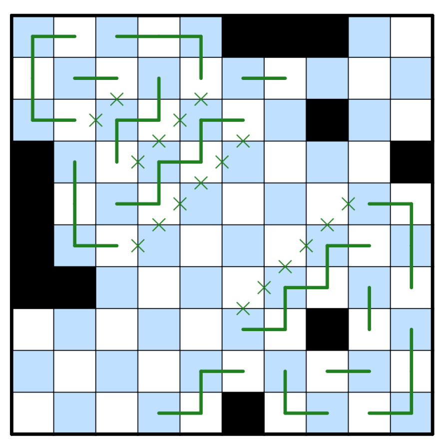


If we apply bouncing uniqueness to the previous layer, we find that r5c4 must connect to r4c4, and therefore cannot connect to r5c5. So, the correct numbering is as shown. Importantly, r6c4 still remains a 2 because r5c4 can still connect to it, but it would break uniqueness if it did.

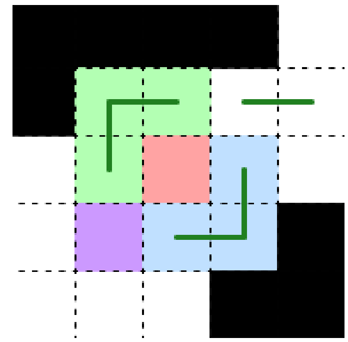


This demonstrates that to apply bouncing uniqueness, it is best to do it in layers, starting from the innermost layer, to avoid such mistakes.

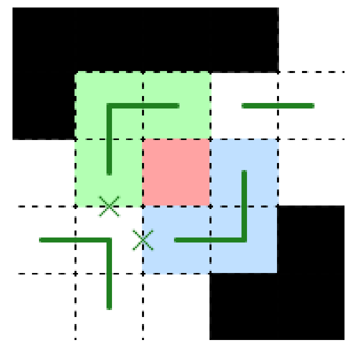
To the right is an example of what bouncing uniqueness looks like when it is applied layer by layer. So far in the solve, all possible bouncing uniqueness deductions have been made.



AB-link uniqueness:



AB-link uniqueness is another case where we can apply logic similar to bouncing uniqueness. In the example on the right, we have an AB-link that is thus known to have exactly two exits. One exit is already known - the bouncing segment in the top right. So one more exit is needed.



If the purple cell is an exit, then either the green or blue segment can exit into it, resulting in a non-unique puzzle. Therefore, by uniqueness, we can mark a couple of crosses and line segments.