

LABELING NONOGRAMS

Maarten Löffler
Utrecht University

Martin Nöllenburg
TU Vienna

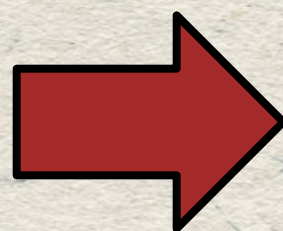
LABELING WHAT?

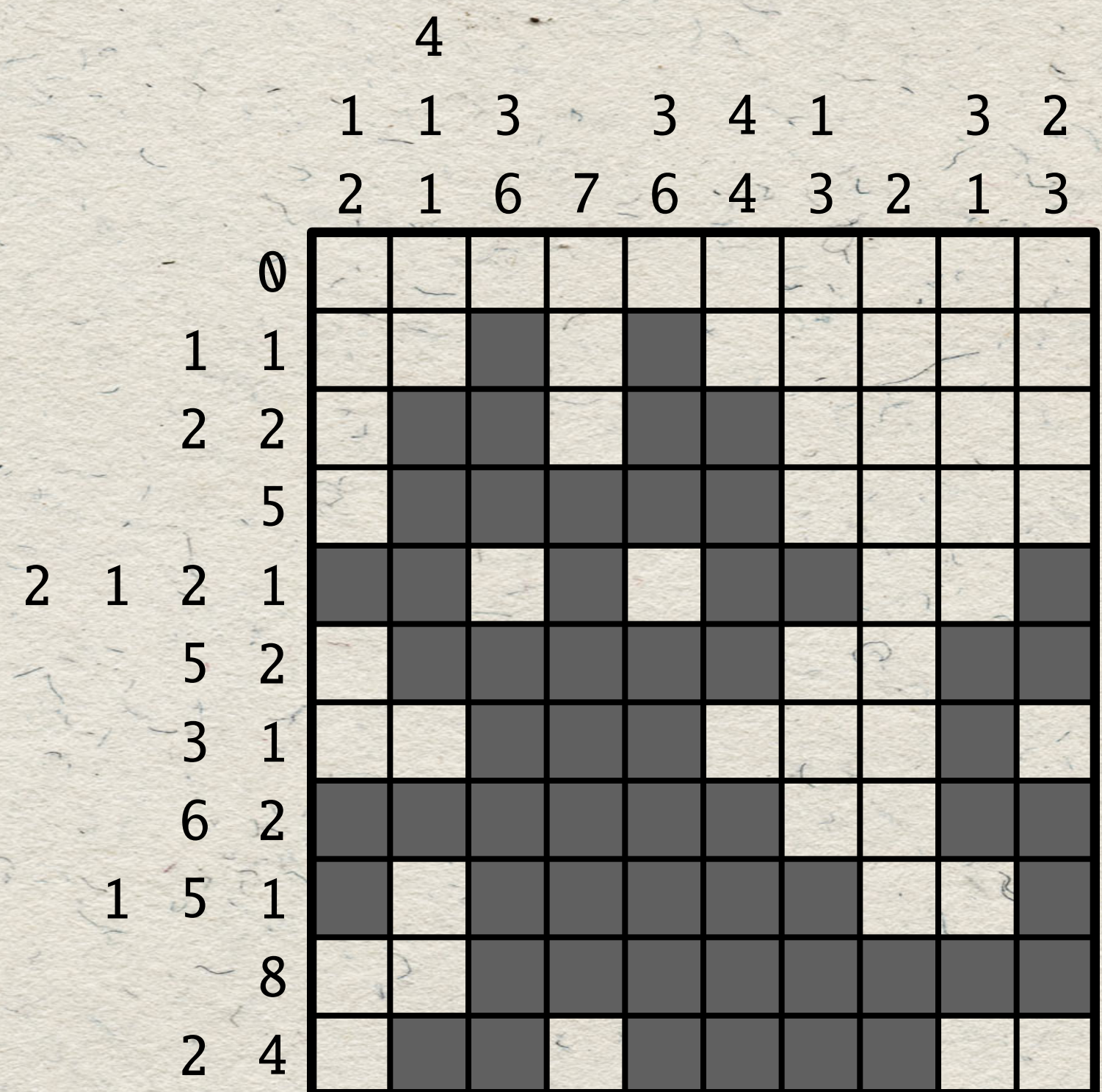
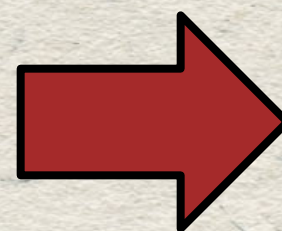
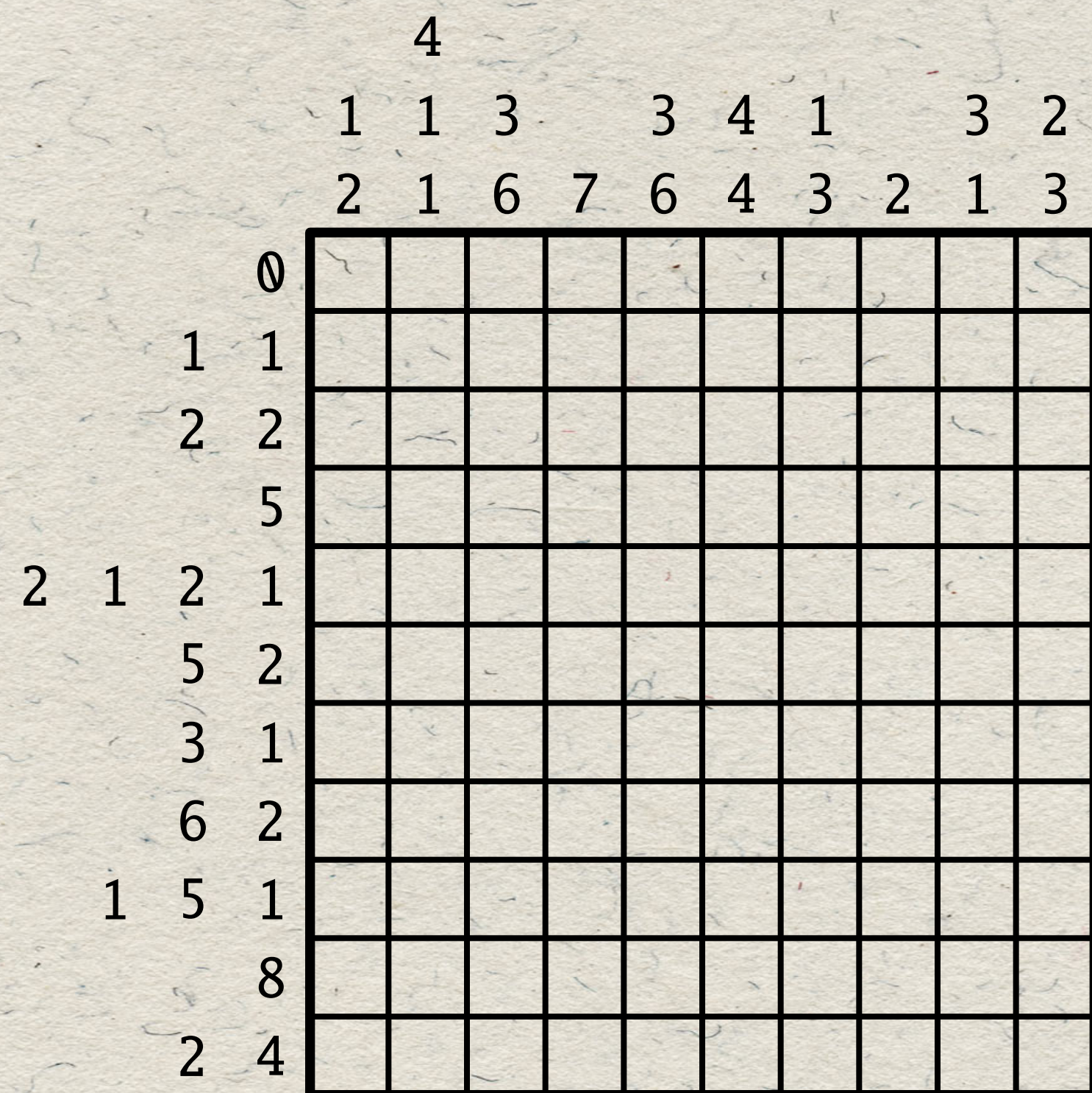
4
1 1 3 3 4 1 3 2
2 1 6 7 6 4 3 2 1 3

	0									
	1	1								
	2	2								
	5									
2	1	2	1							
	5	2								
	3	1								
	6	2								
1	5	1								
	8									
2	4									

4
1 1 3 3 4 1 3 2
2 1 6 7 6 4 3 2 1 3

	0									
1	1									
2	2									
5										
2 1	2 1									
5	2									
3	1									
6	2									
1	5 1									
8										
2	4									





Design and Automated Generation of Japanese Picture Puzzles

M. van de Kerkhof¹, T. de Jong², R. Parment³, M. Löffler^{†1}, A. Vaxman¹, and M. van Kreveld^{‡1}

¹Dept. of Information and Computing Sciences, Utrecht University, the Netherlands

²Milvum, the Netherlands

³Macq, Belgium

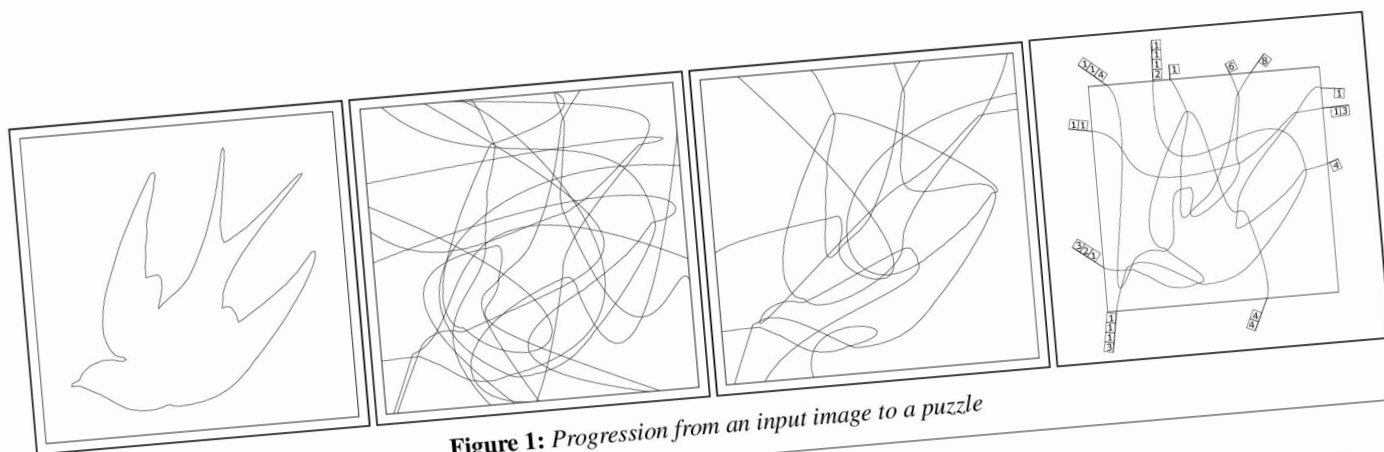


Figure 1: Progression from an input image to a puzzle

Abstract

We introduce the generalized nonogram, an extension of the well-known nonogram or Japanese picture puzzle. It is not based on a regular square grid but on a subdivision (arrangement) with differently shaped cells, bounded by straight lines or curves. To generate a good, clear puzzle from a filled line drawing, the arrangement that is formed for the puzzle must meet a number of criteria. Some of these relate to the puzzle and some to the geometry. We give an overview of these criteria and show that a puzzle can be generated by an optimization method like simulated annealing. Experimentally, we analyze the convergence of the method and the remaining penalty score on several input pictures along with various other design options.

CCS Concepts

• Applied computing → Computer games; • Human-centered computing → Visualization systems and tools;

1. Introduction

An application where design and computation come together is drawing puzzle generation. In a drawing puzzle, the puzzler may need to draw outlines, fill in areas, and use colors. Examples are connect-the-dots puzzles, picture mazes (where the solution path is a picture), and nonograms, also known as Japanese picture puzzles. Nonograms are grid-based puzzles where clues are given for each row and each column of the grid; clues tell which grid cells should be filled to retrieve the hidden picture. This picture is usually black-and-white pixel art, but color versions are also common.

The puzzle aspect comes from the fact that we do not get information on any cell explicitly, only about the whole row and the whole column in which a cell lies. Algorithms to solve nonograms exist, but the problem is generally NP-hard. The puzzle instances given in puzzle booklets and on websites are usually much simpler and can be solved by applying simple reasoning.

In this paper we introduce new types of nonograms that generalize the basic type. We show that puzzles can be constructed based on any set of lines and even of curves. For these new types of generalized nonograms we review the design choices and let these inspire an automated method to generate a nonogram from a drawing. For nonograms based on curves, the curves replace the grid lines of basic nonograms and form an arrangement of cells with varying shape and size. Since rows and columns no longer exist, we need a new way to give clues indicating which cells should be filled by the

[†] Supported by the Netherlands Organisation for Scientific Research on grant no. 614.001.504.

[‡] Supported by the Netherlands Organisation for Scientific Research on grant no. 612.001.651.

Design and Automated Generation of Japanese Picture Puzzles

M. van de Kerkhof¹, T. de Jong², R. Parment³, M. Löffler^{†1}, A. Vaxman^{‡1}, and M. van Kreveld^{‡1}

¹Dept. of Information and Computing Sciences, Utrecht University, the Netherlands
²Milvum, the Netherlands
³Macq, Belgium

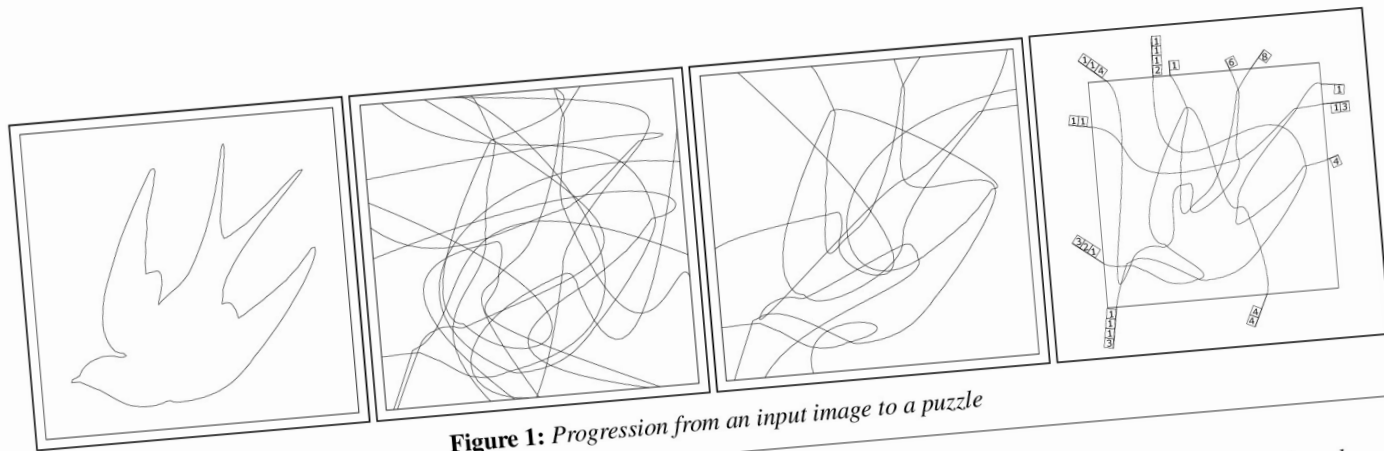


Figure 1: Progression from an input image to a puzzle

Abstract
We introduce the generalized nonogram, an extension of the well-known nonogram or Japanese picture puzzle. It is not based on a regular square grid but on a subdivision (arrangement) with differently shaped cells, bounded by straight lines or curves. To generate a good, clear puzzle from a filled line drawing, the arrangement that is formed for the puzzle must meet a number of criteria. Some of these relate to the puzzle and some to the geometry. We give an overview of these criteria and show that a puzzle can be generated by an optimization method like simulated annealing. Experimentally, we analyze the convergence of the method and the remaining penalty score on several input pictures along with various other design options.

CCS Concepts

• **Applied computing** → Computer games; • **Human-centered computing** → Visualization systems and tools

1. Introduction

An application where design and computation come together is drawing puzzle generation. In a drawing puzzle, the puzzler may need to draw outlines, fill in areas, and use colors. Examples are connect-the-dots puzzles, picture mazes (where the solution path is a picture), and nonograms, also known as Japanese picture puzzles. Nonograms are grid-based puzzles where clues are given for each row and each column of the grid; *clues* tell which grid cells should be filled to retrieve the hidden picture. This picture is usually black-and-white pixel art, but color versions are also common.

[†] Supported by the Netherlands Organisation for Scientific Research on grant no. 614.001.504.

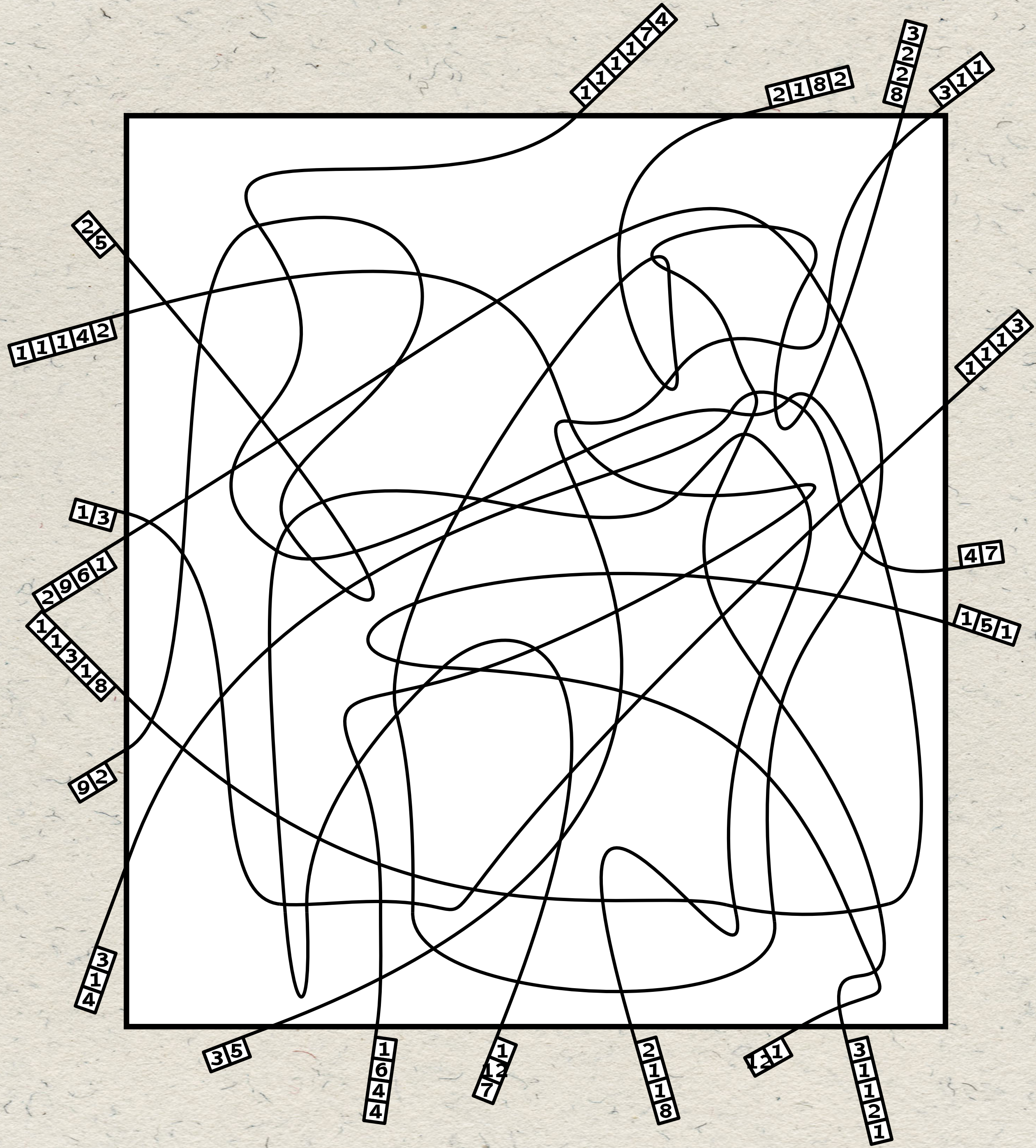
[‡] Supported by the Netherlands Organisation for Scientific Research on grant no. 612.001.651.

The puzzle aspect comes from the fact that we do not get information on any cell explicitly, only about the whole row and the whole column in which a cell lies. Algorithms to solve nonograms exist, but the problem is generally NP-hard. The puzzle instances given in puzzle booklets and on websites are usually much simpler and can be solved by applying simple reasoning.

In this paper we introduce new types of nonograms that generalize the basic type. We show that puzzles can be constructed based on any set of lines and even of curves. For these new types of generalized nonograms we review the design choices and let these inspire an automated method to generate a nonogram from a drawing. For nonograms based on curves, the curves replace the grid lines of basic nonograms and form an *arrangement* of cells with varying shape and size. Since rows and columns no longer exist, we need a new way to give clues indicating which cells should be filled by the

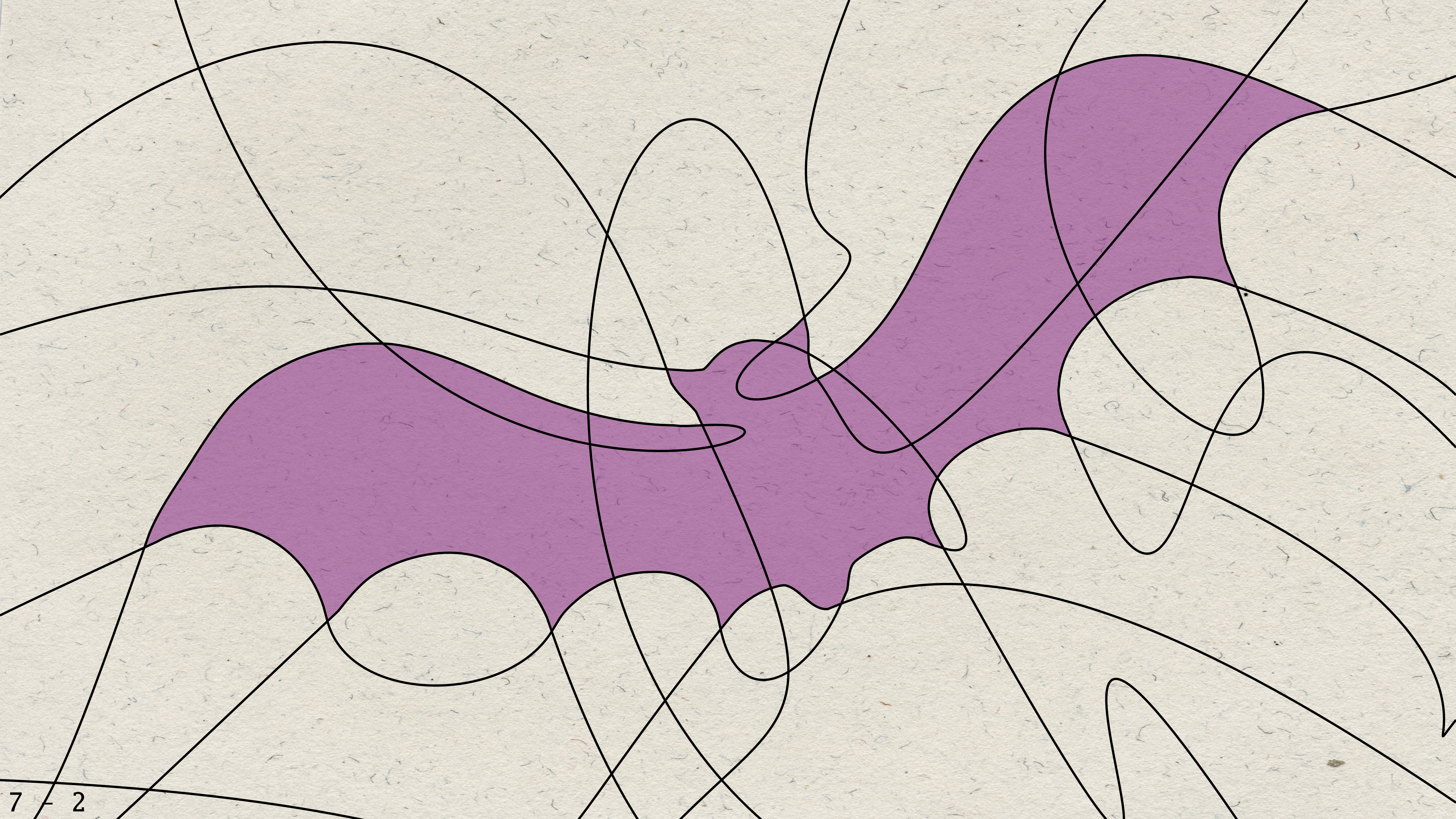
CURVED NONOGRAMS

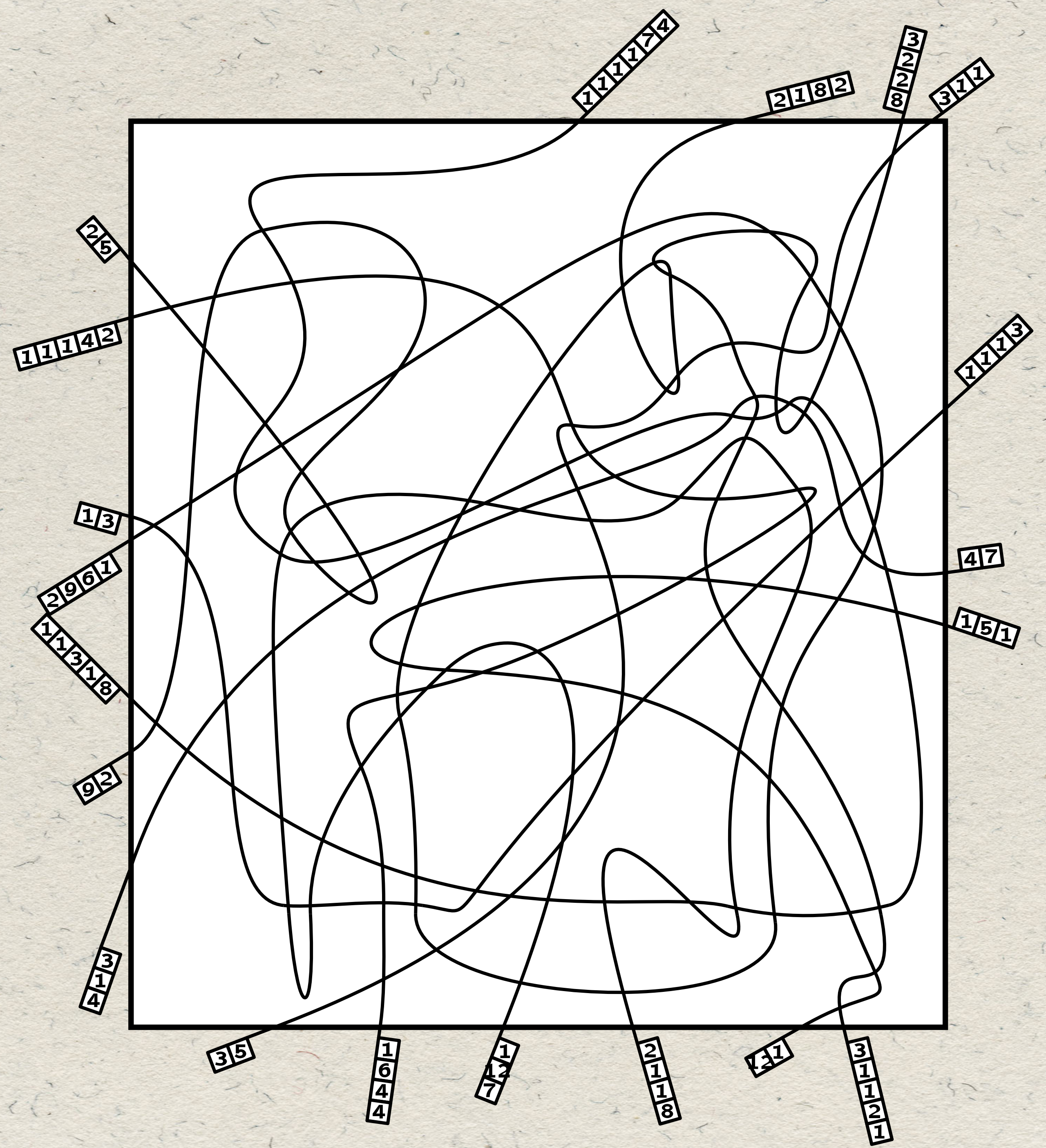
Mees van de Kerkhof
Tim de Jong
Raphael Parment
Maarten Löffler
Amir Vaxman
Marc van Kreveld

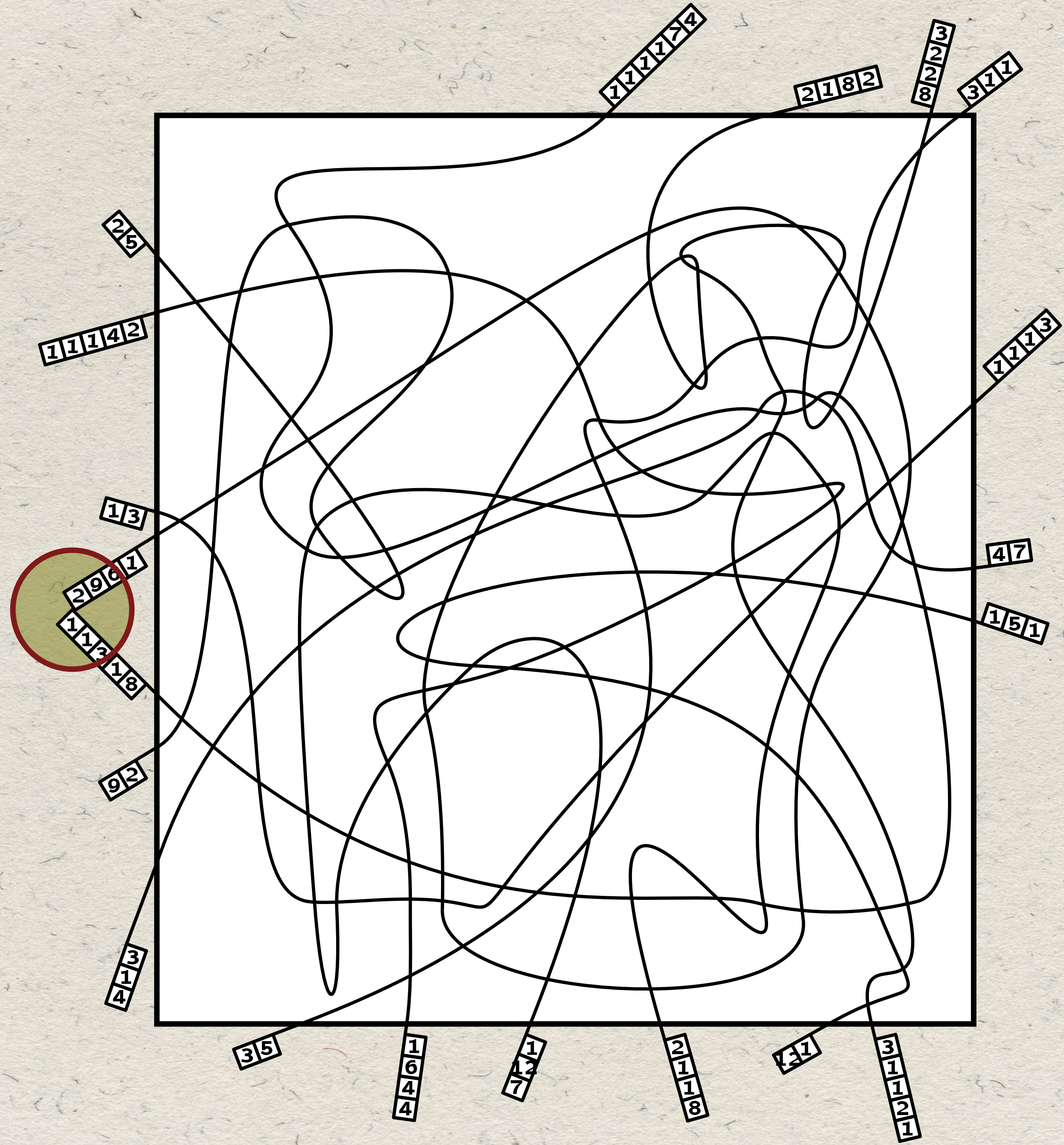


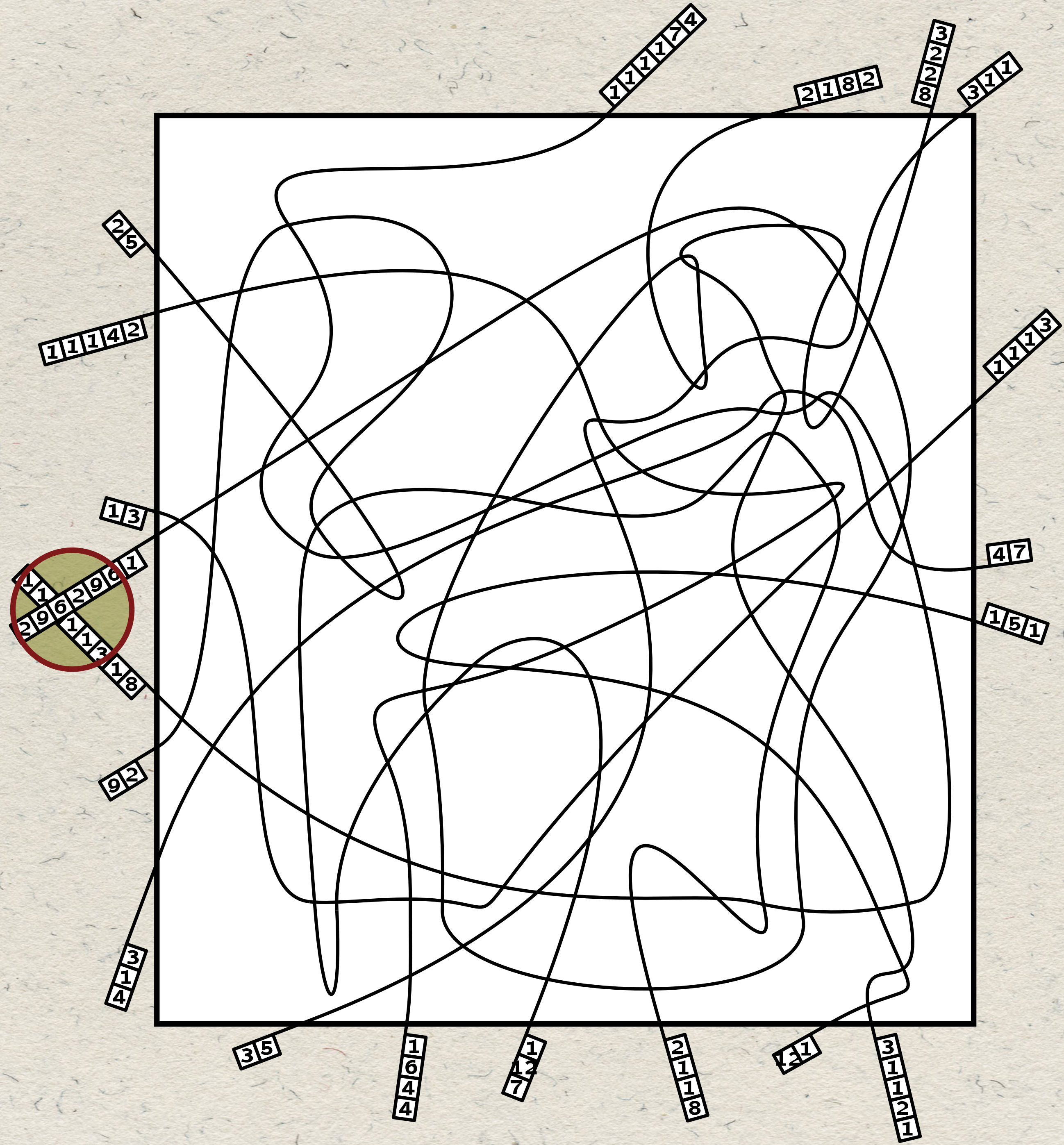
~~~~ THE RULES ~~~~

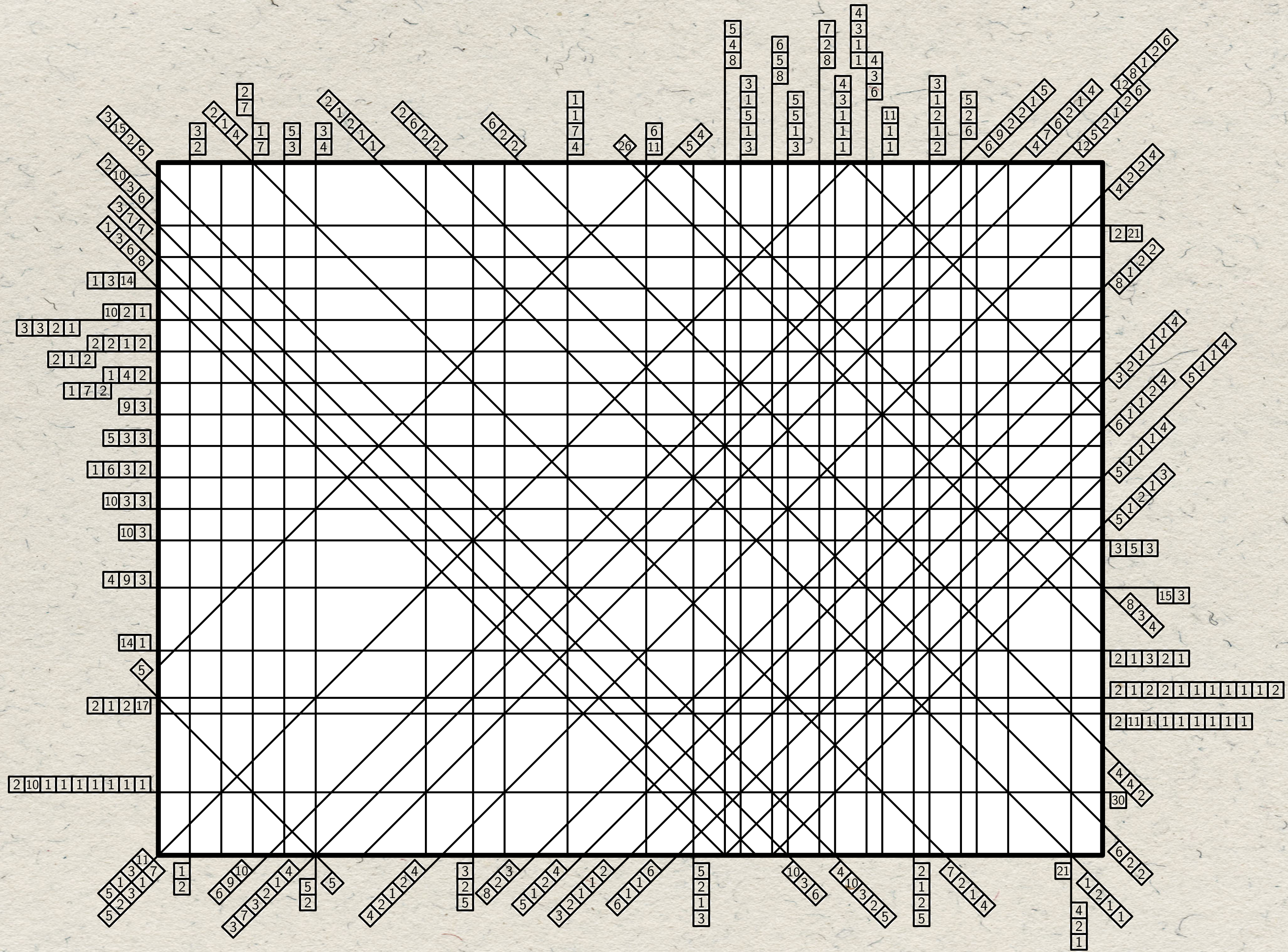
1. Thou shalt either fill one cell completely, or leave it untouched.
2. Thou shalt follow each sequence of cells incident to a common curve.
3. Thou shalt count the number of adjacent cells that thou filleth.
4. Thou shalt leave at least one cell empty between two such sequences.
5. Thou shalt match the numbers thus obtained with the given clues.
6. Thou shalt fill cells such as to satisfy all clues simultaneously.
7. Thou shalt not get confused by multiply adjacent cells.
8. Thou shalt allow for certain curves to be closed or otherwise unclued.
9. Thou shalt verify that thou hath found the unique solution.
10. Thou shalt step back and admire thine result.



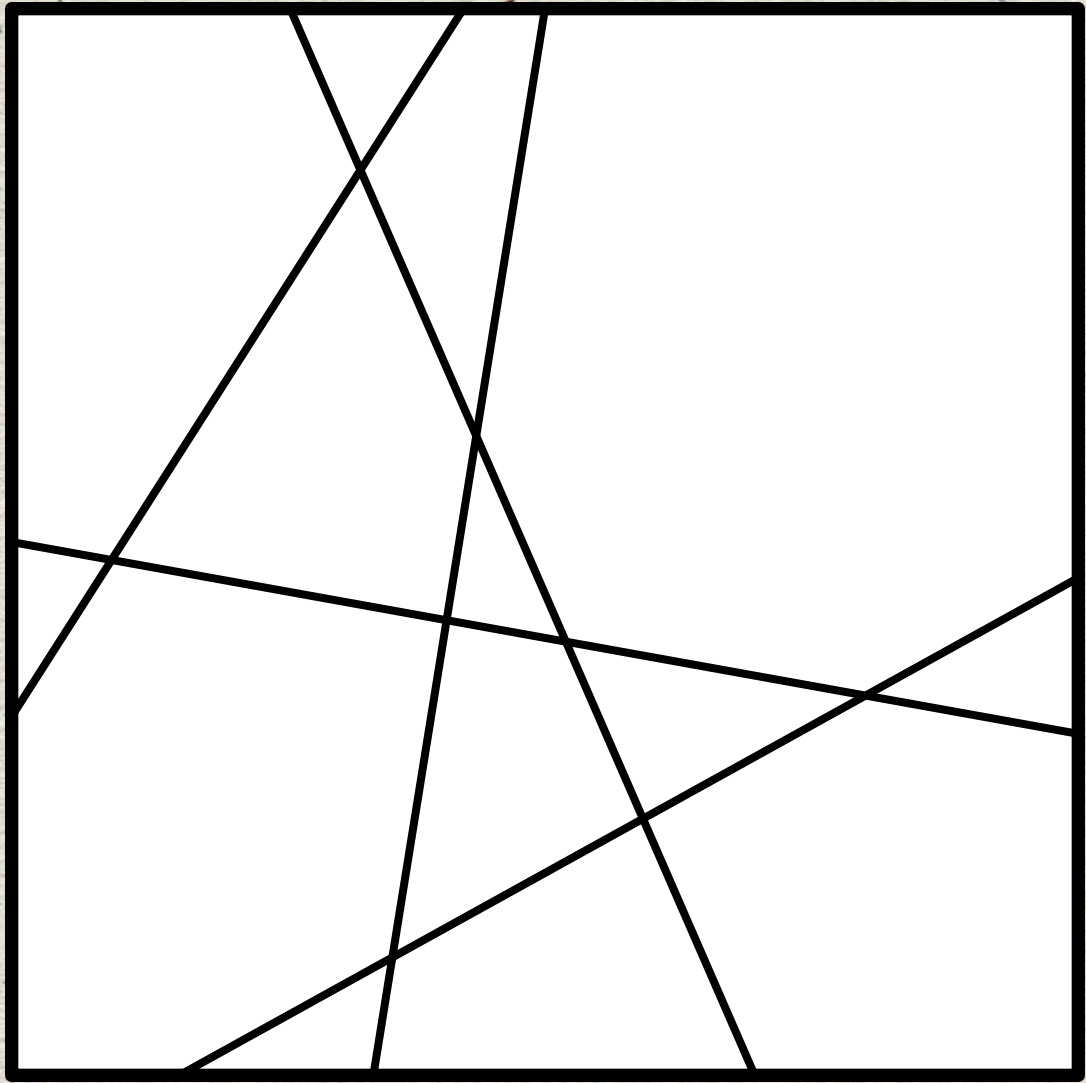


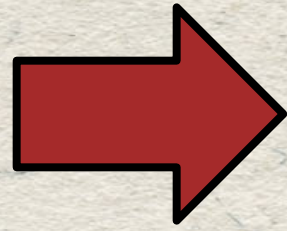
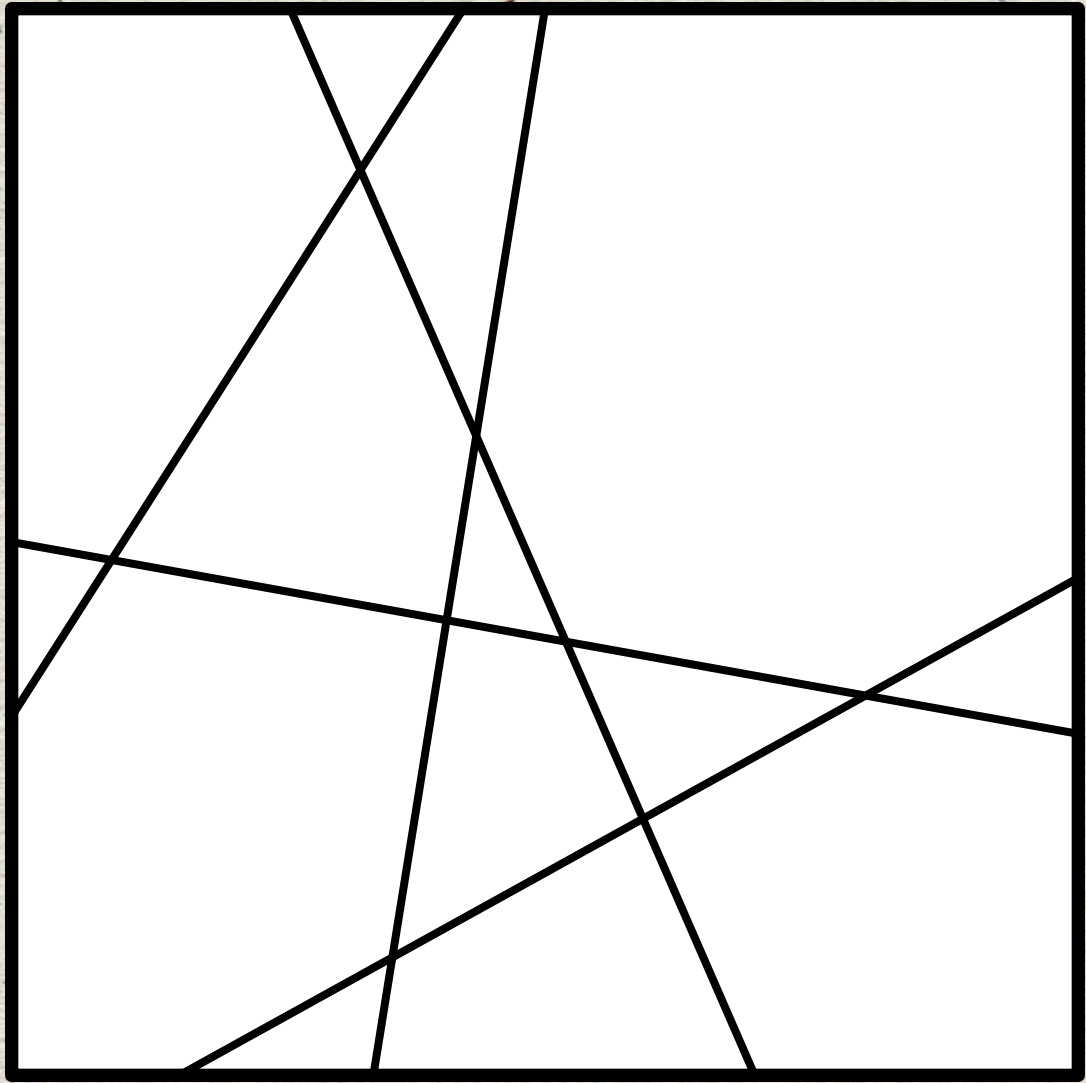


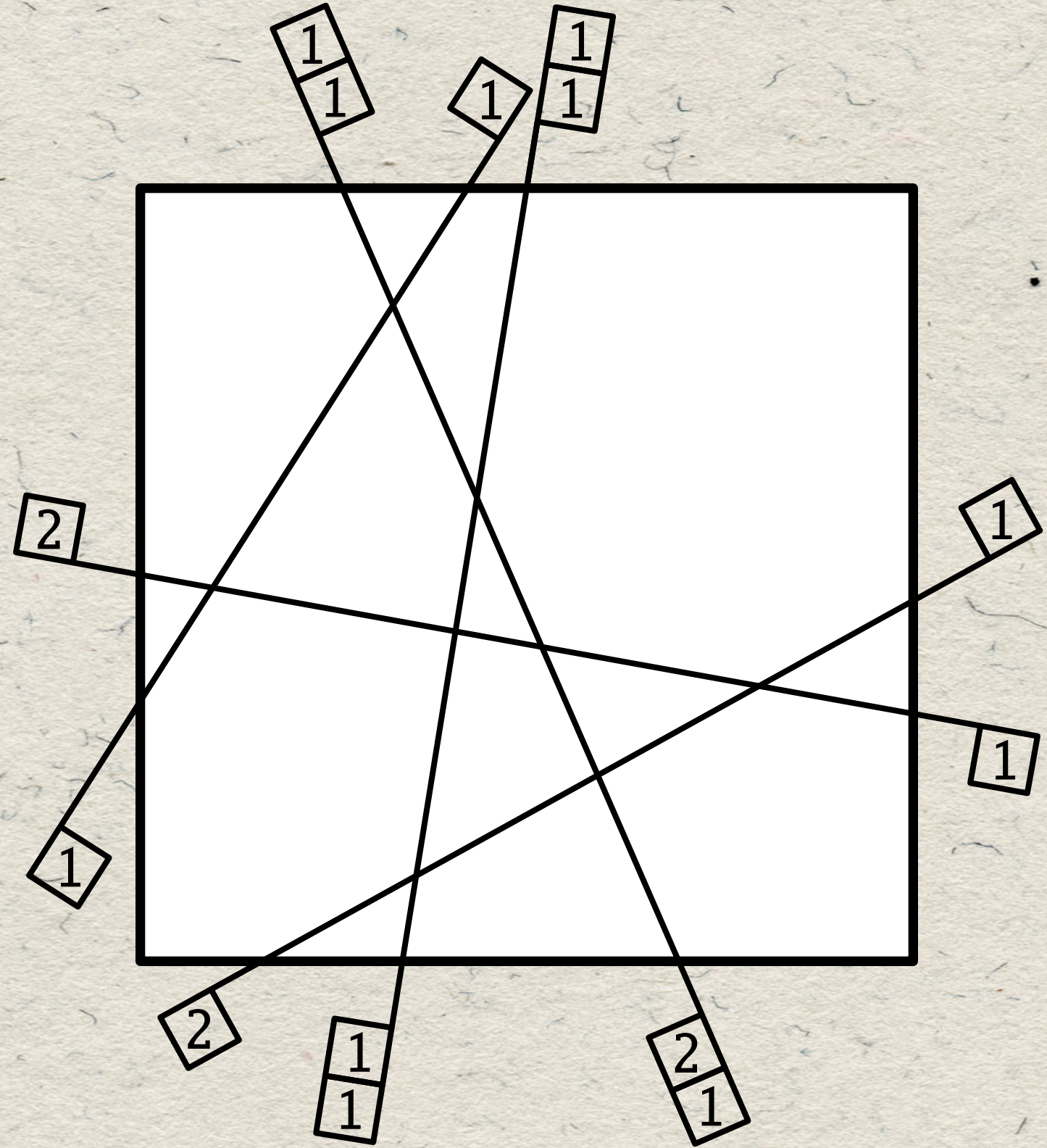
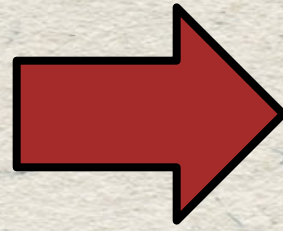
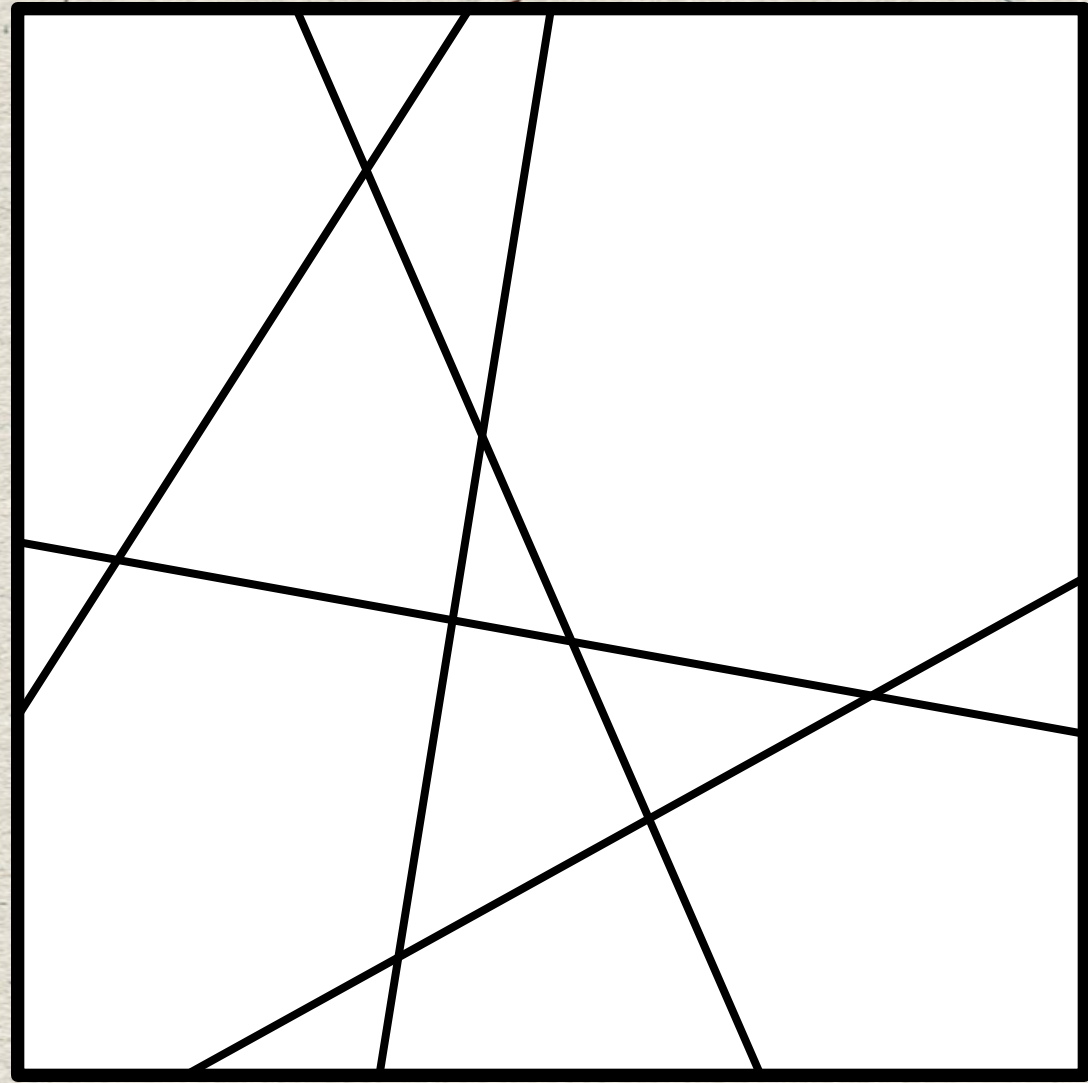


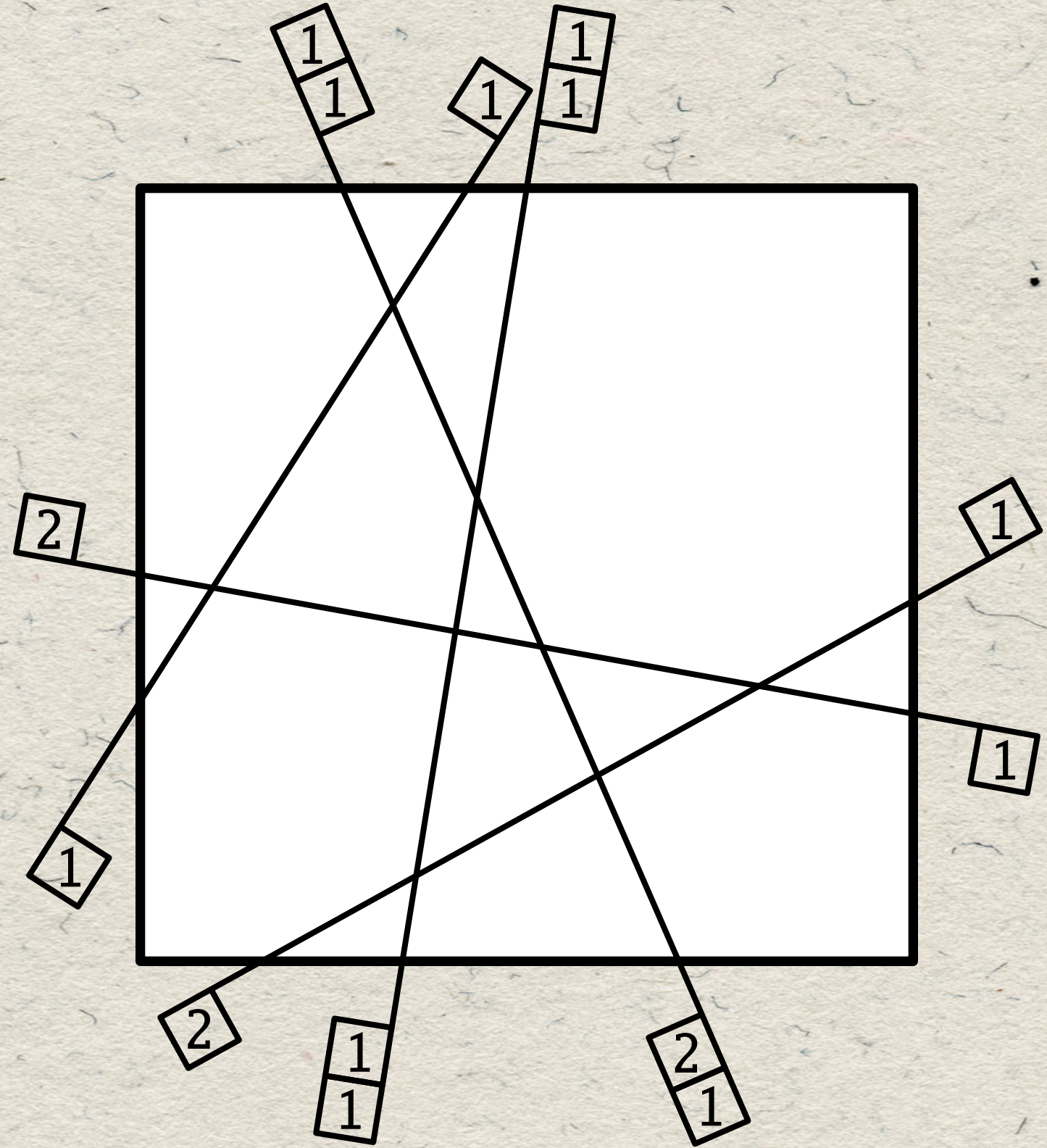
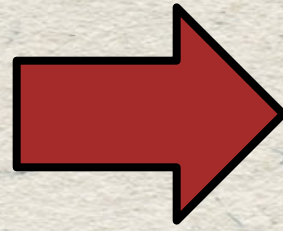
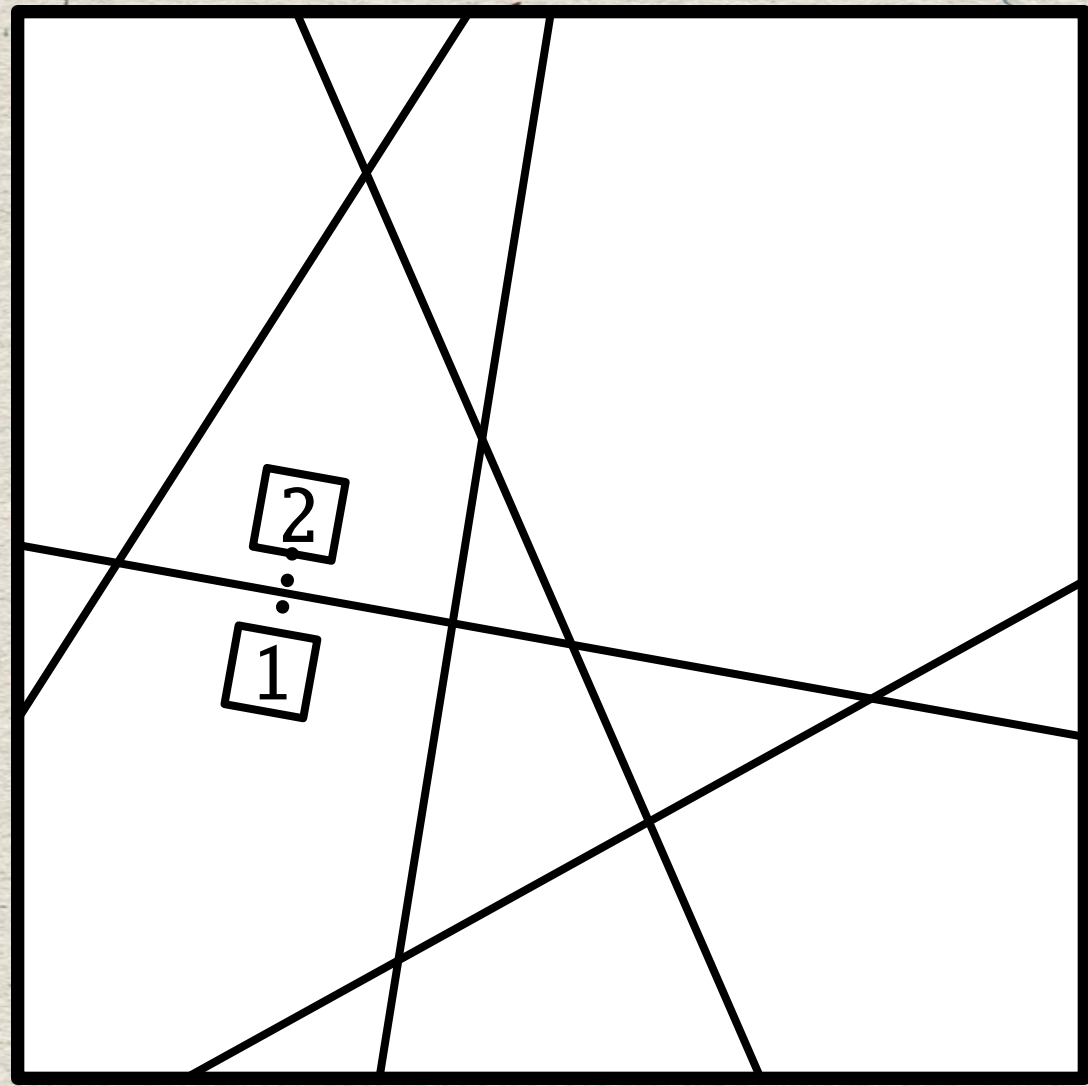


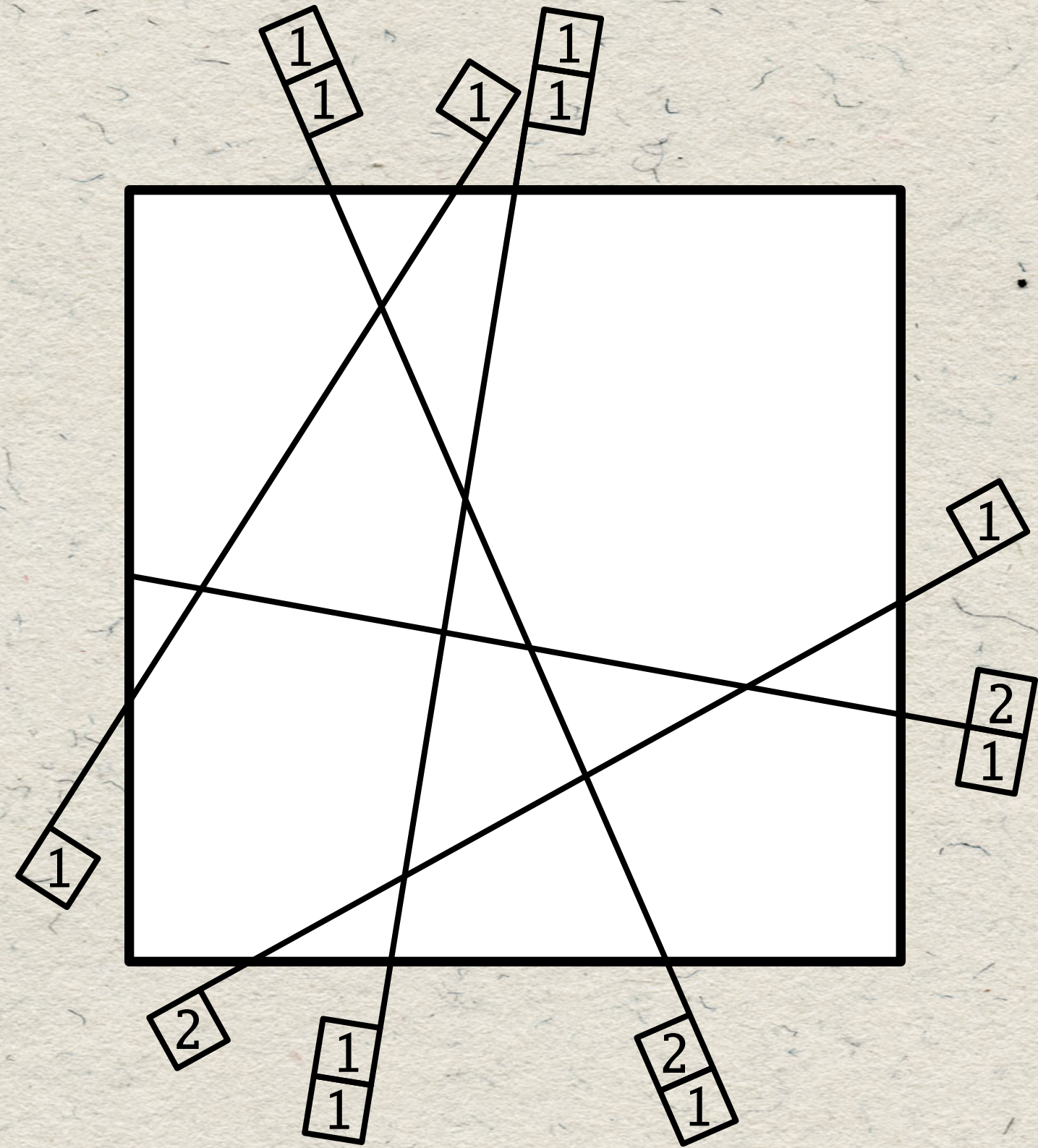
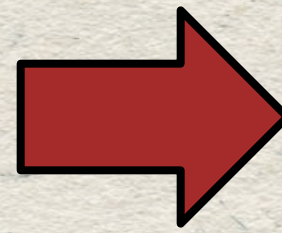
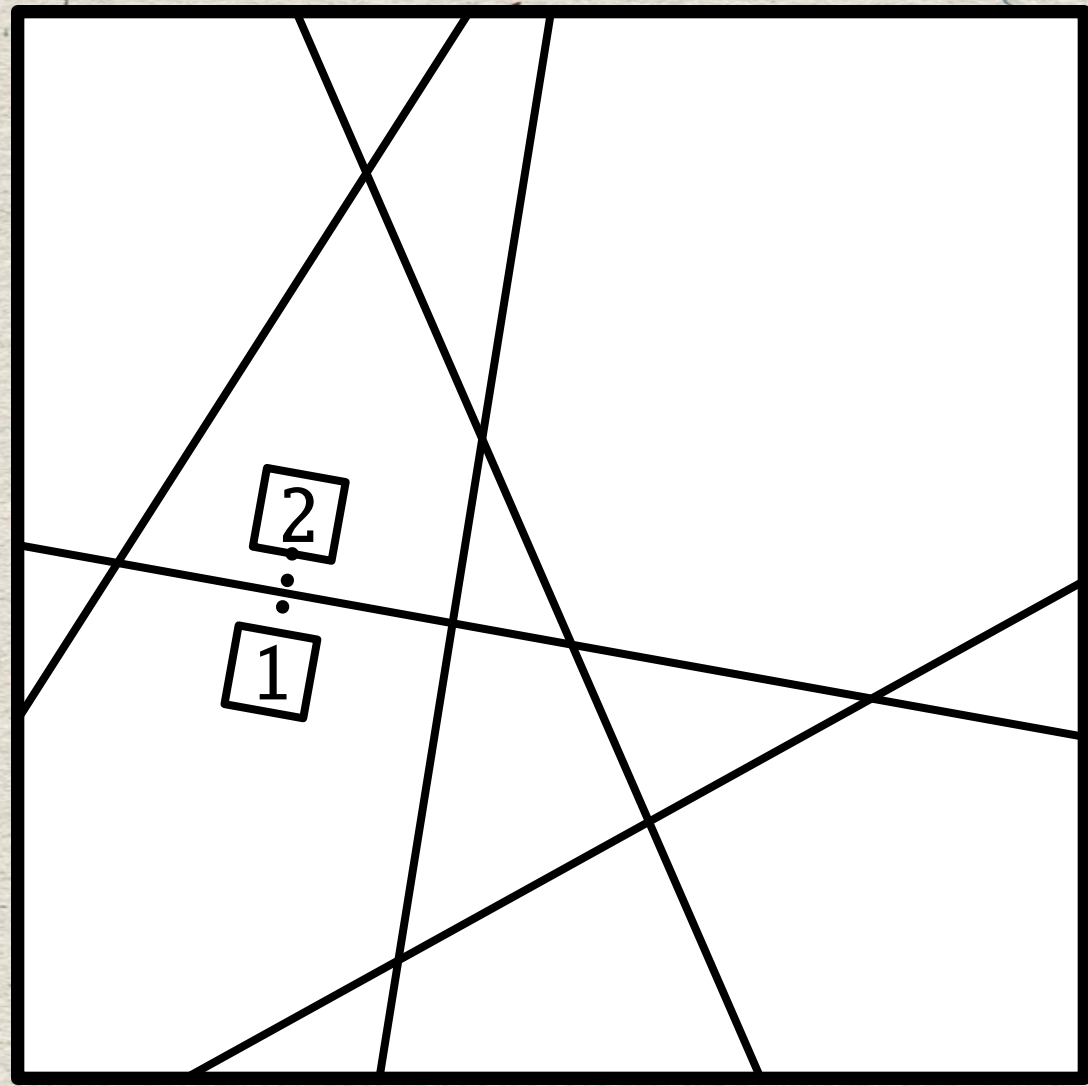
THE PROBLEM

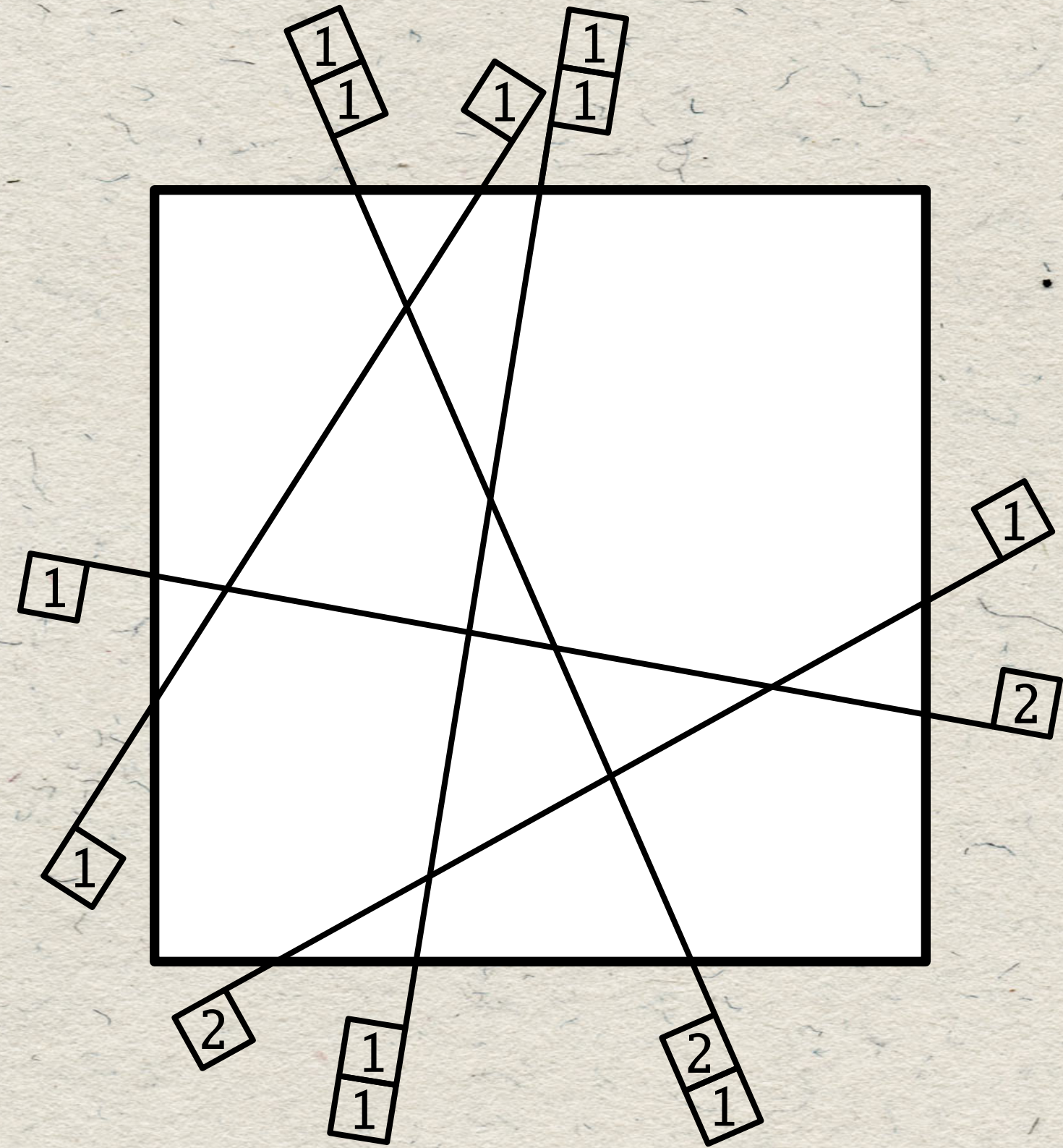
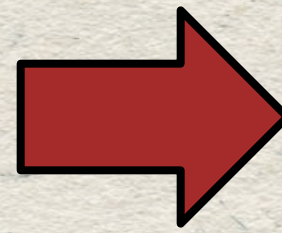
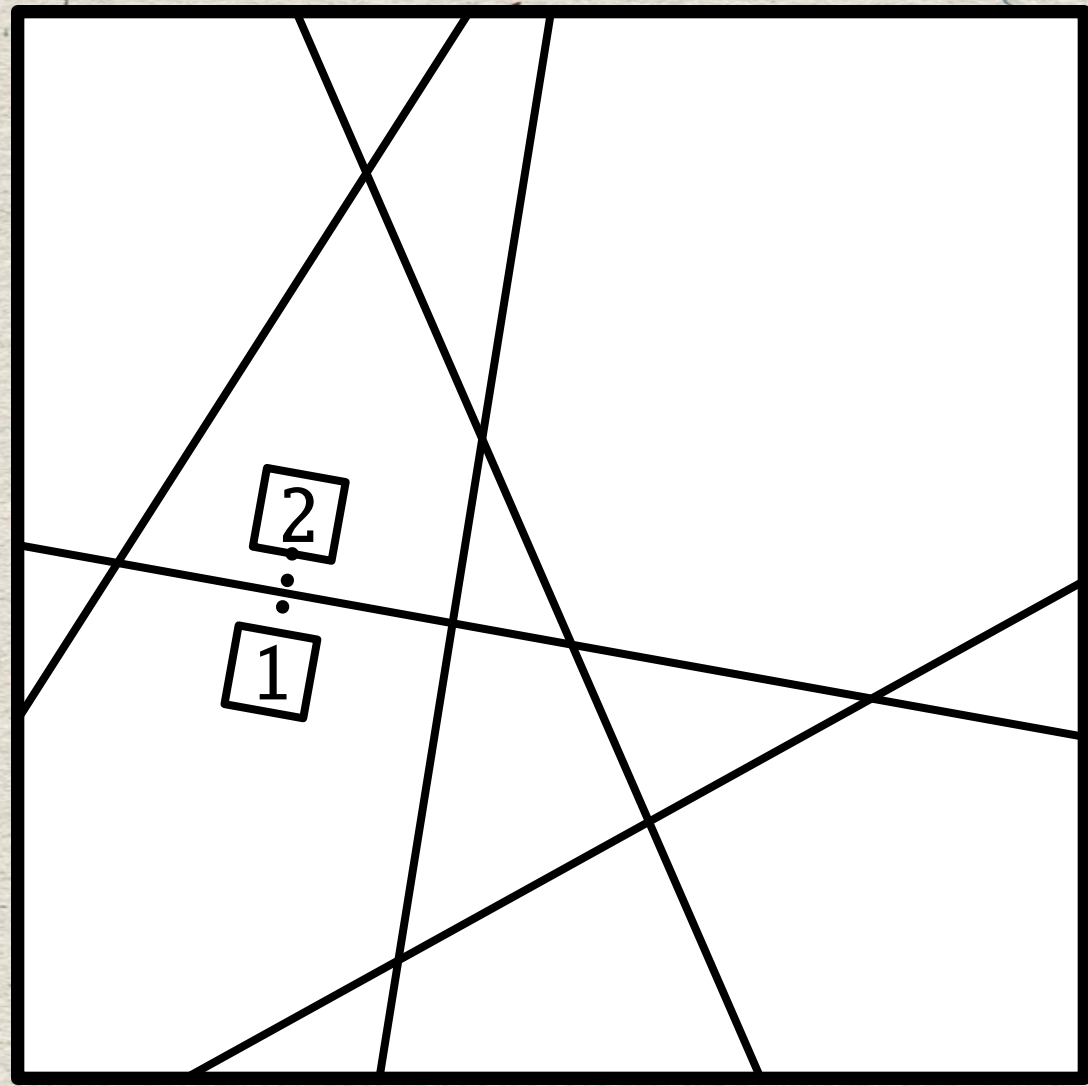


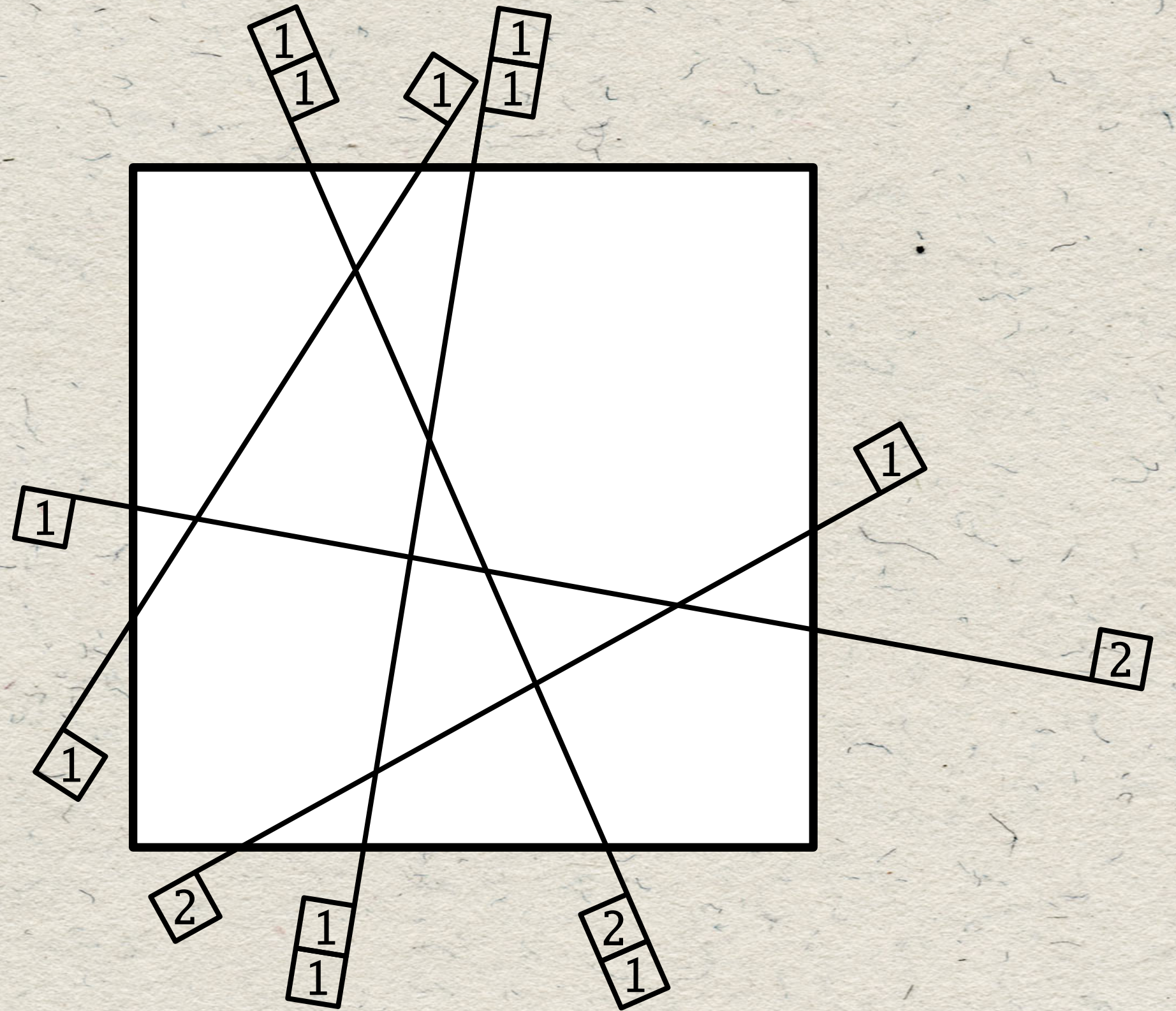
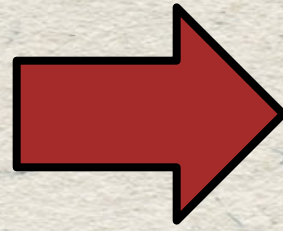
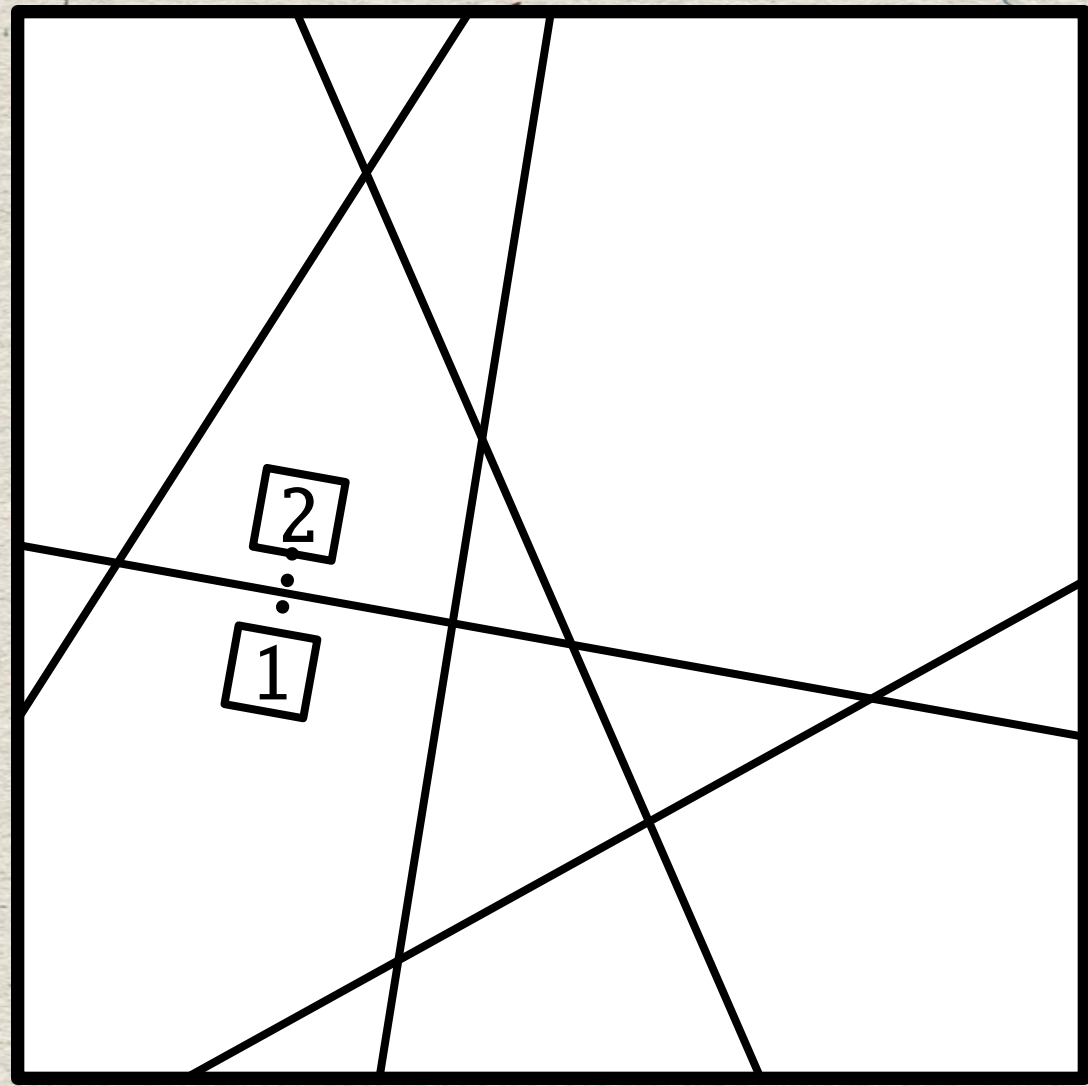


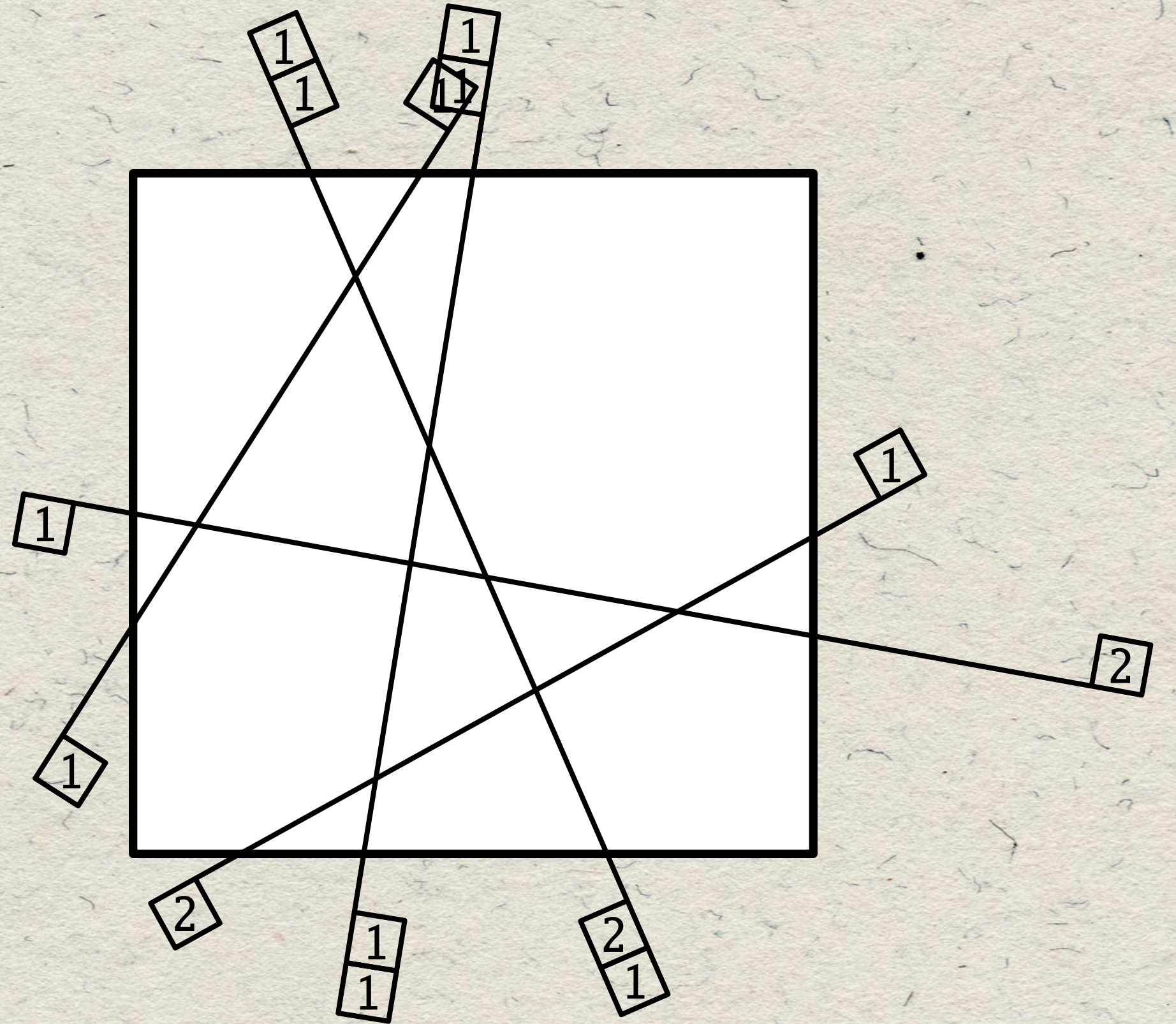
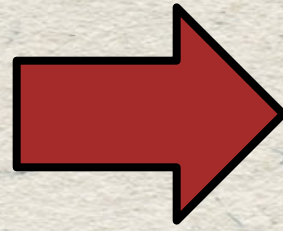
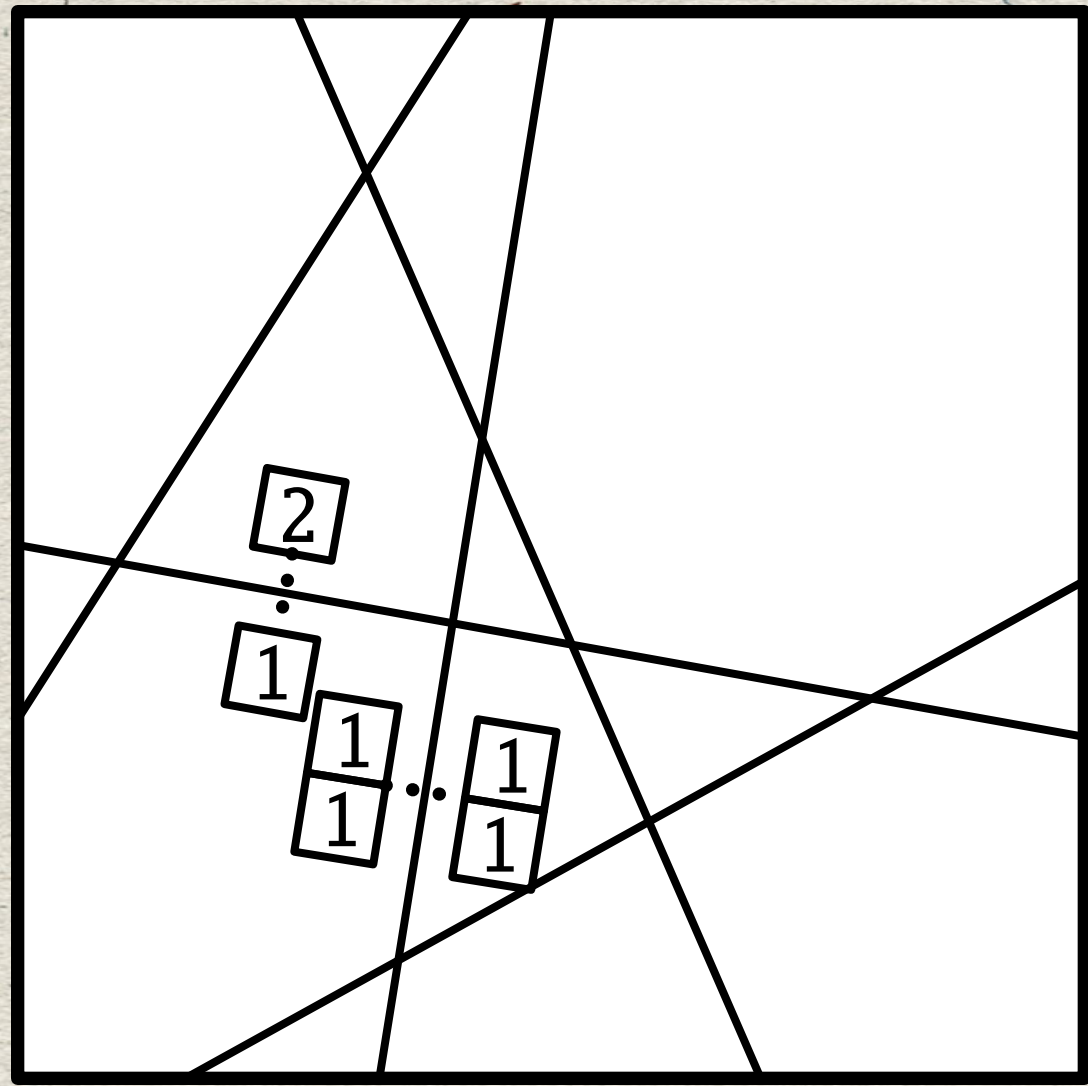


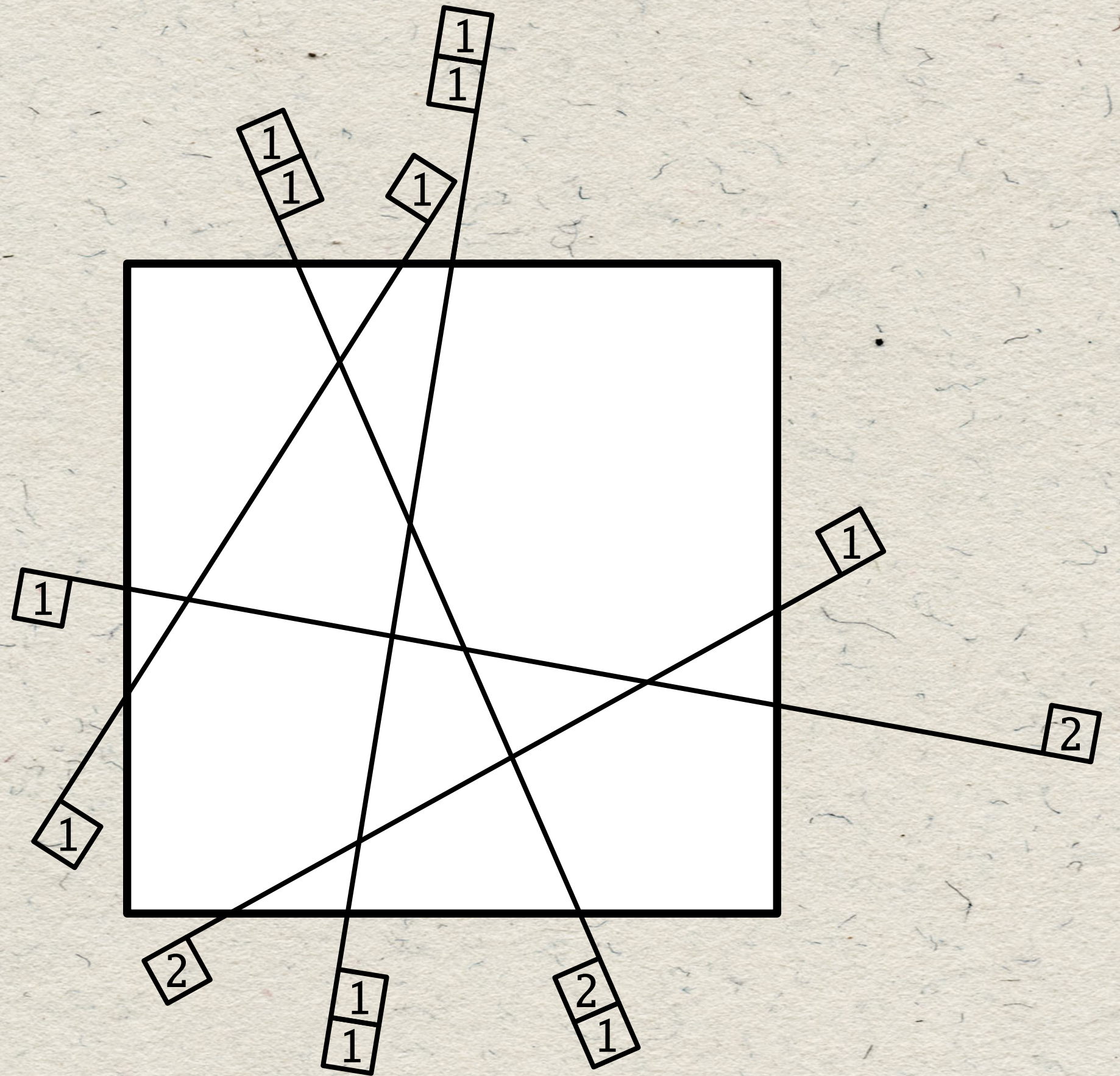
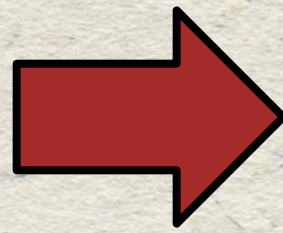
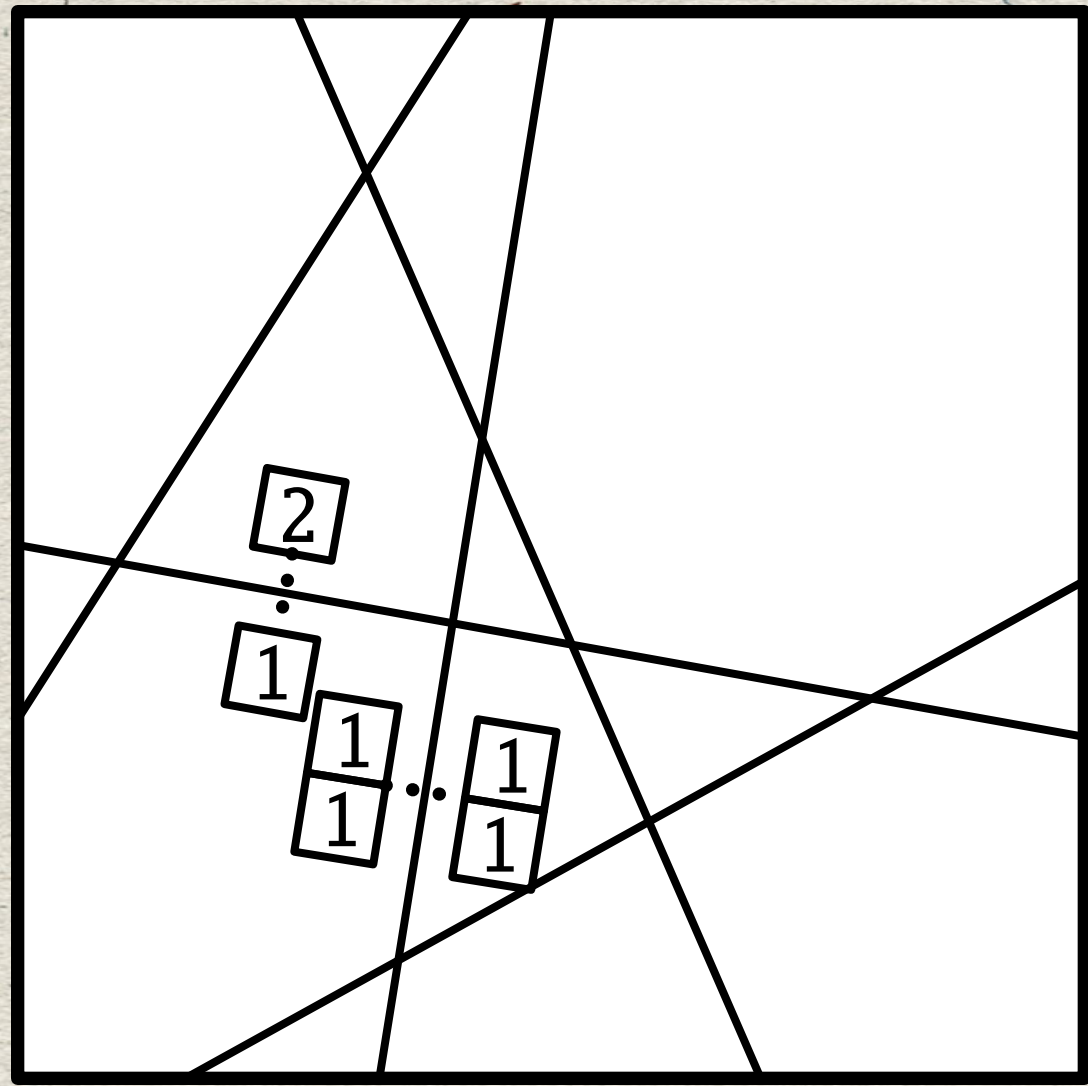


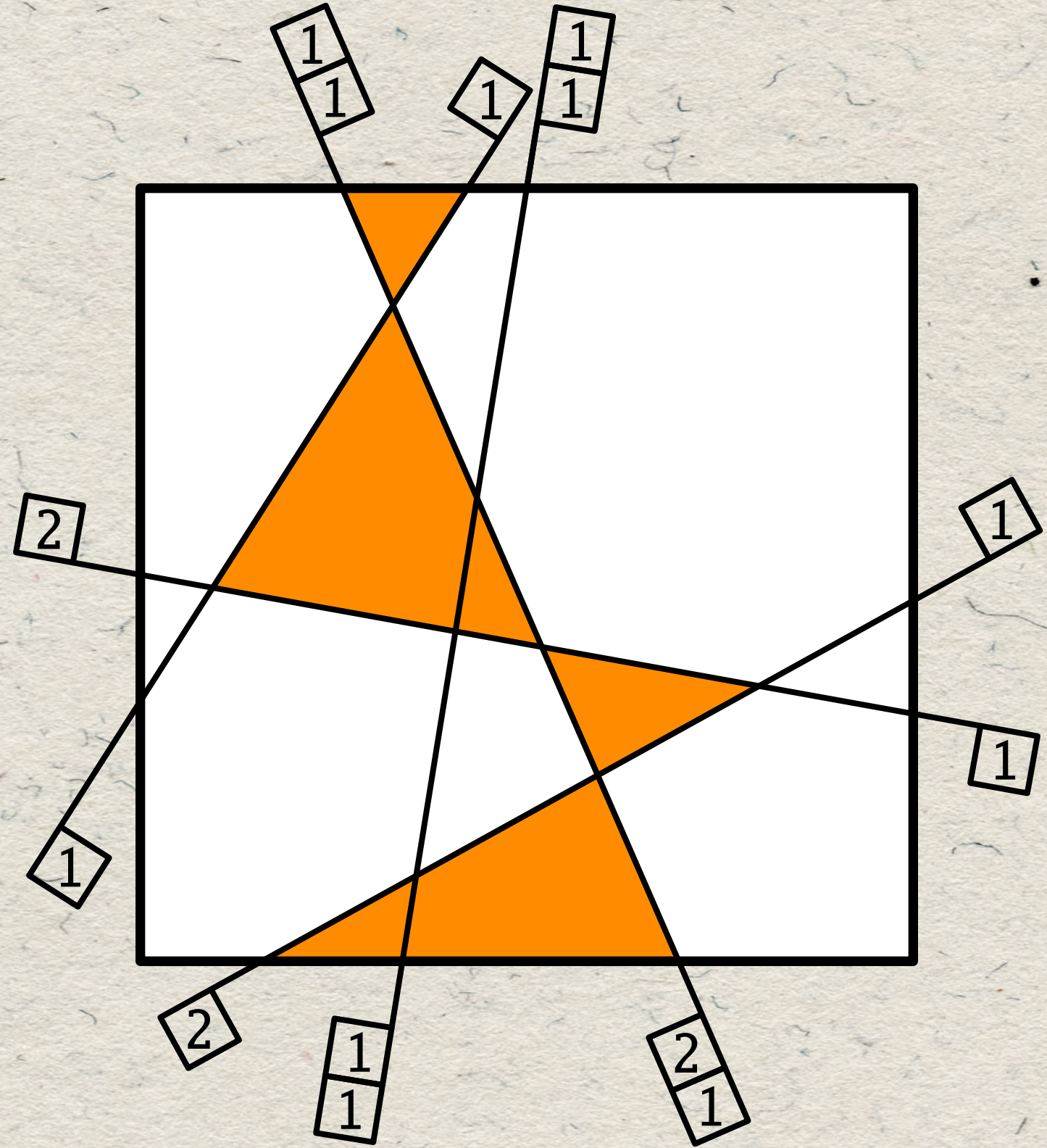
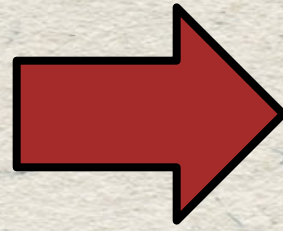
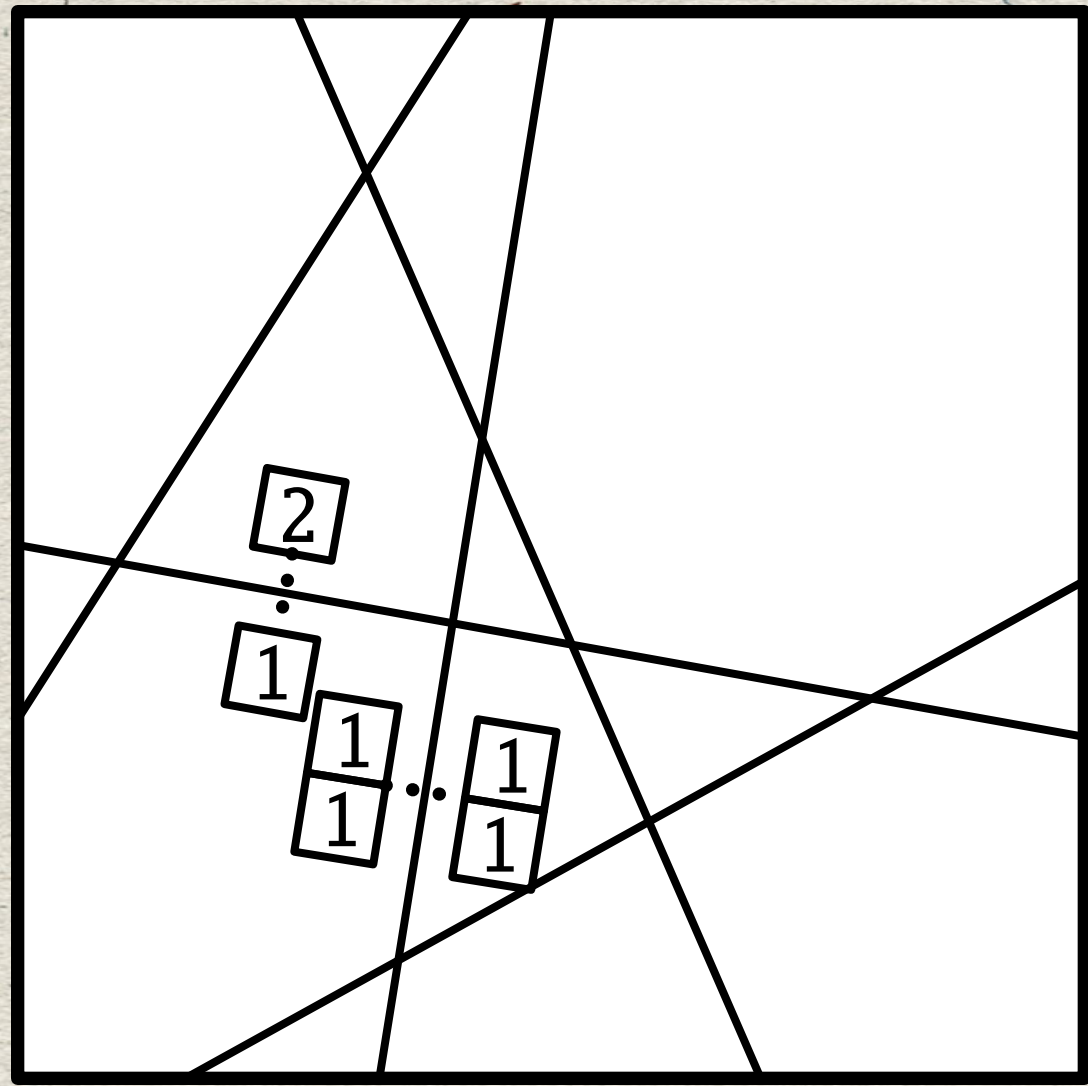


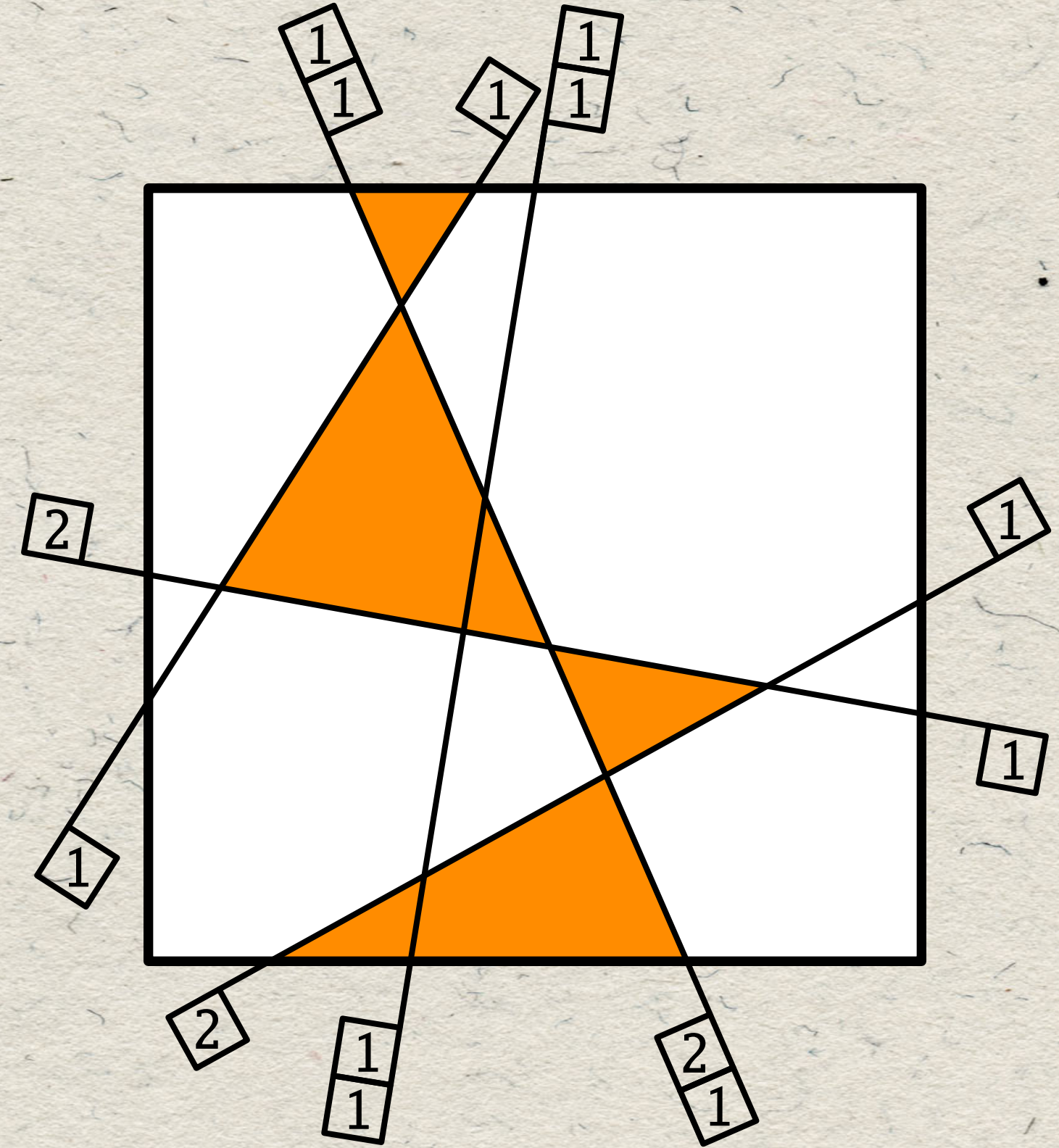
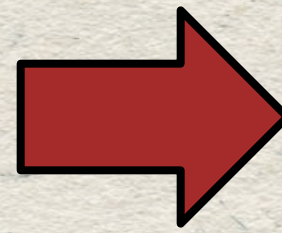
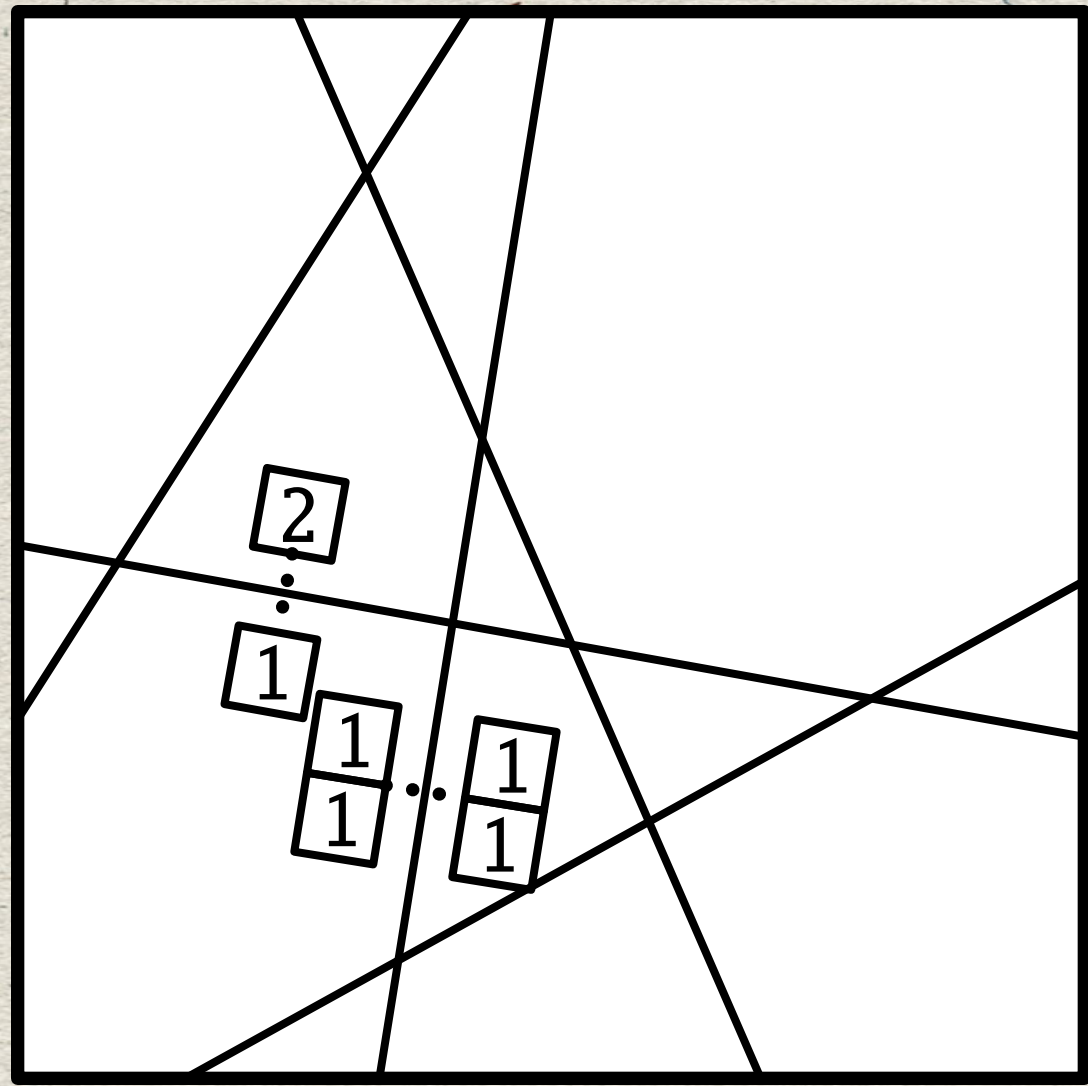


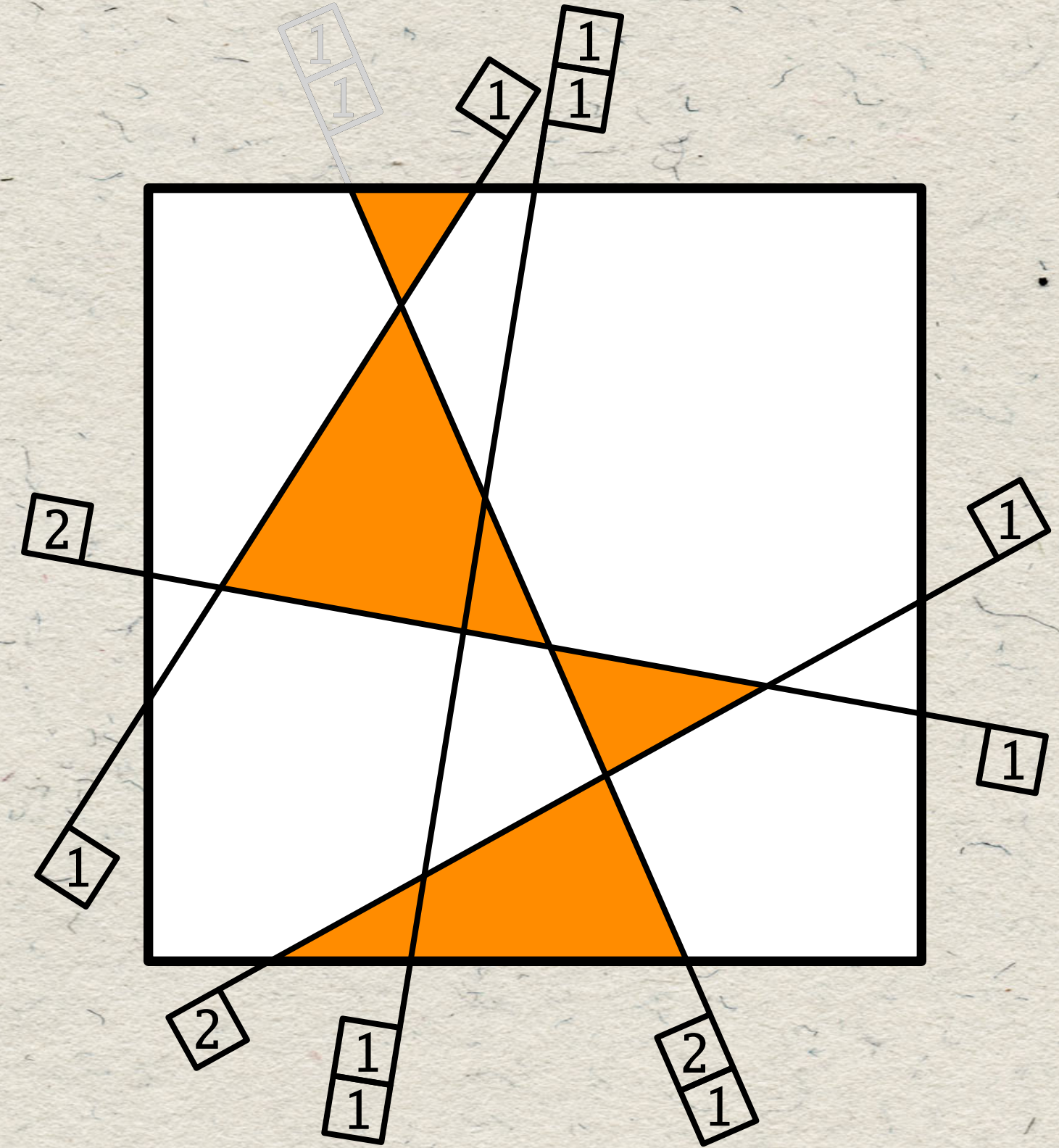
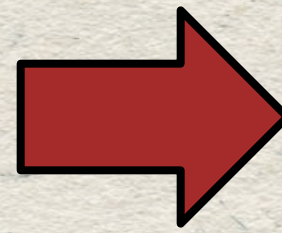
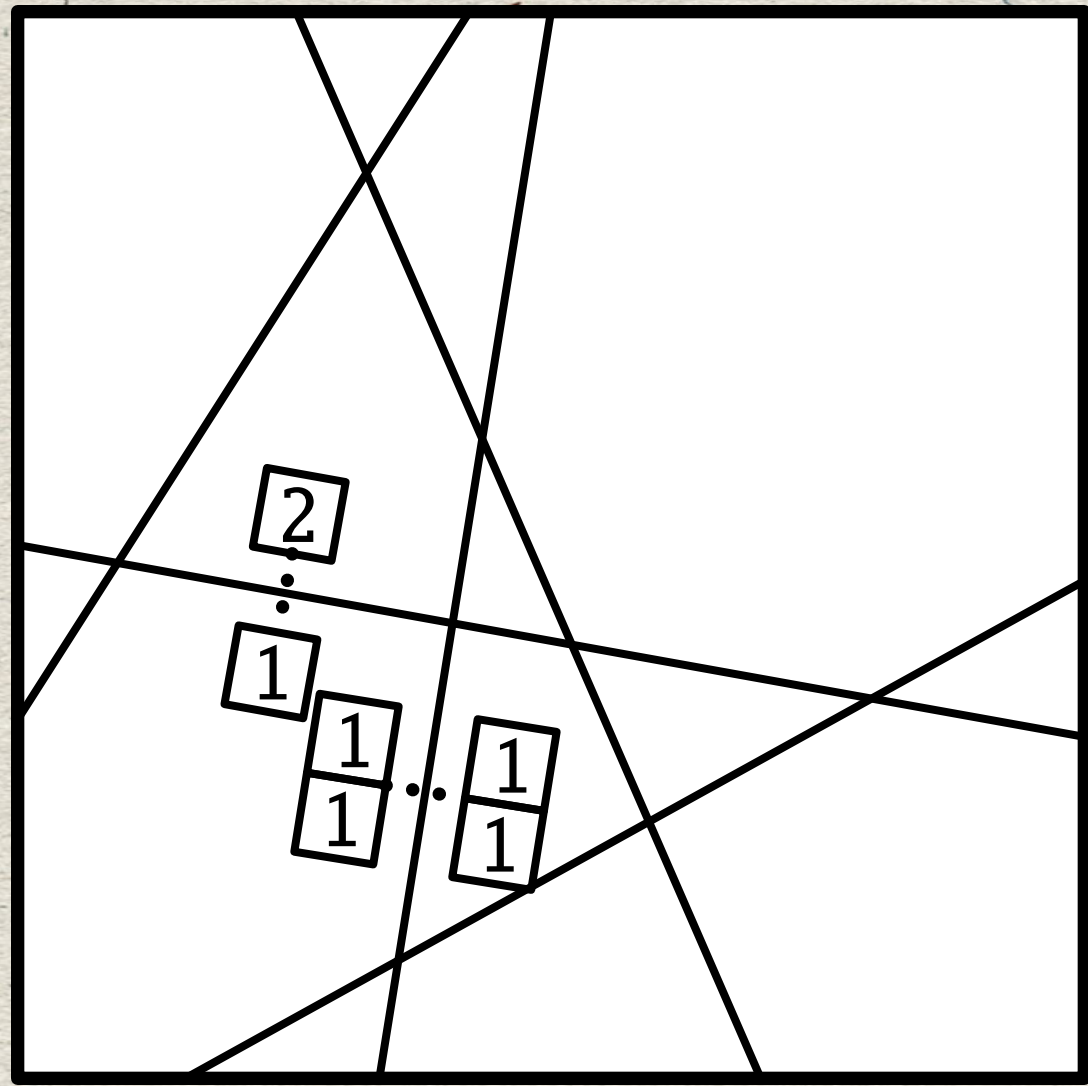


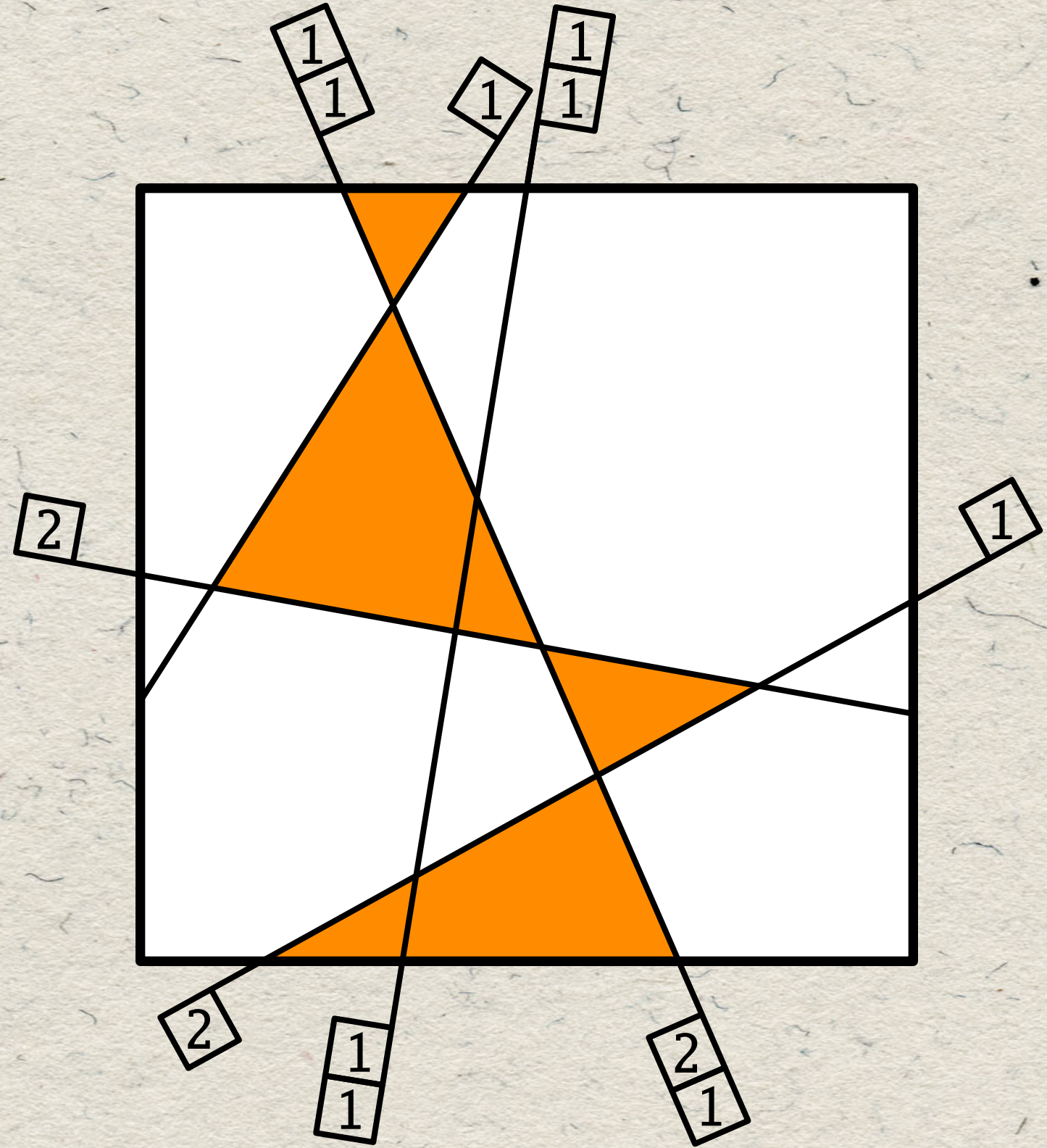
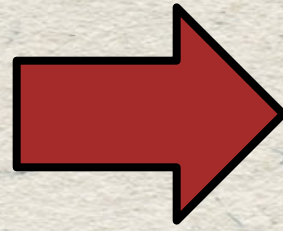
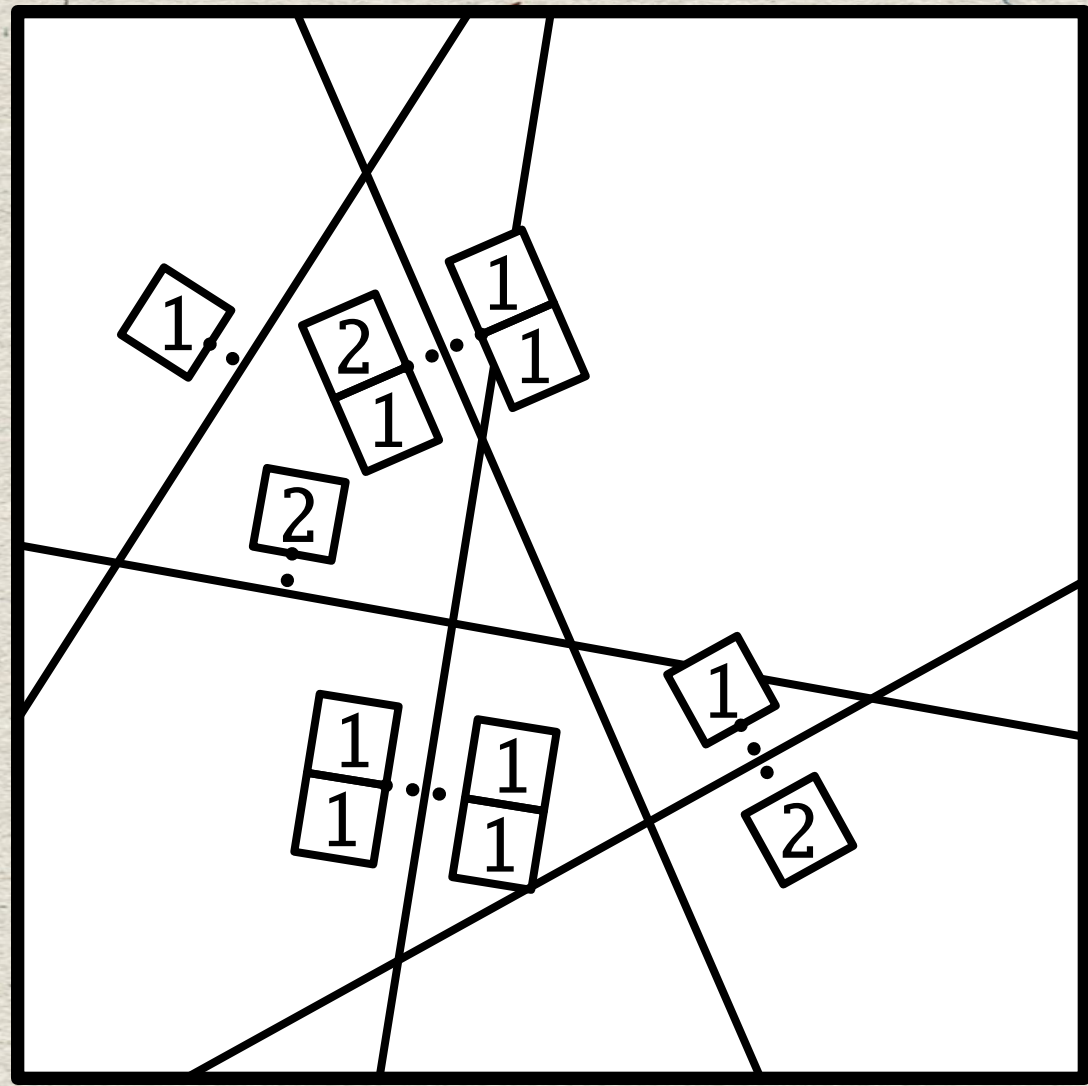


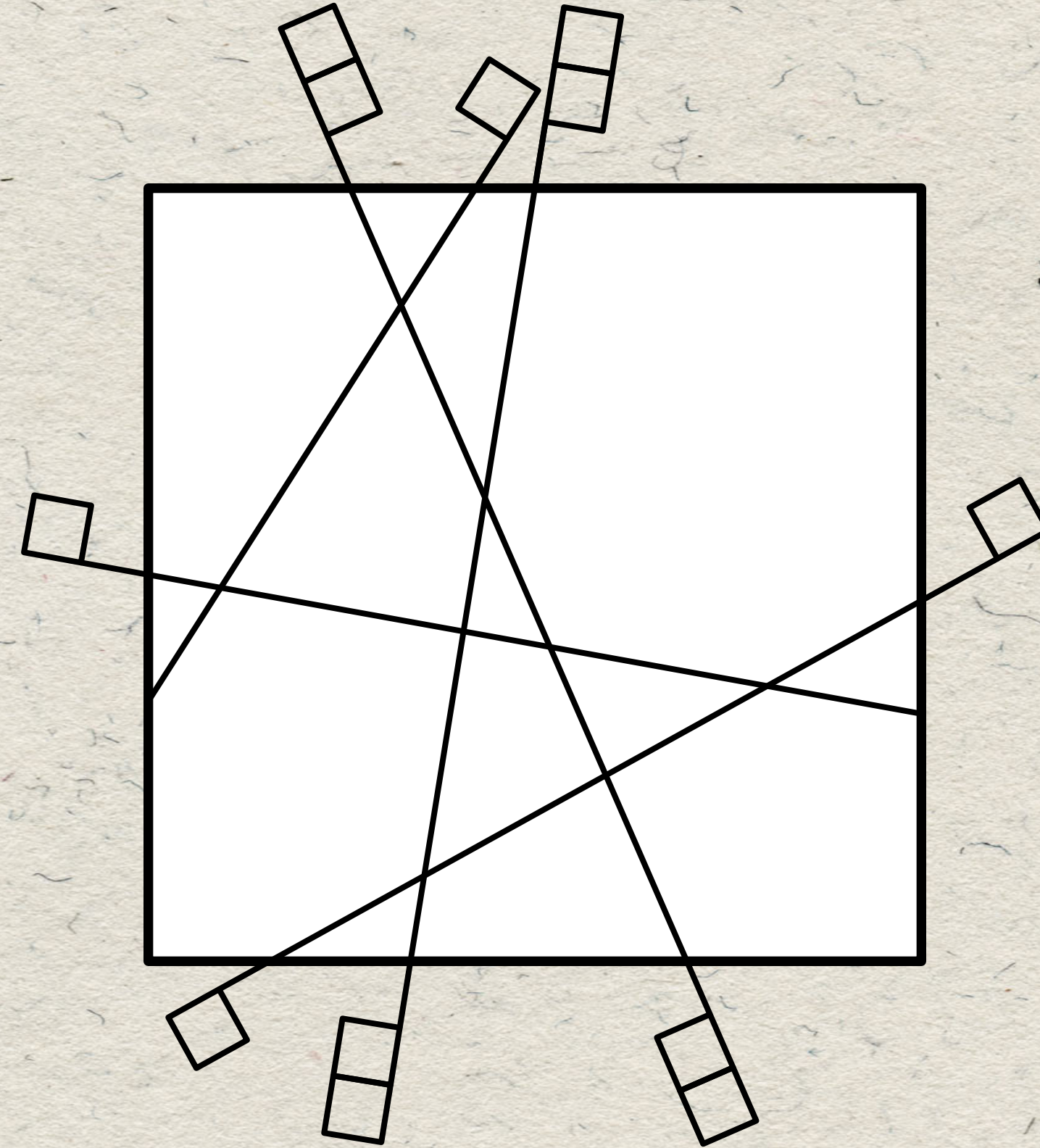
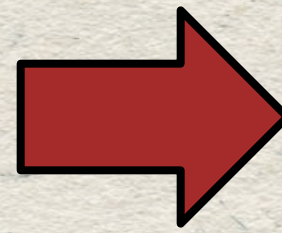
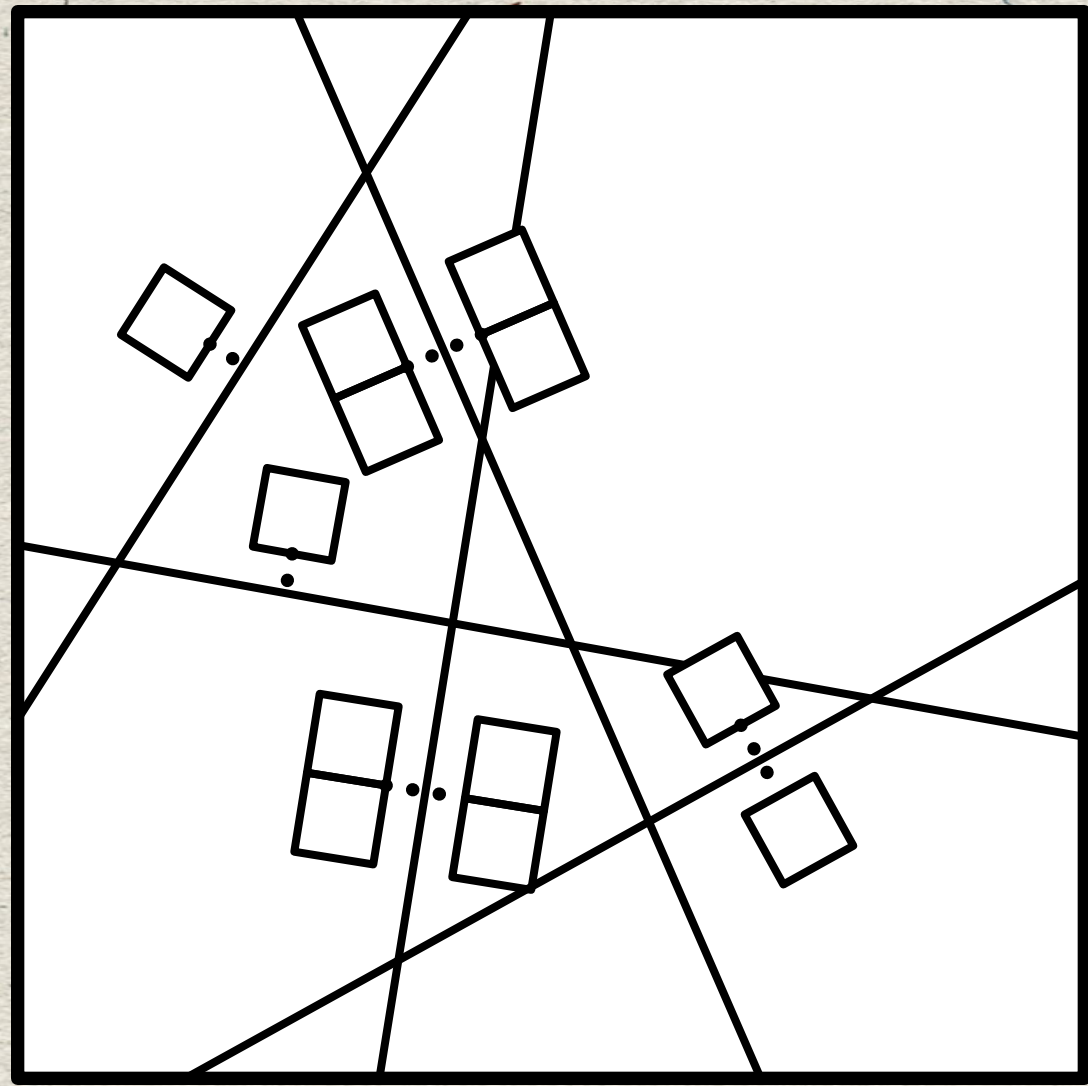




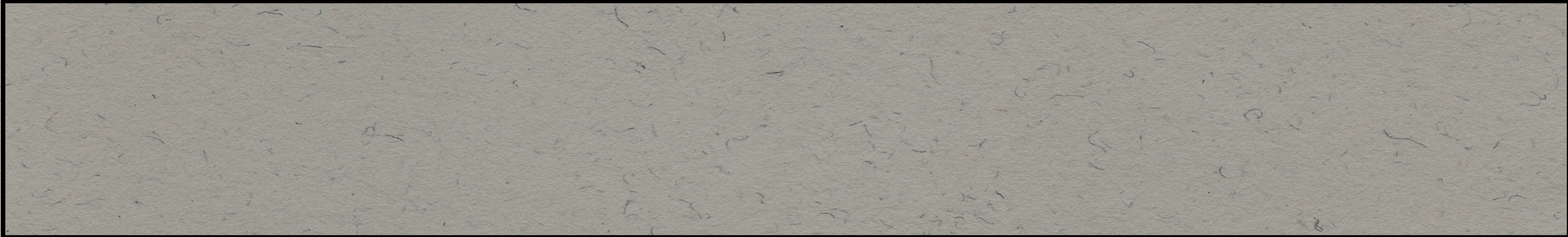






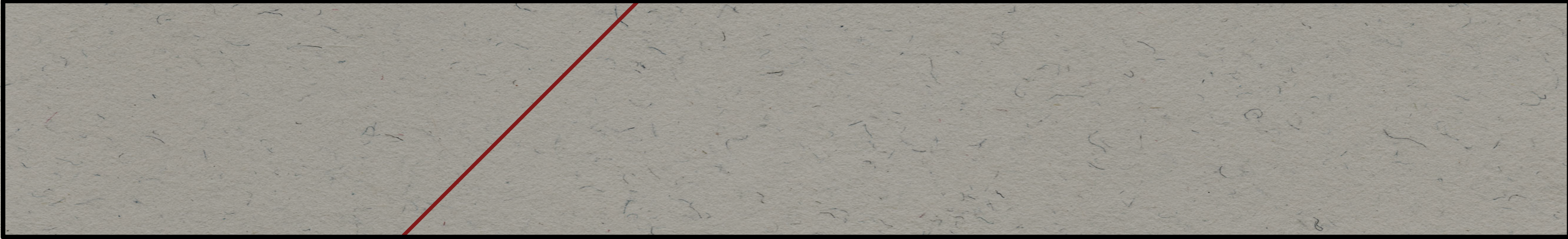


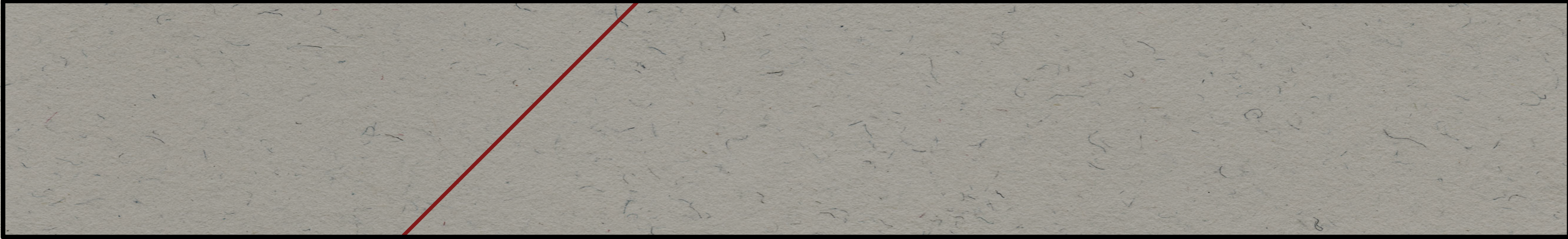
LABELING NONOGRAMS IS NP-HARD

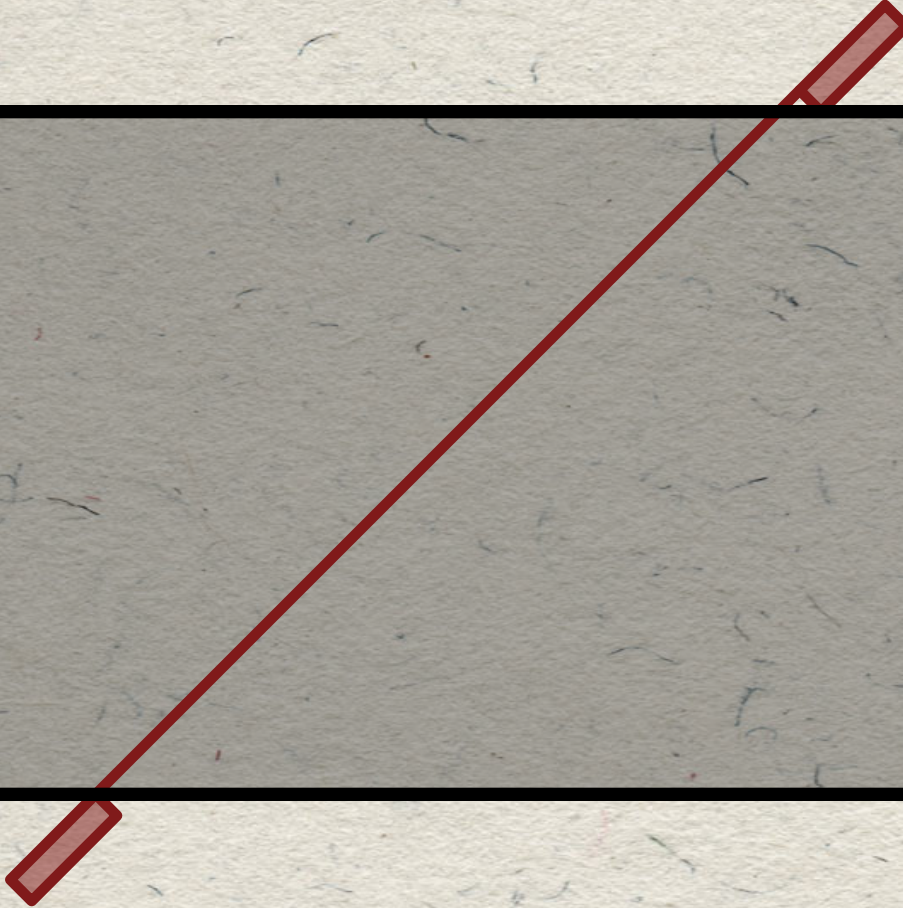
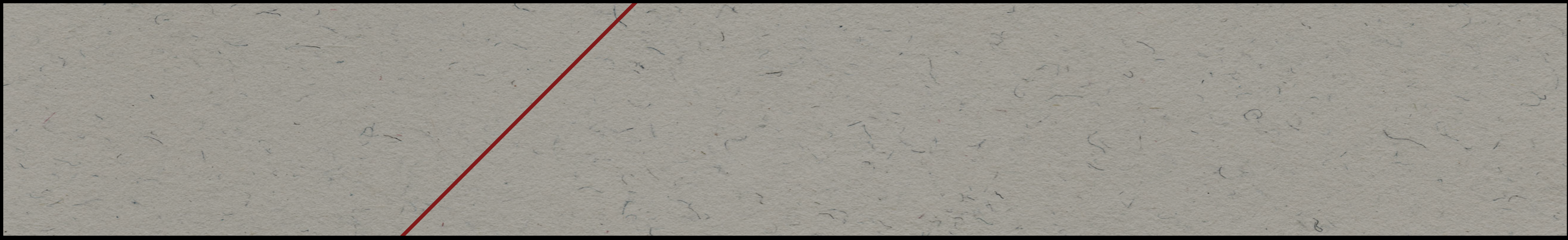


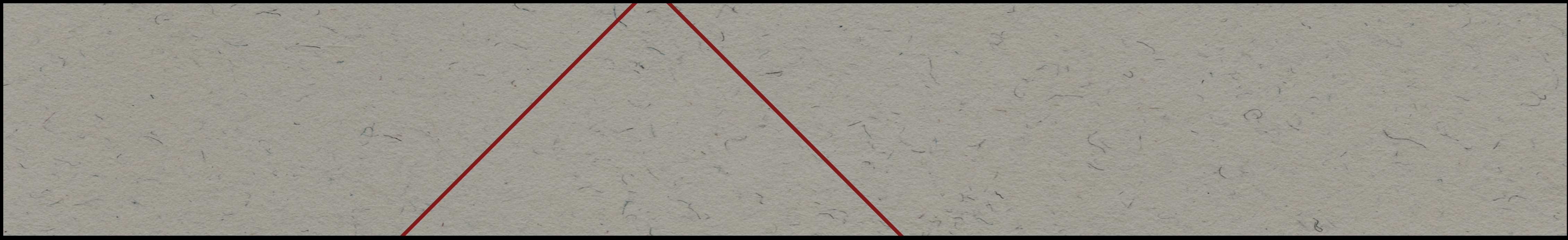


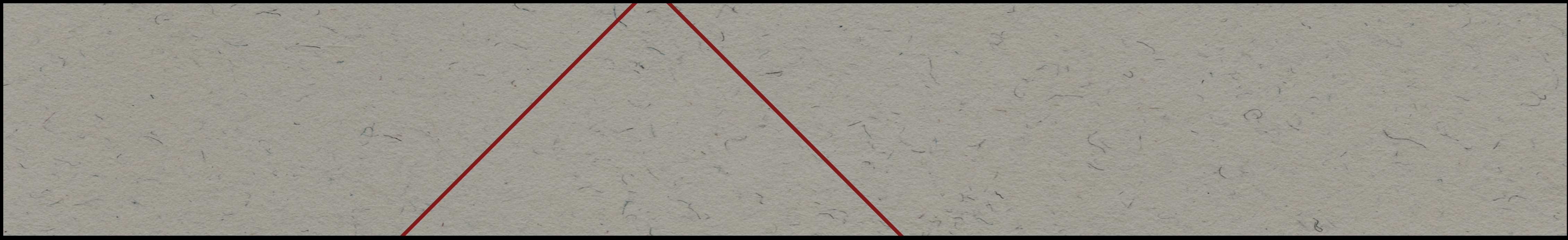


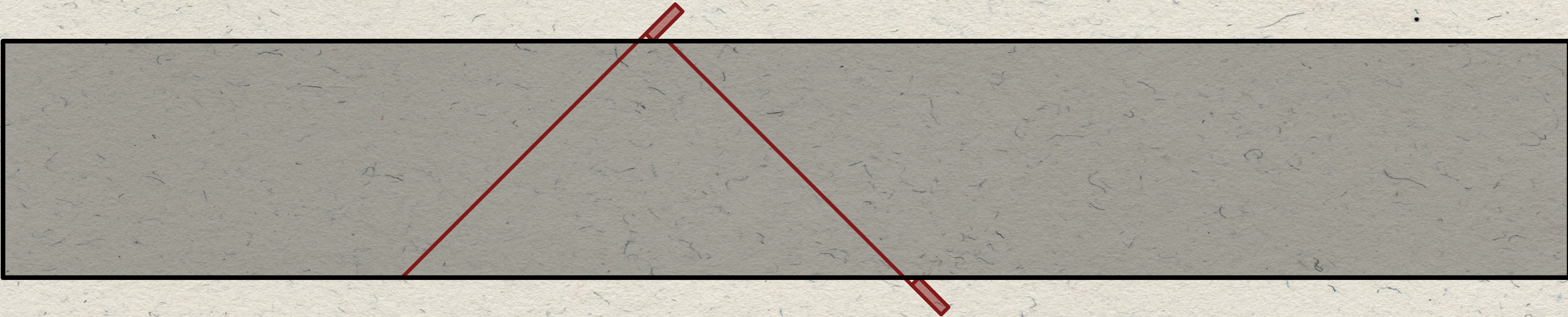


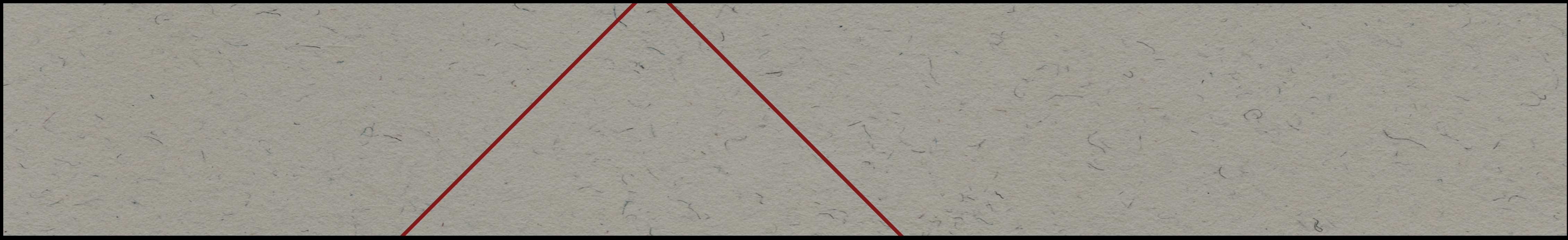


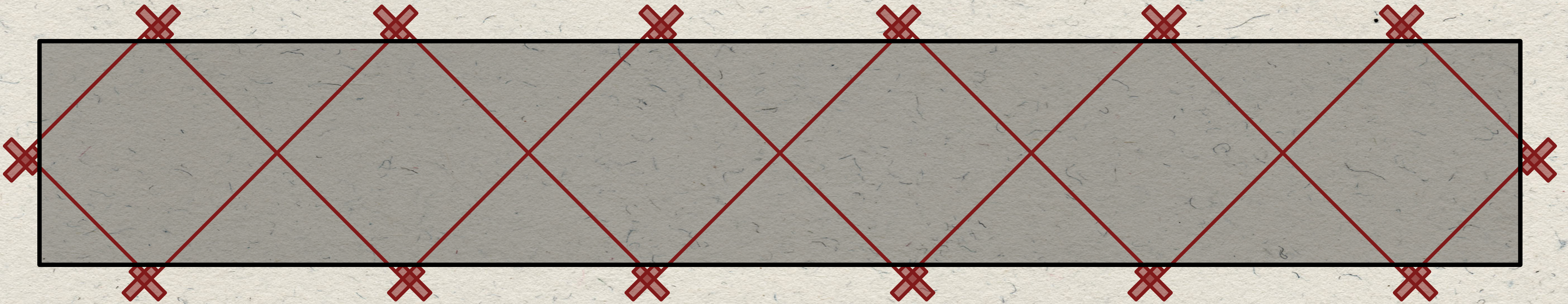


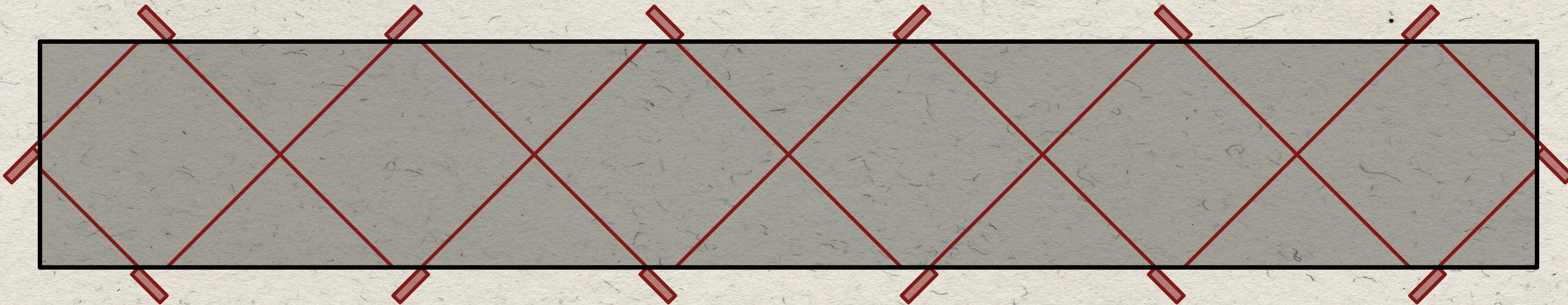


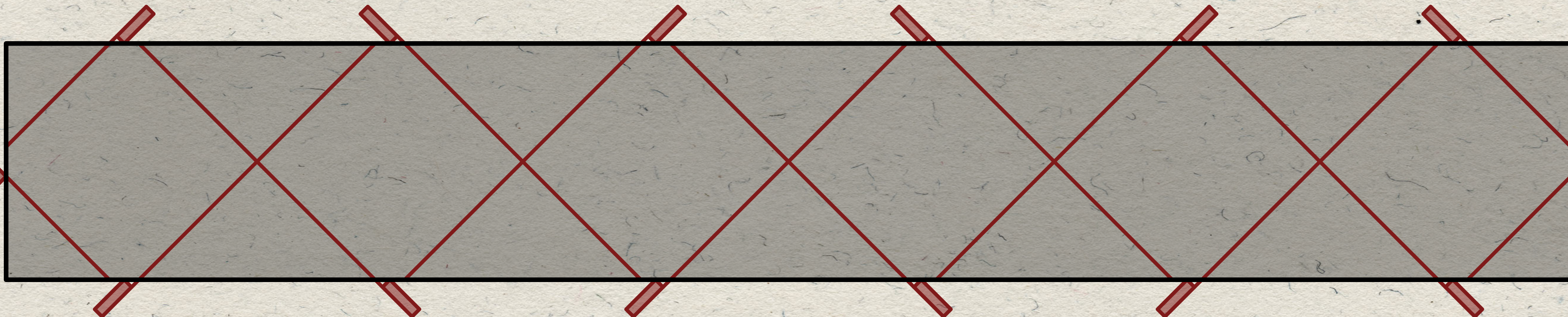


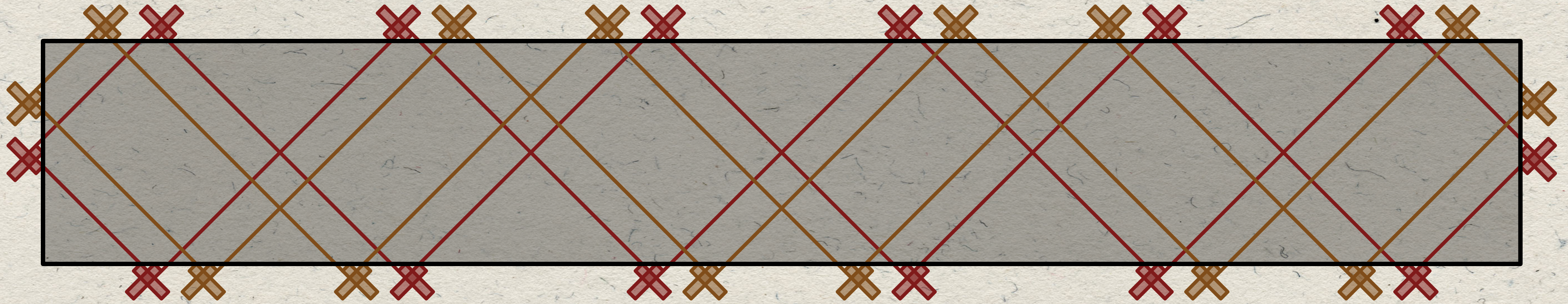


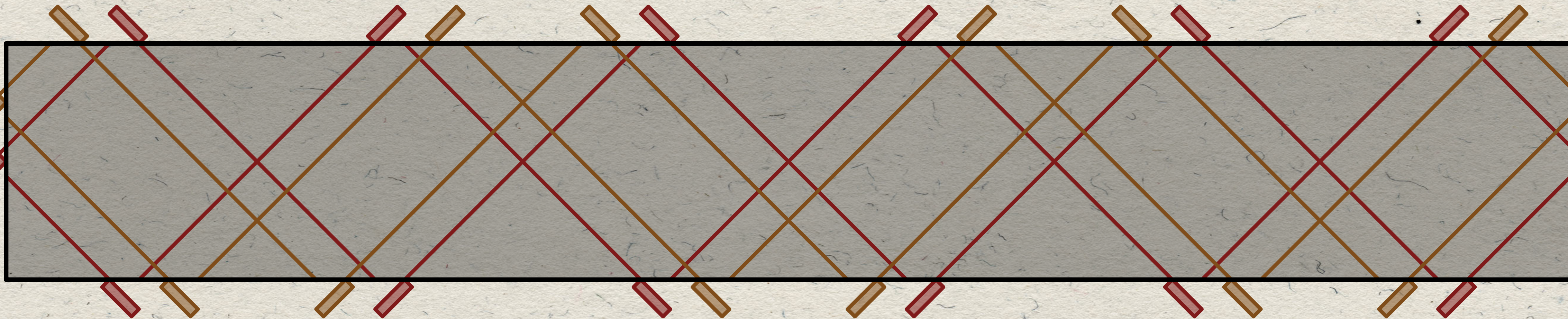


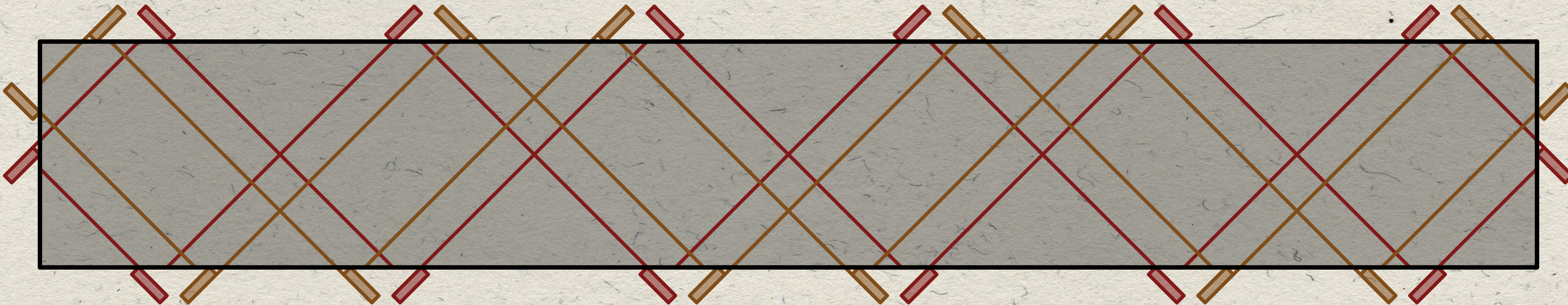


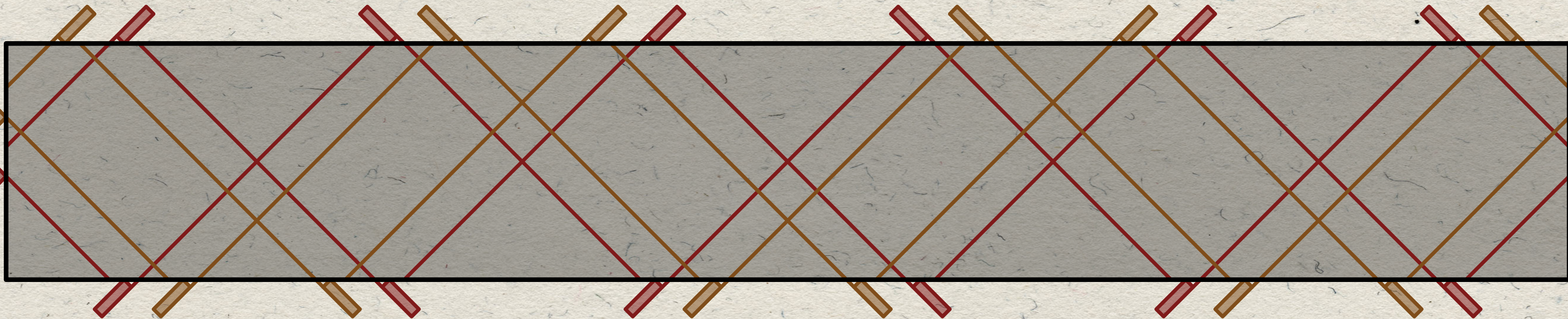


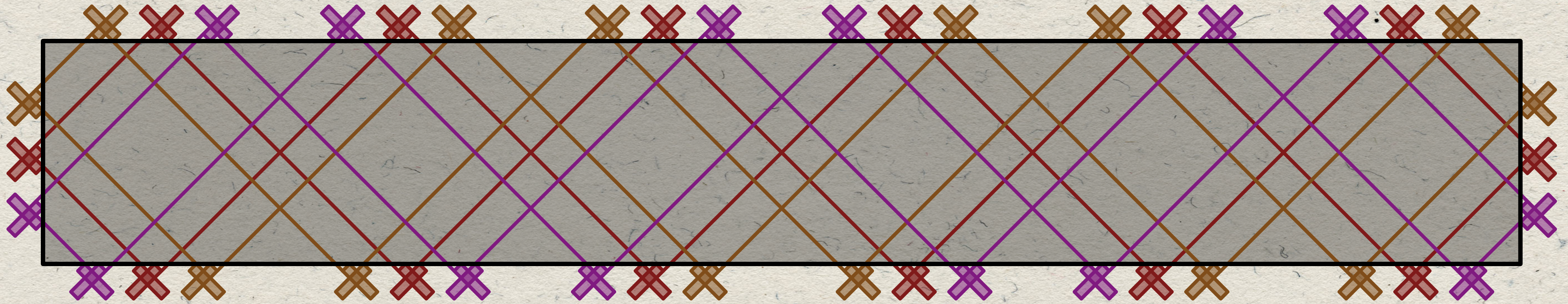


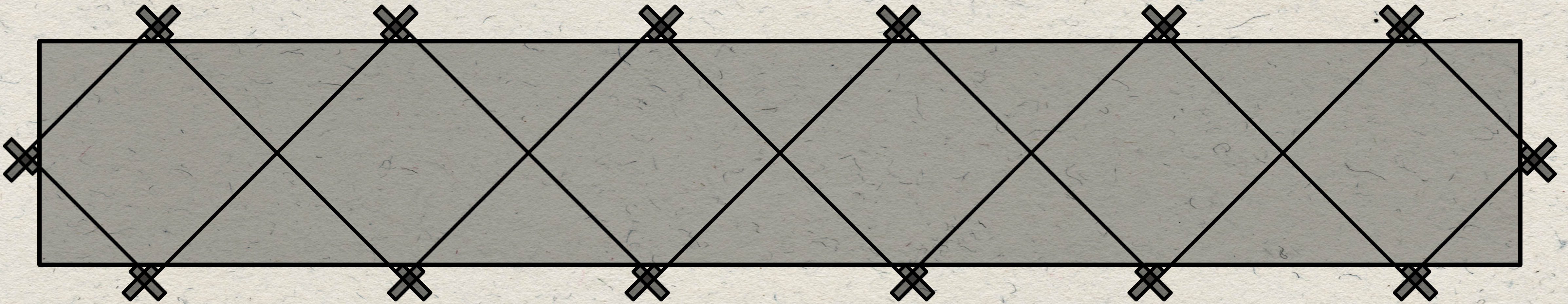


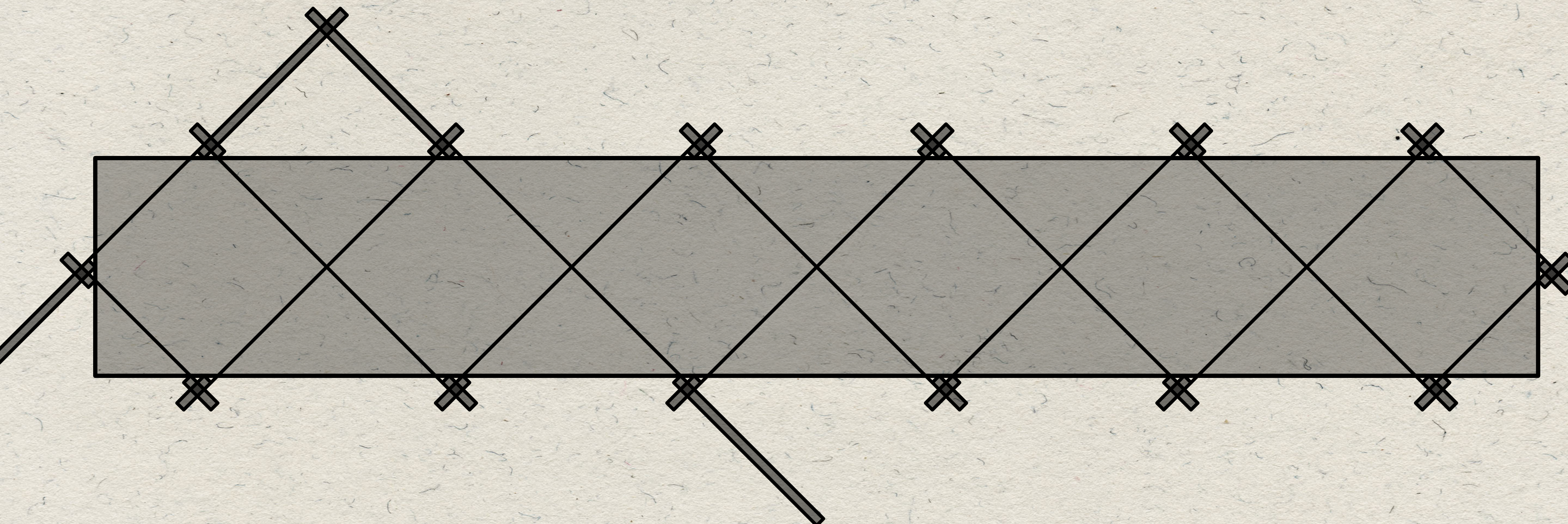


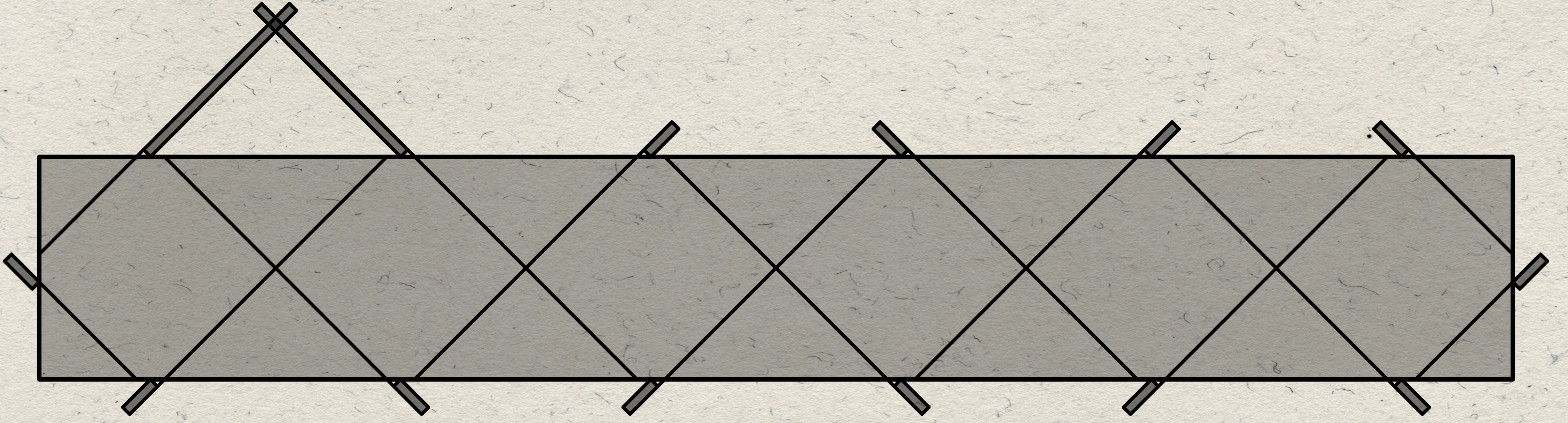


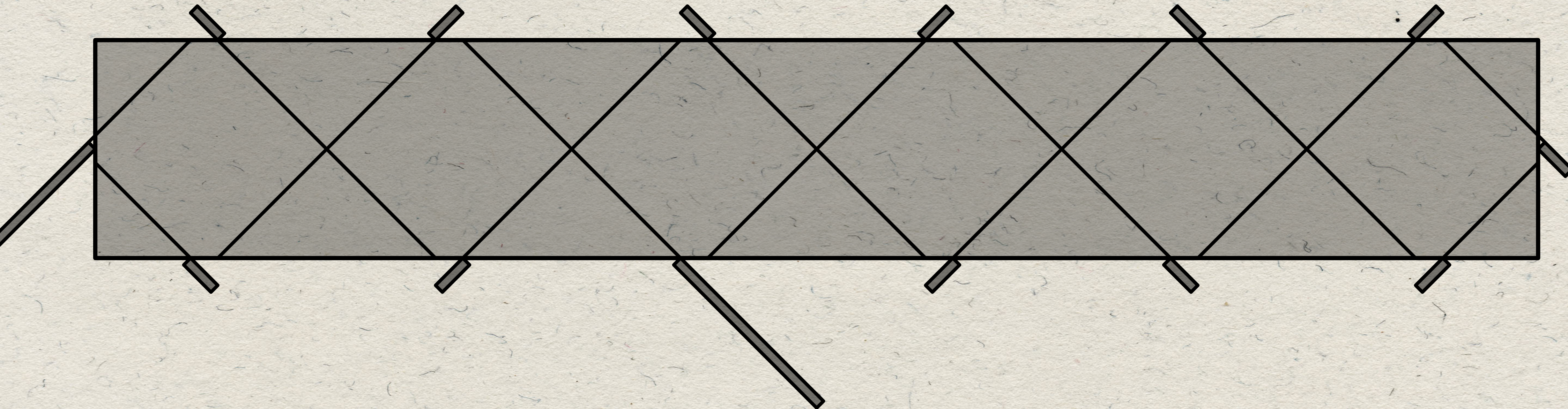


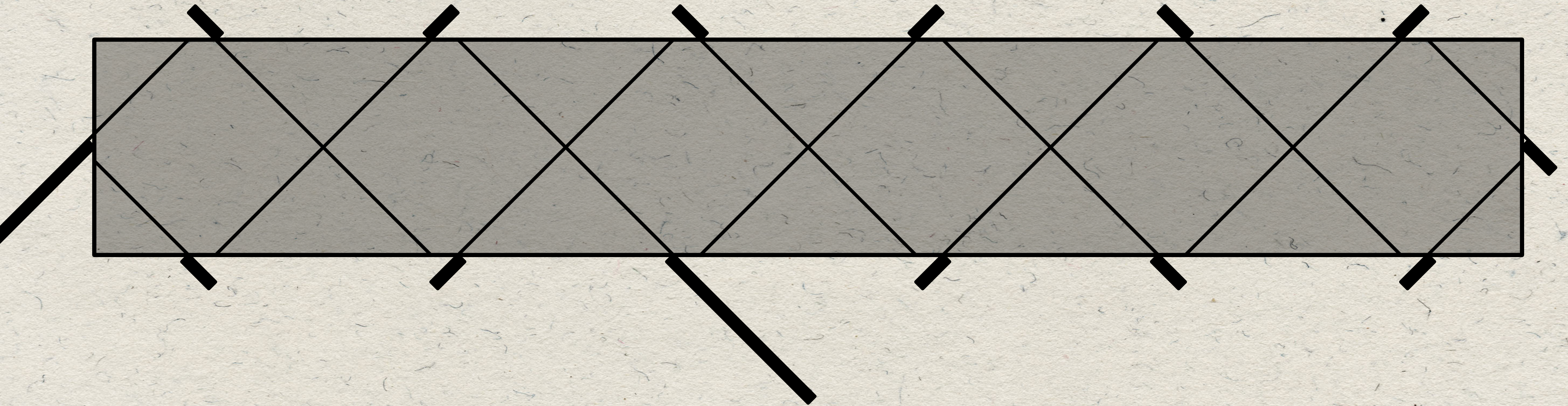


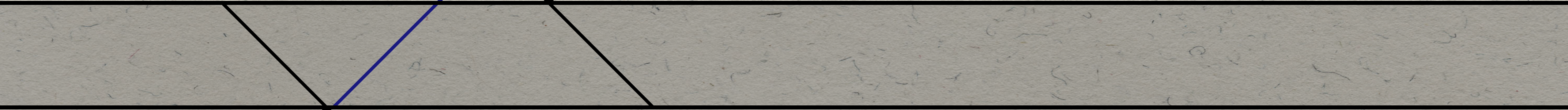


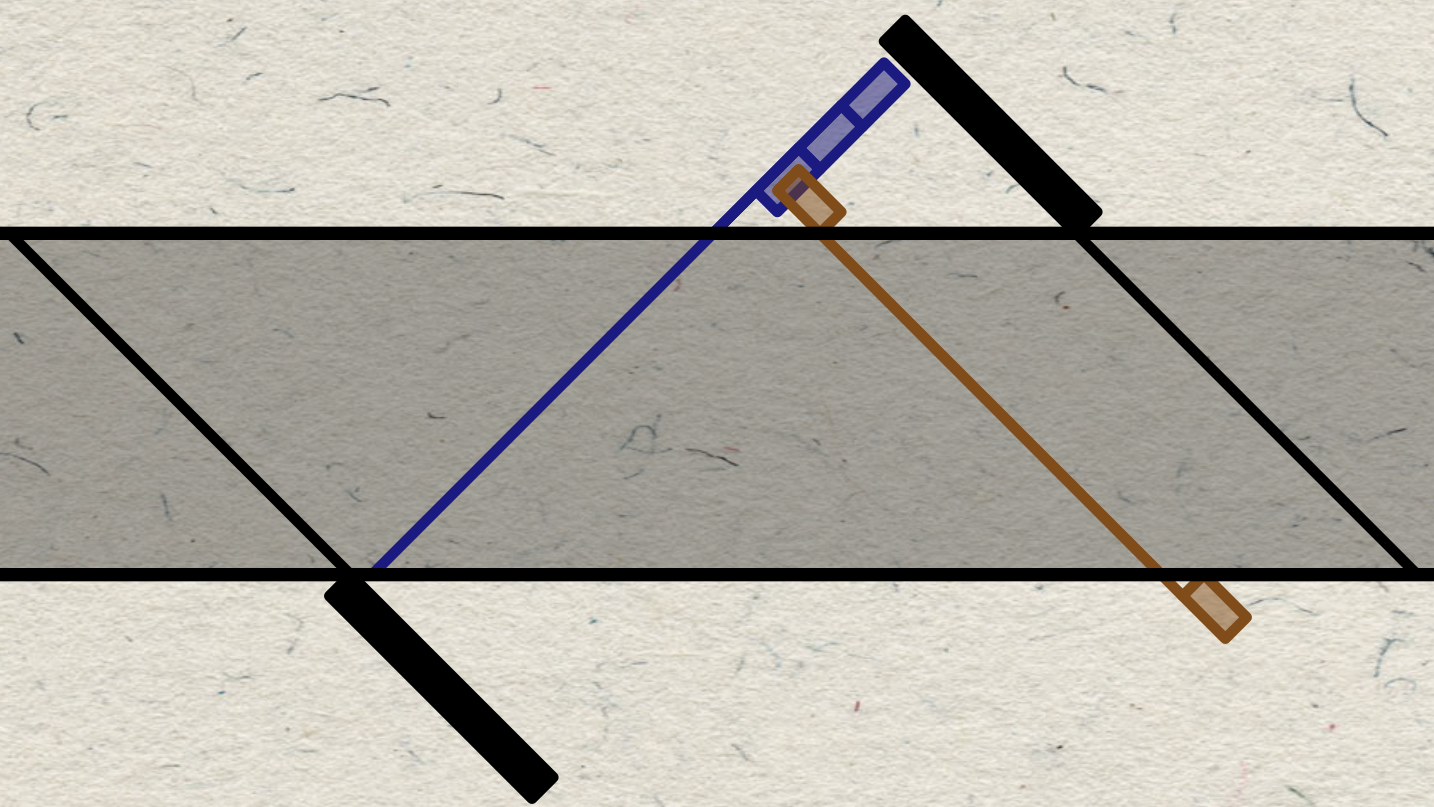


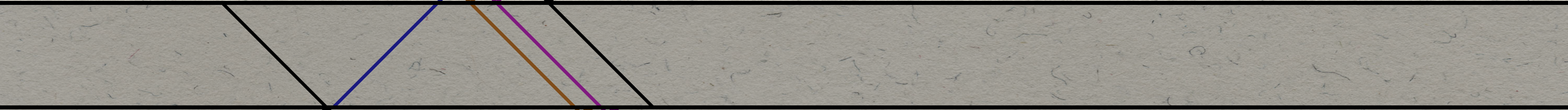


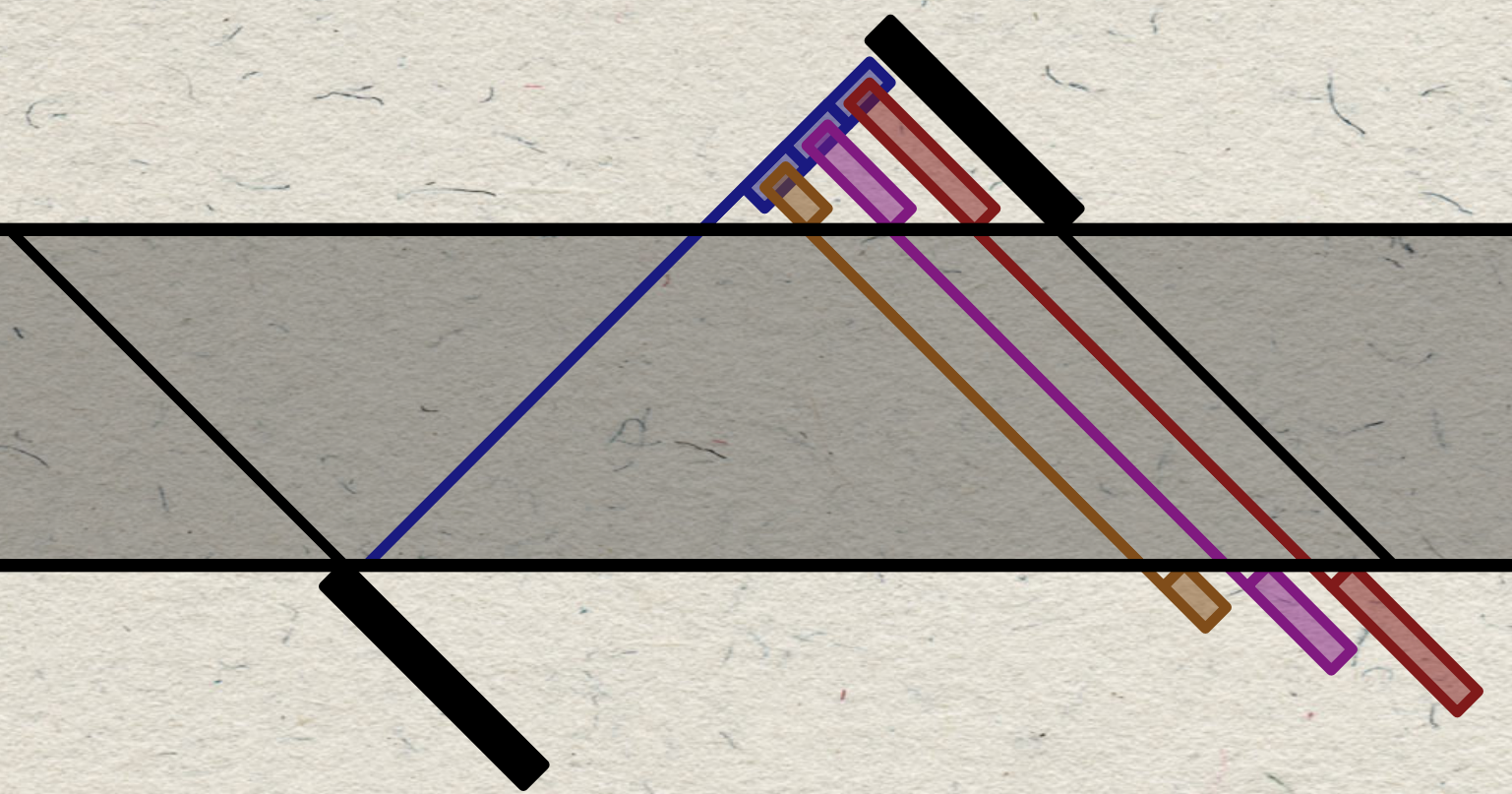
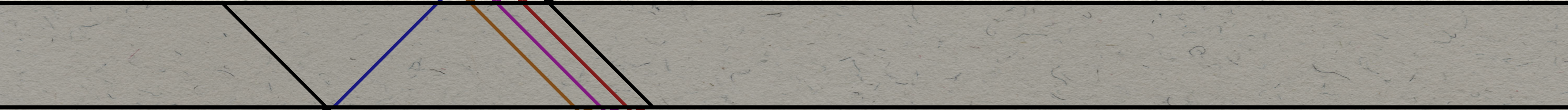




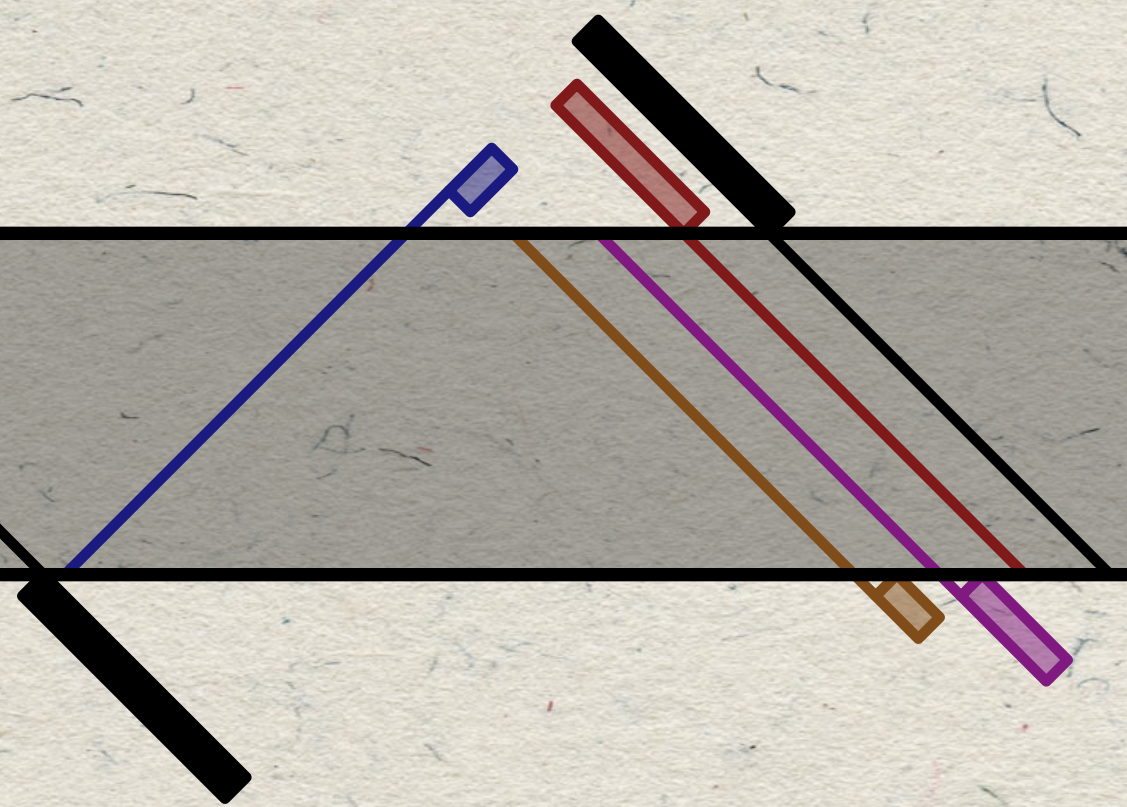


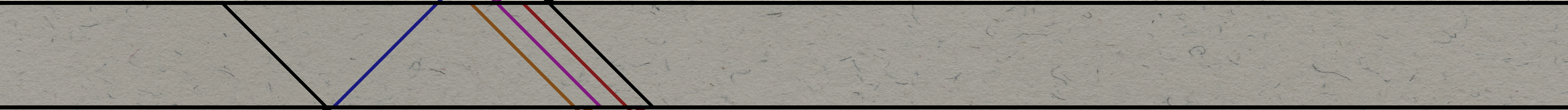


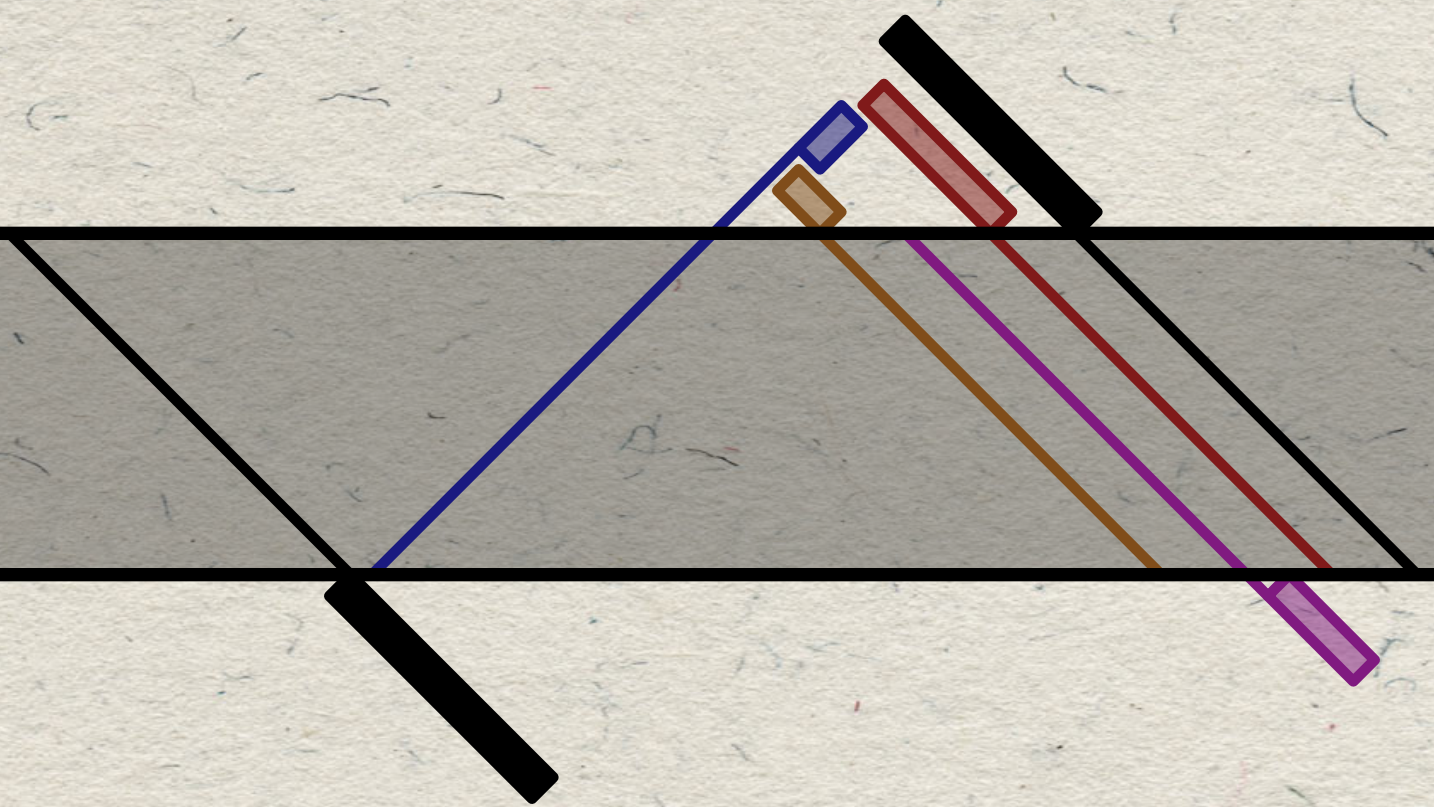
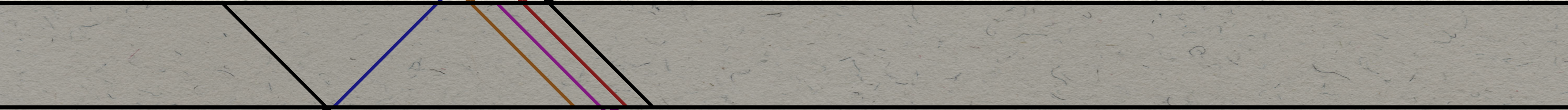


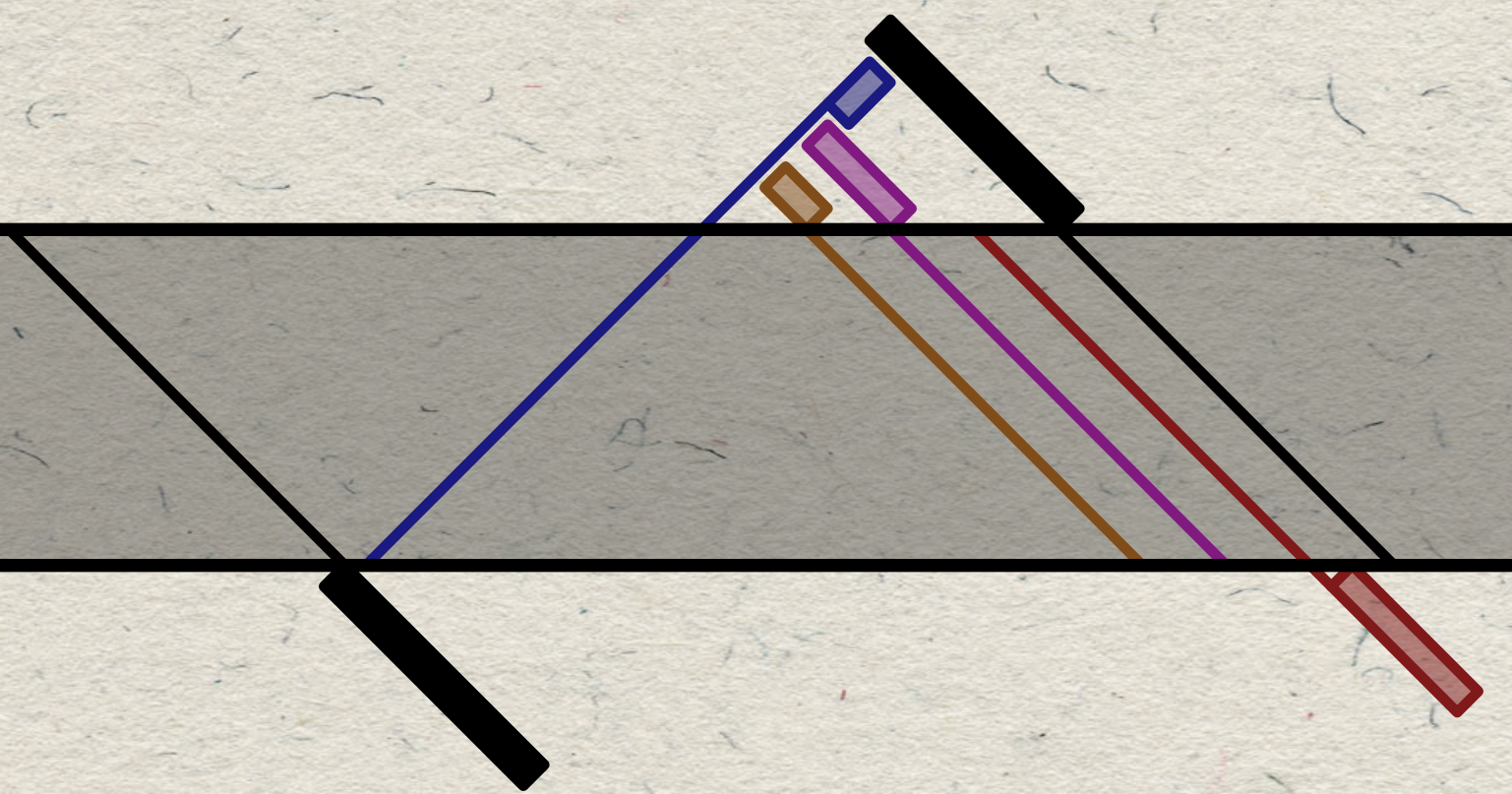
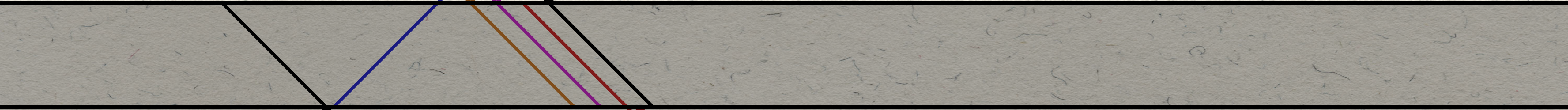




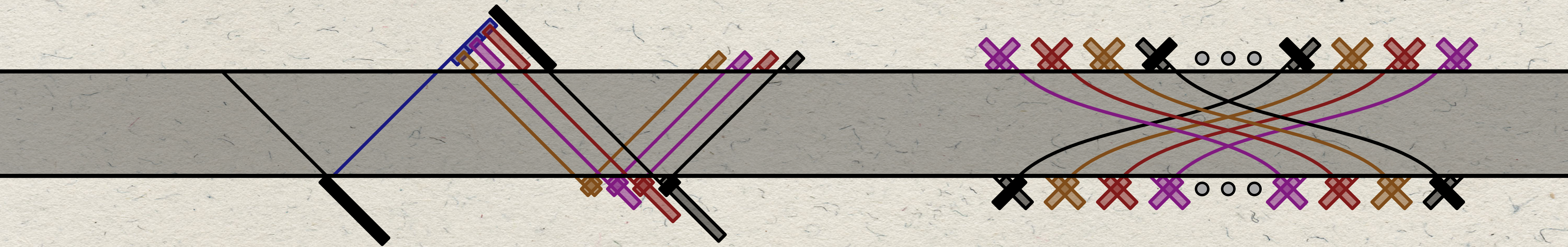


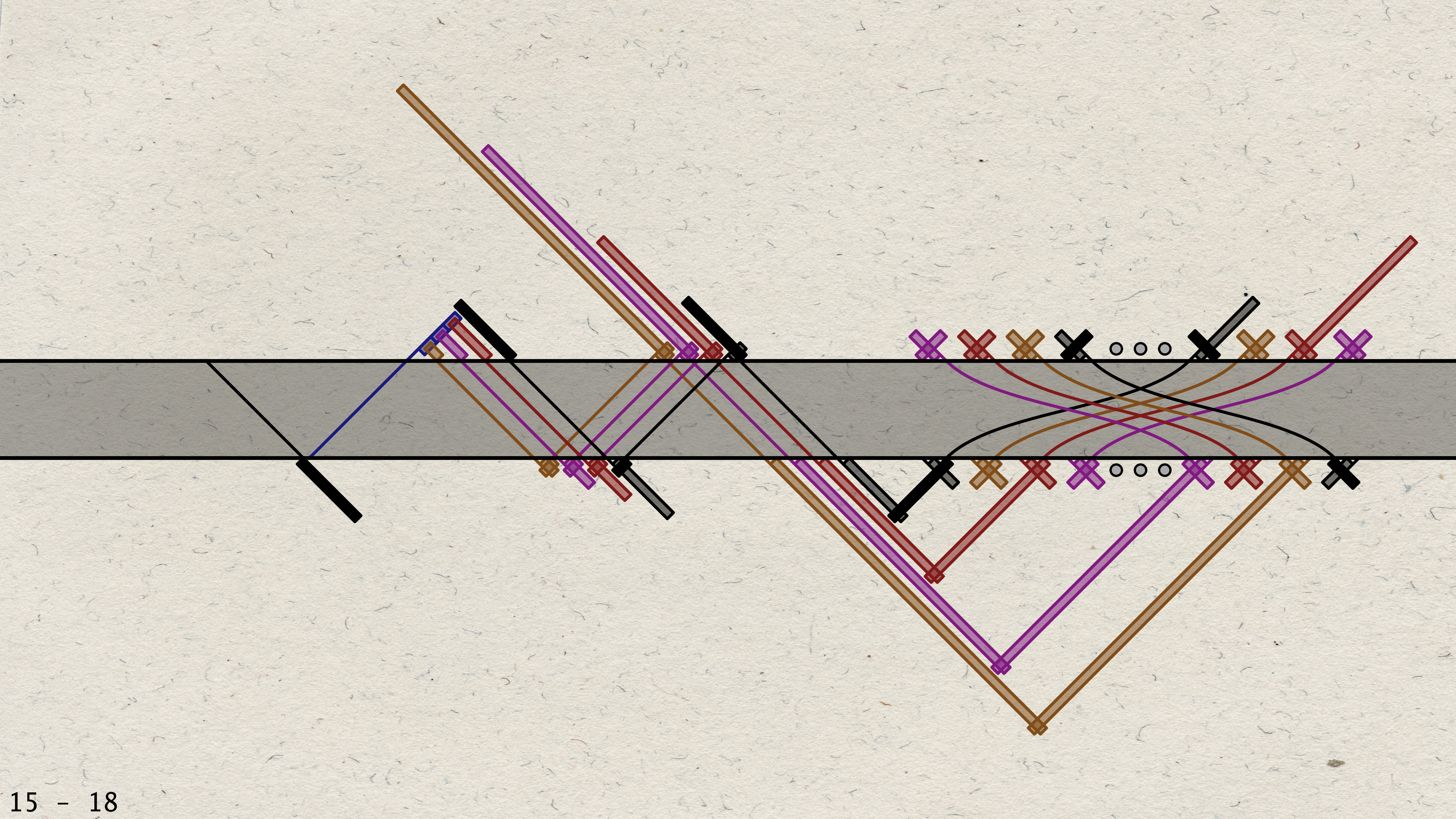


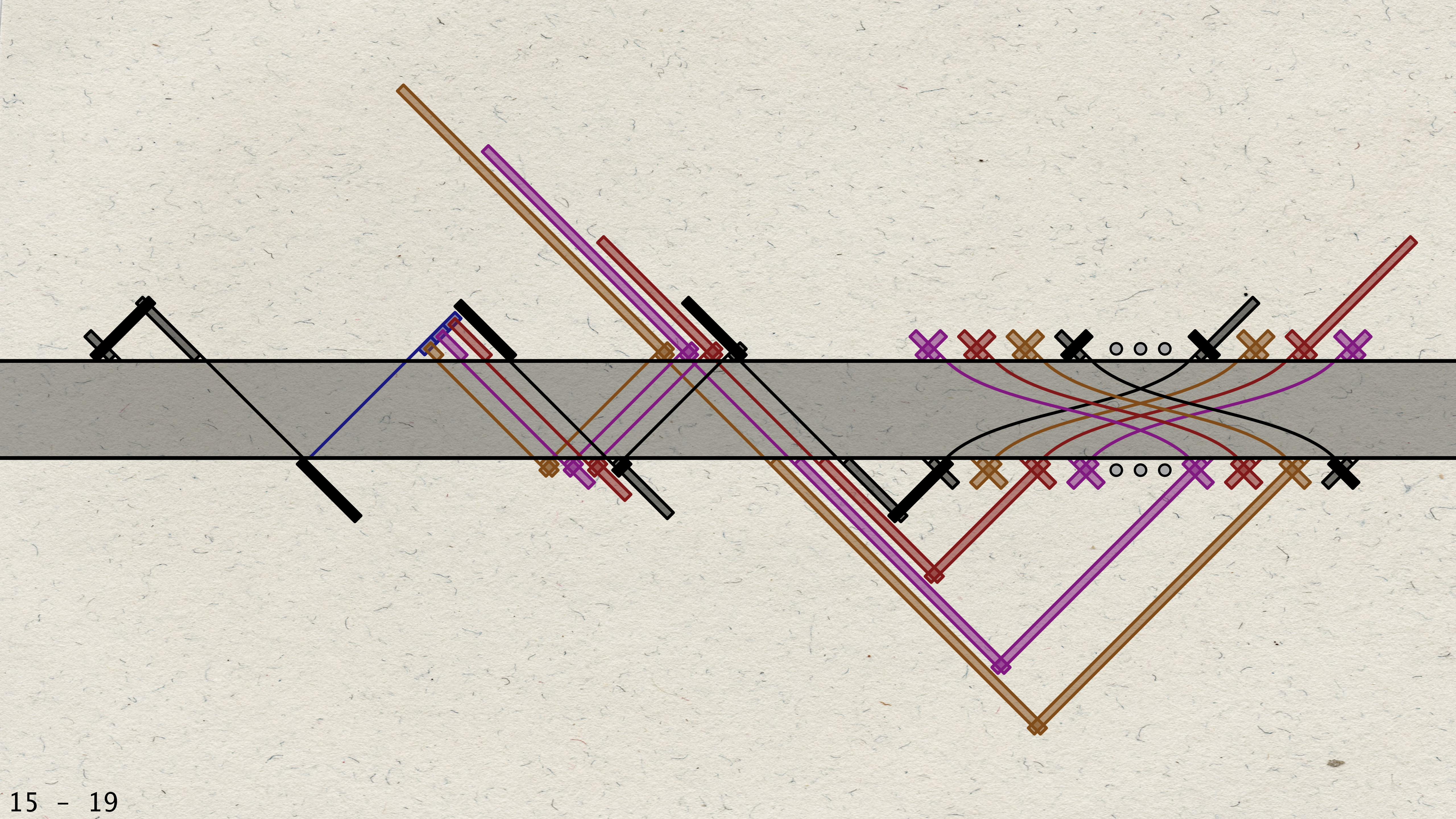


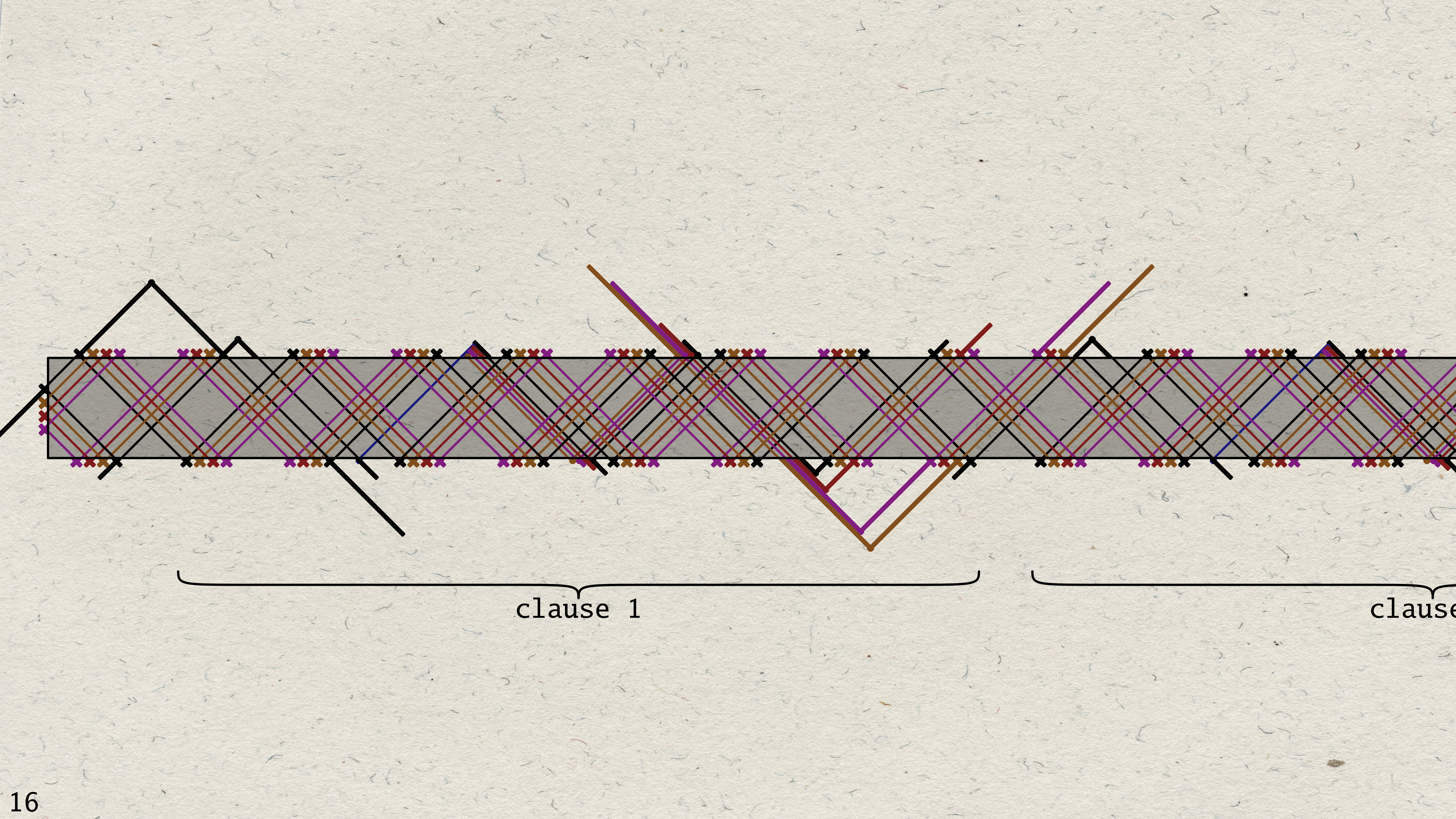






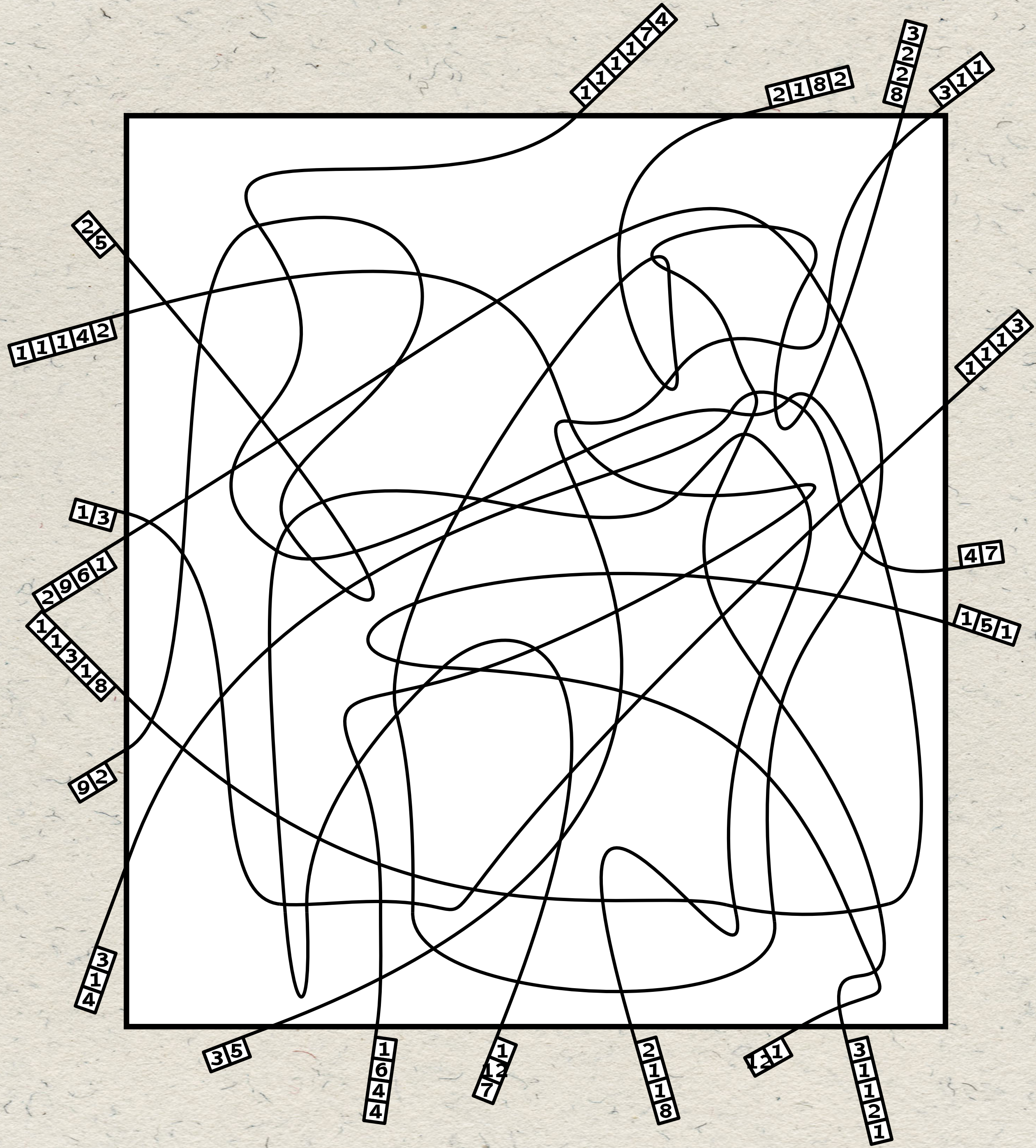


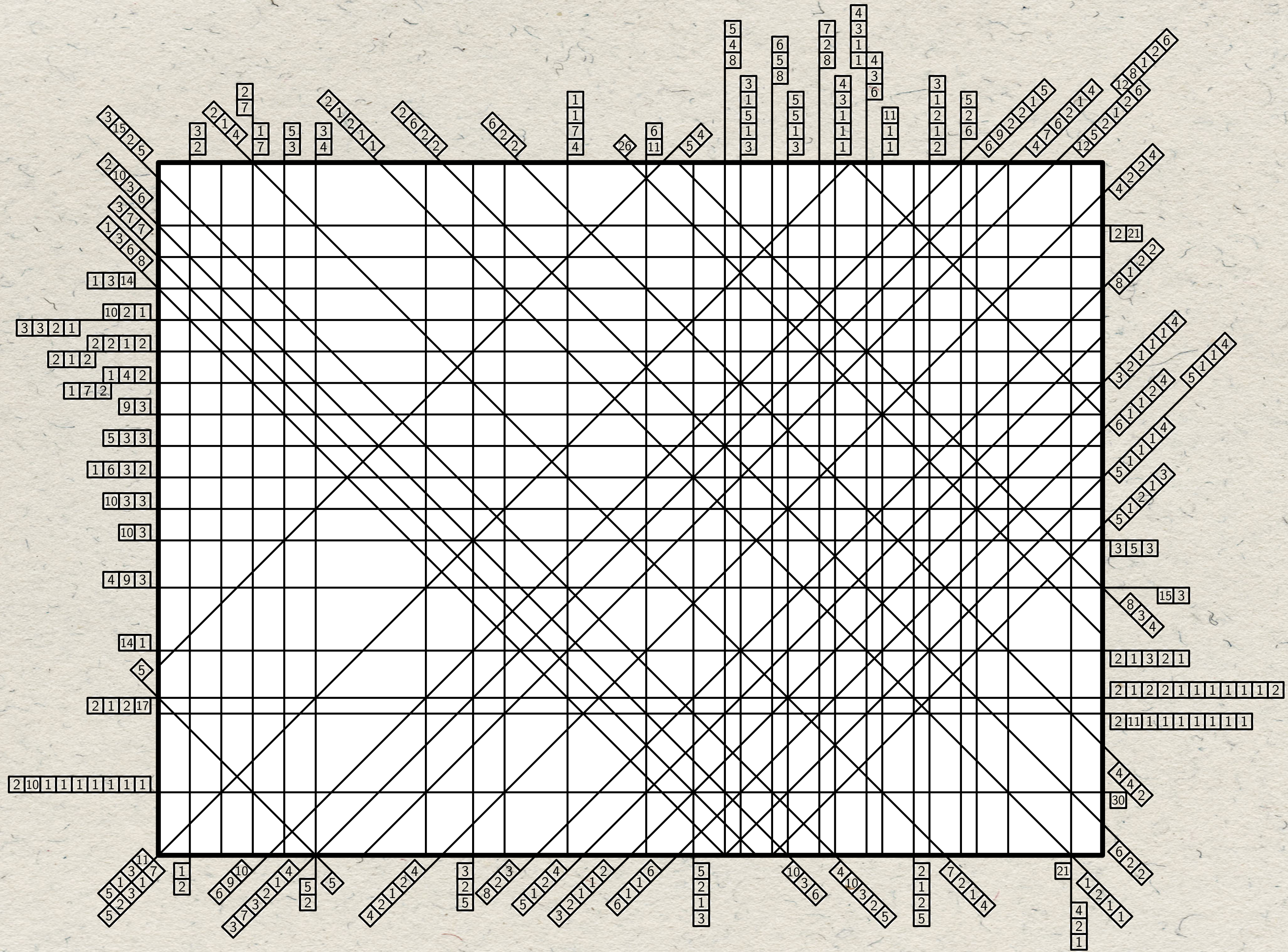


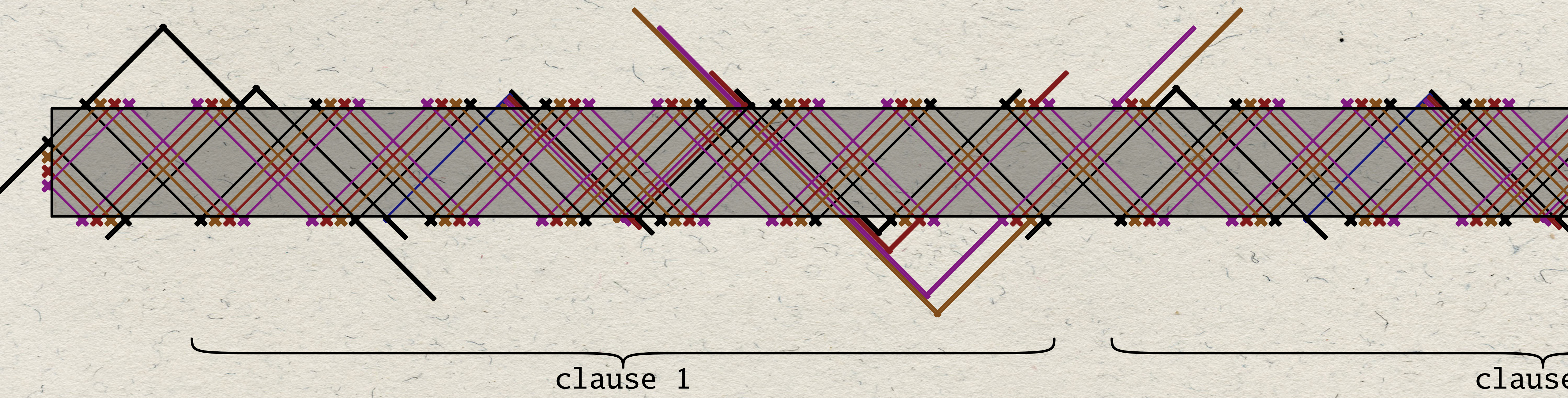


clause 1

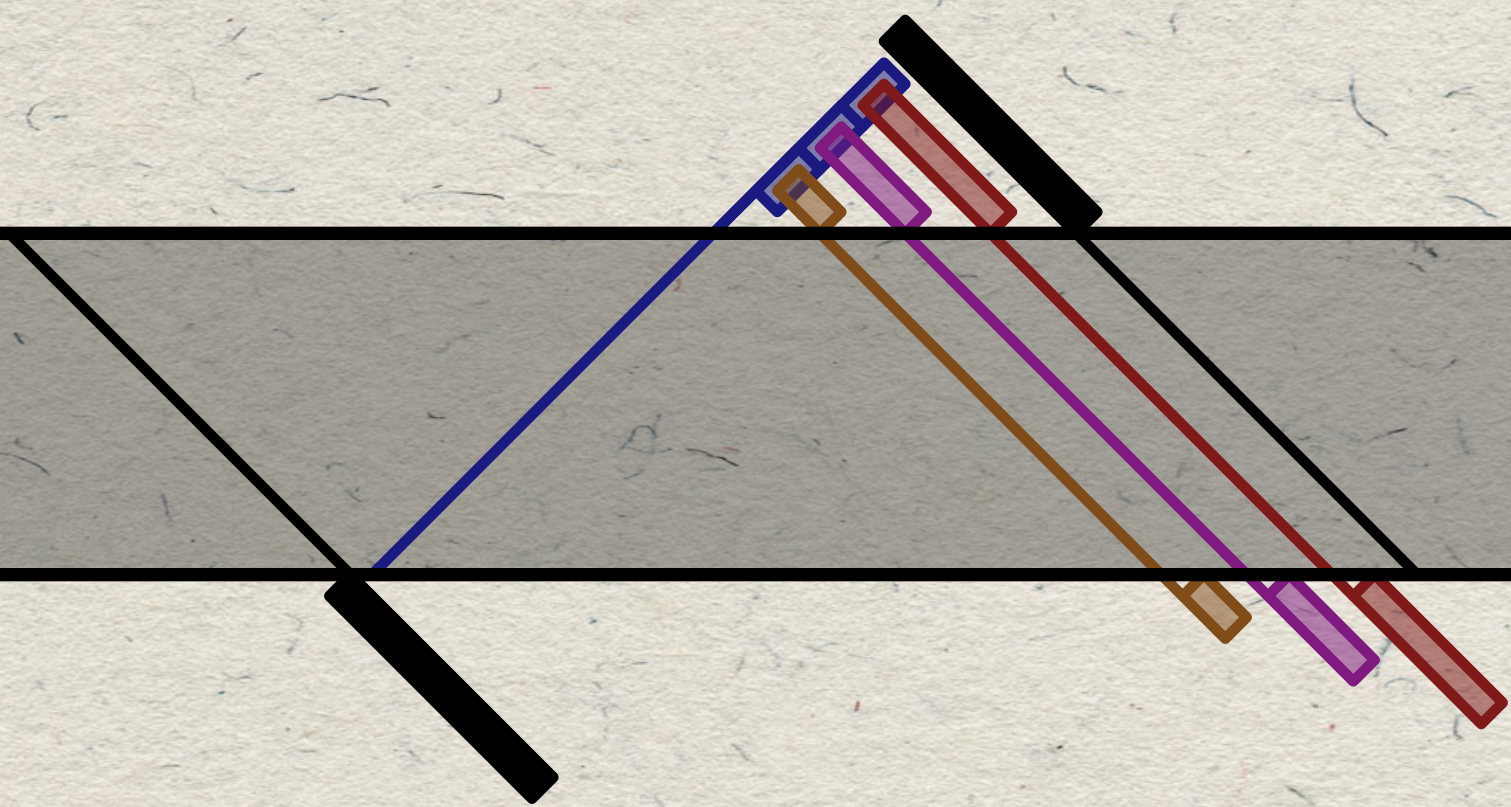
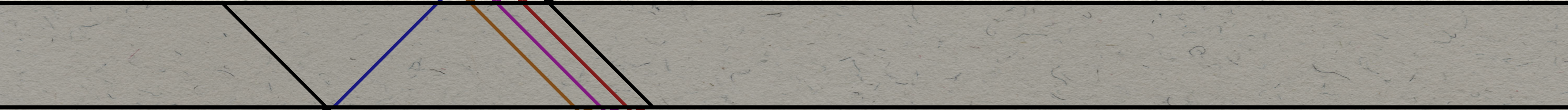
clause

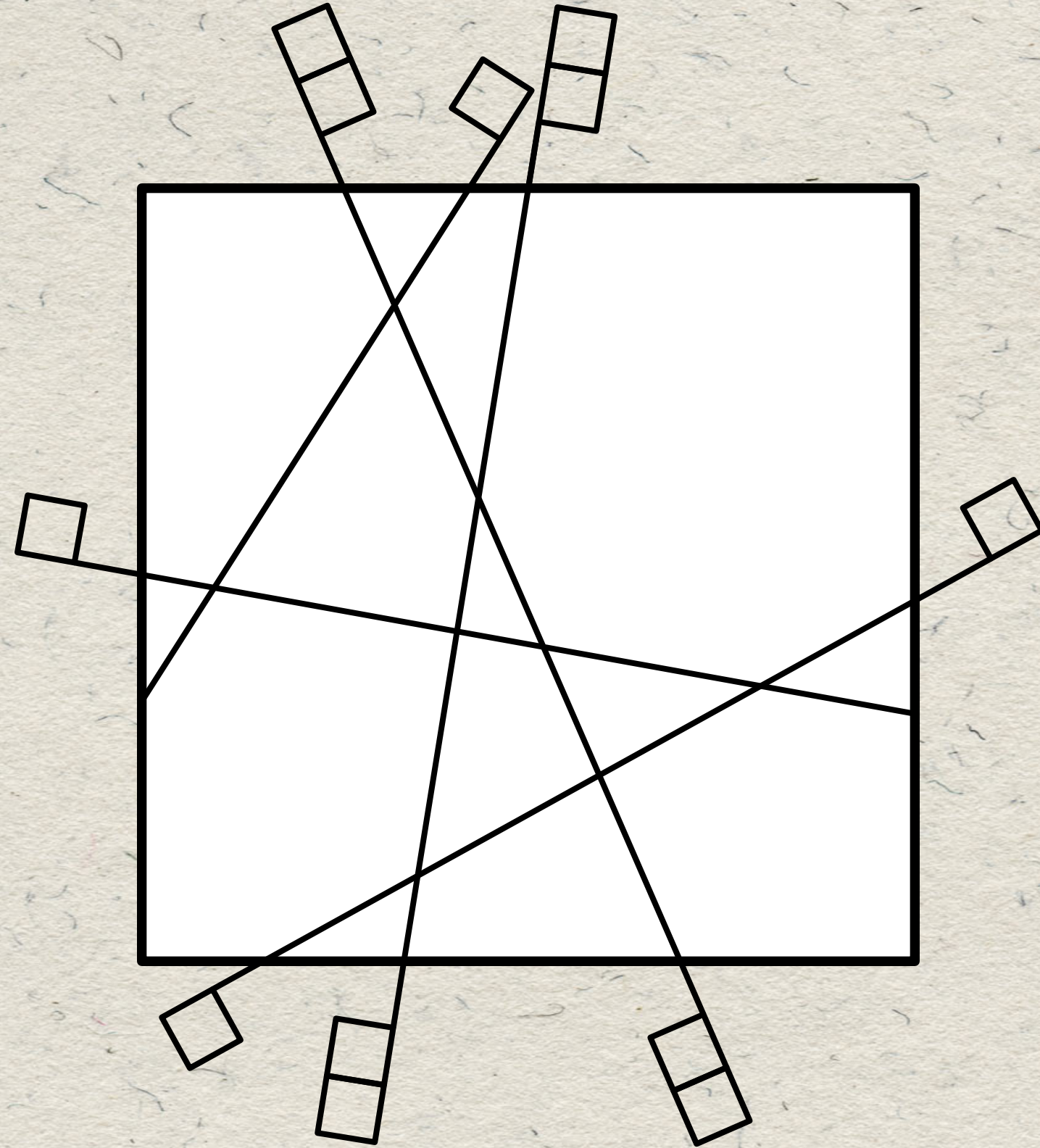


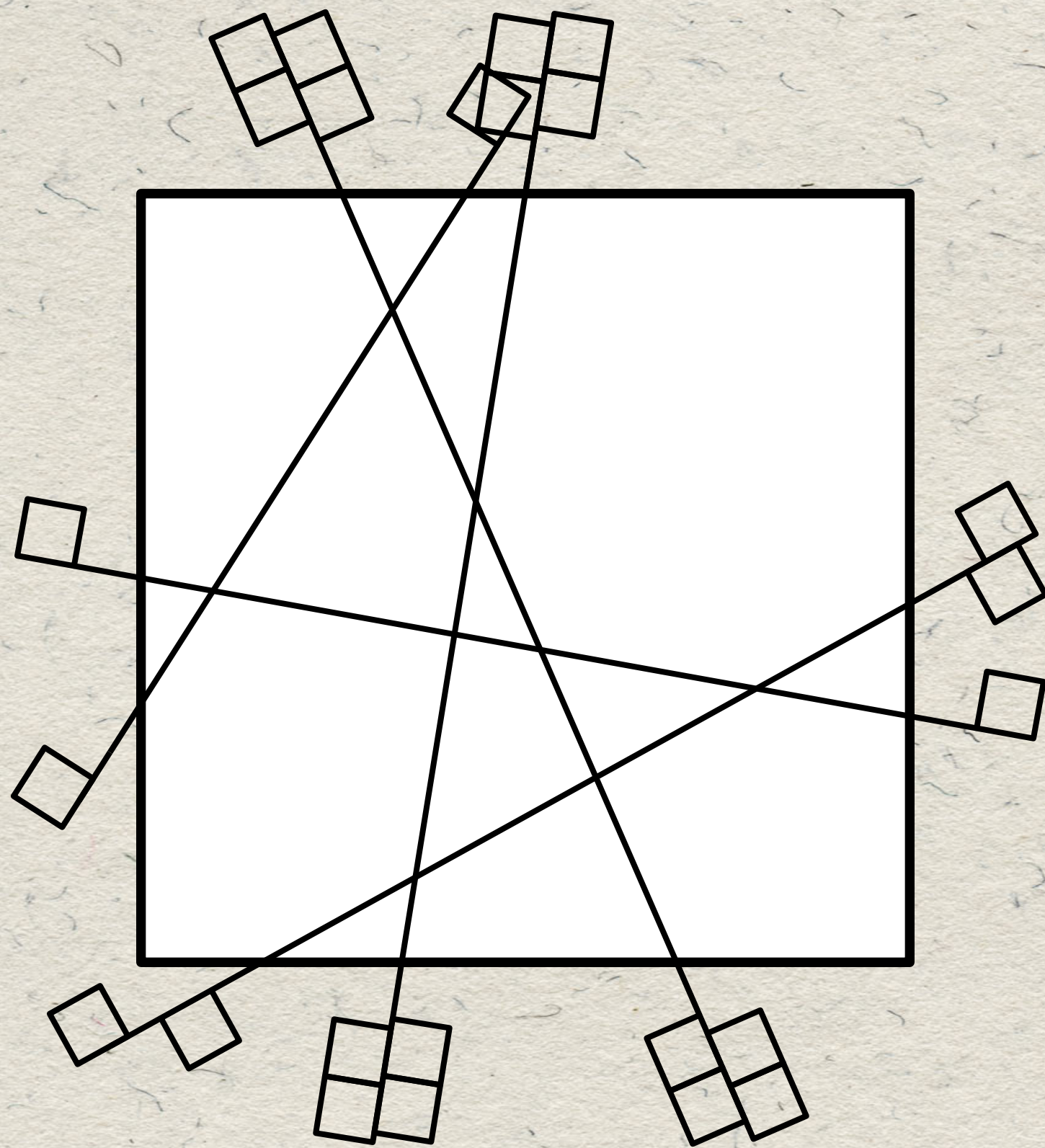


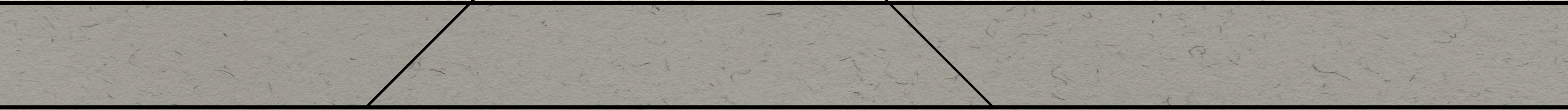


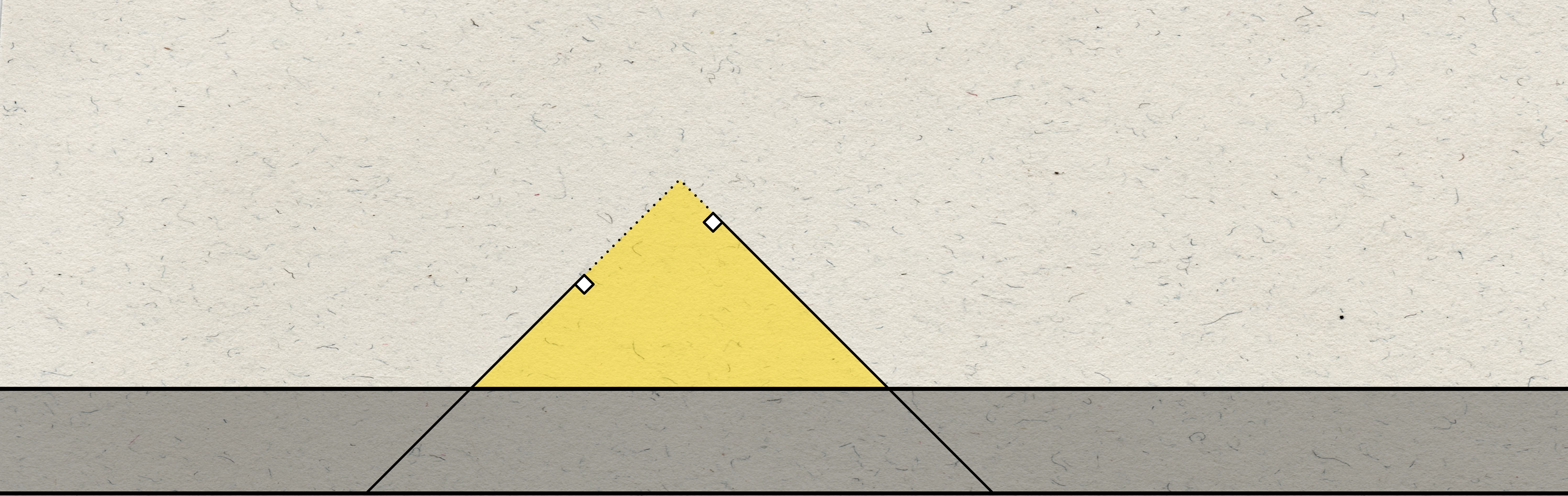
ALGORITHMS

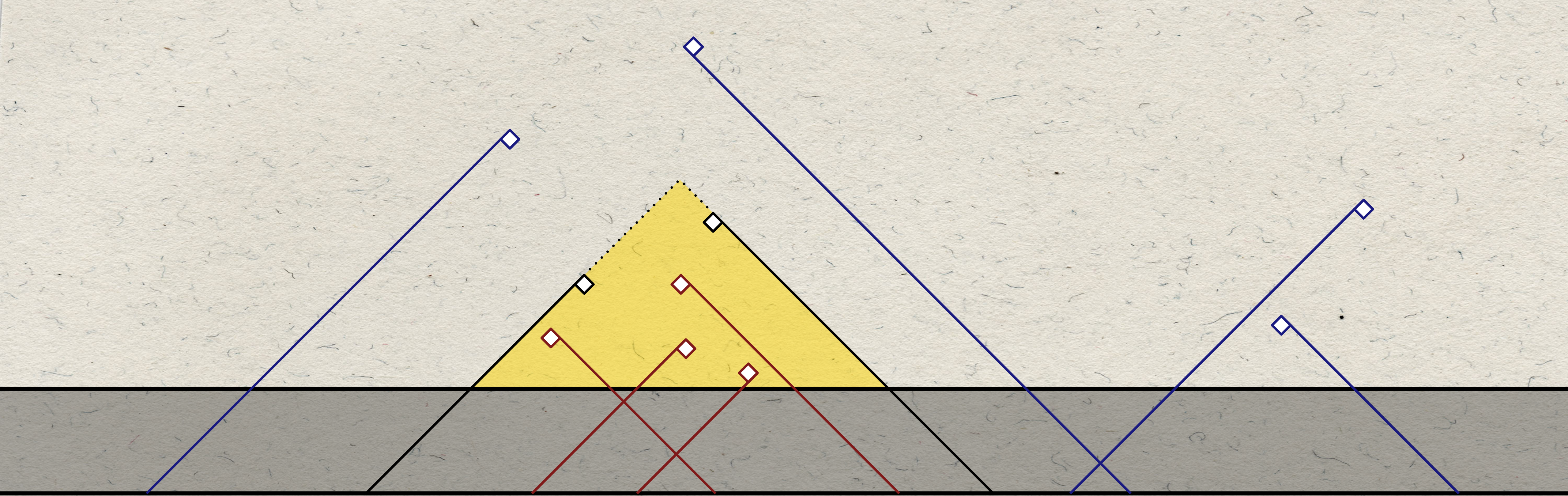


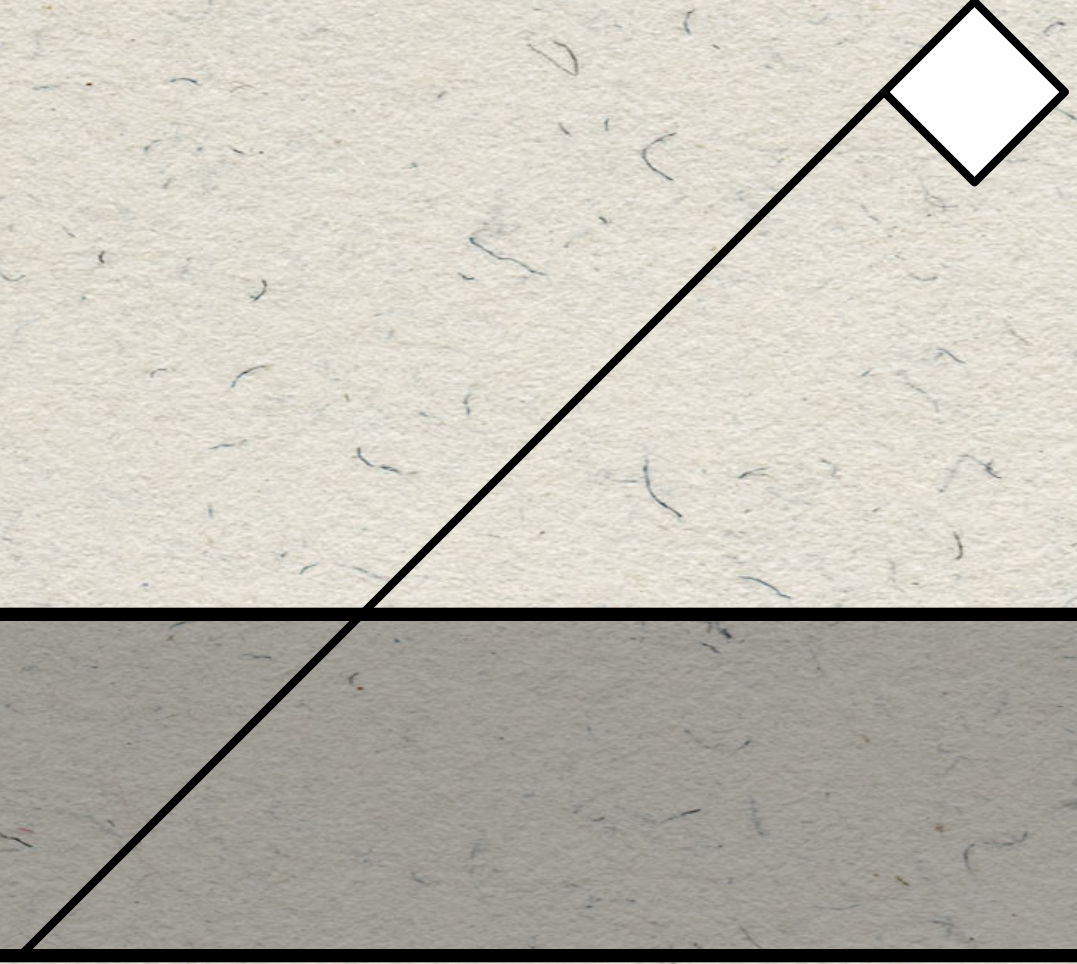


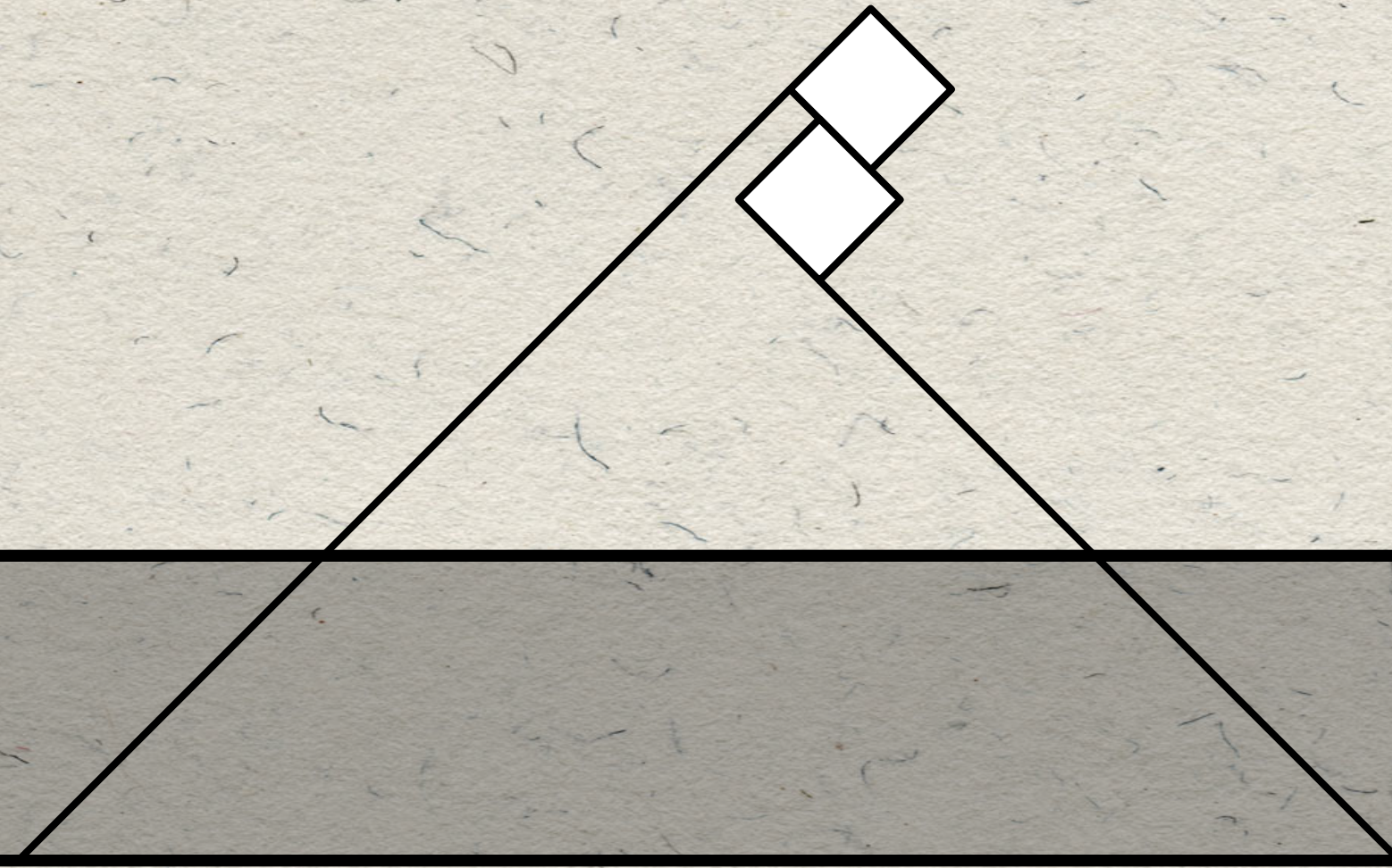
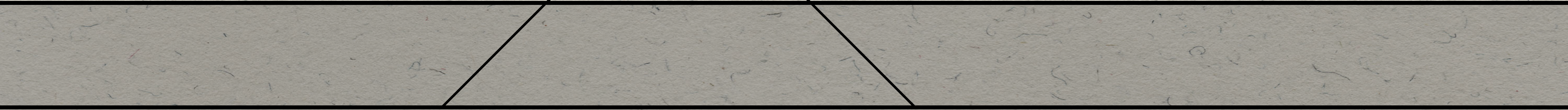


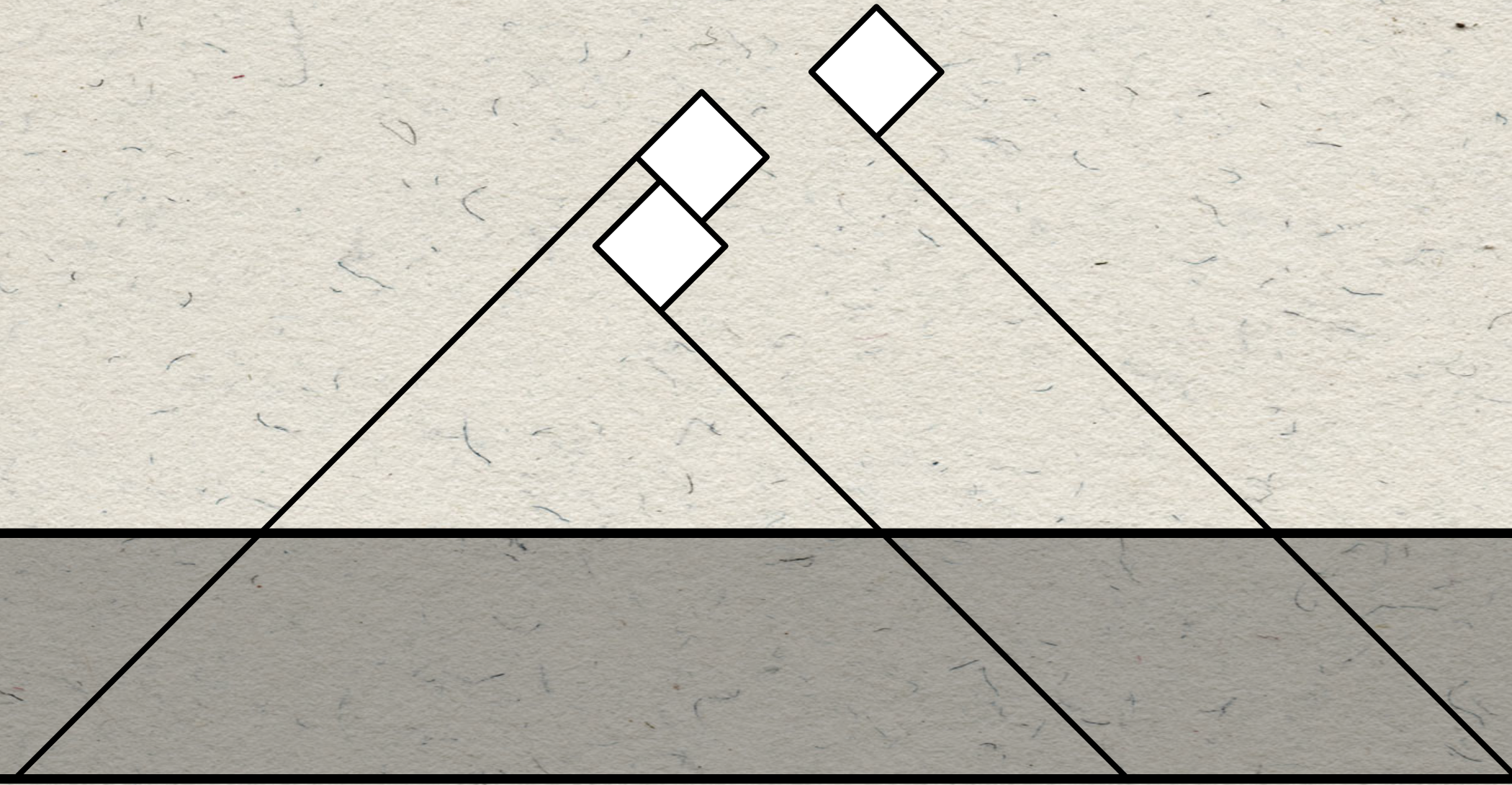
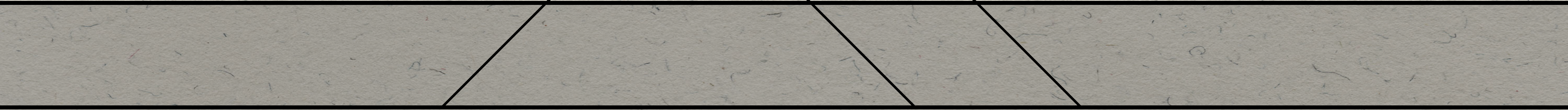


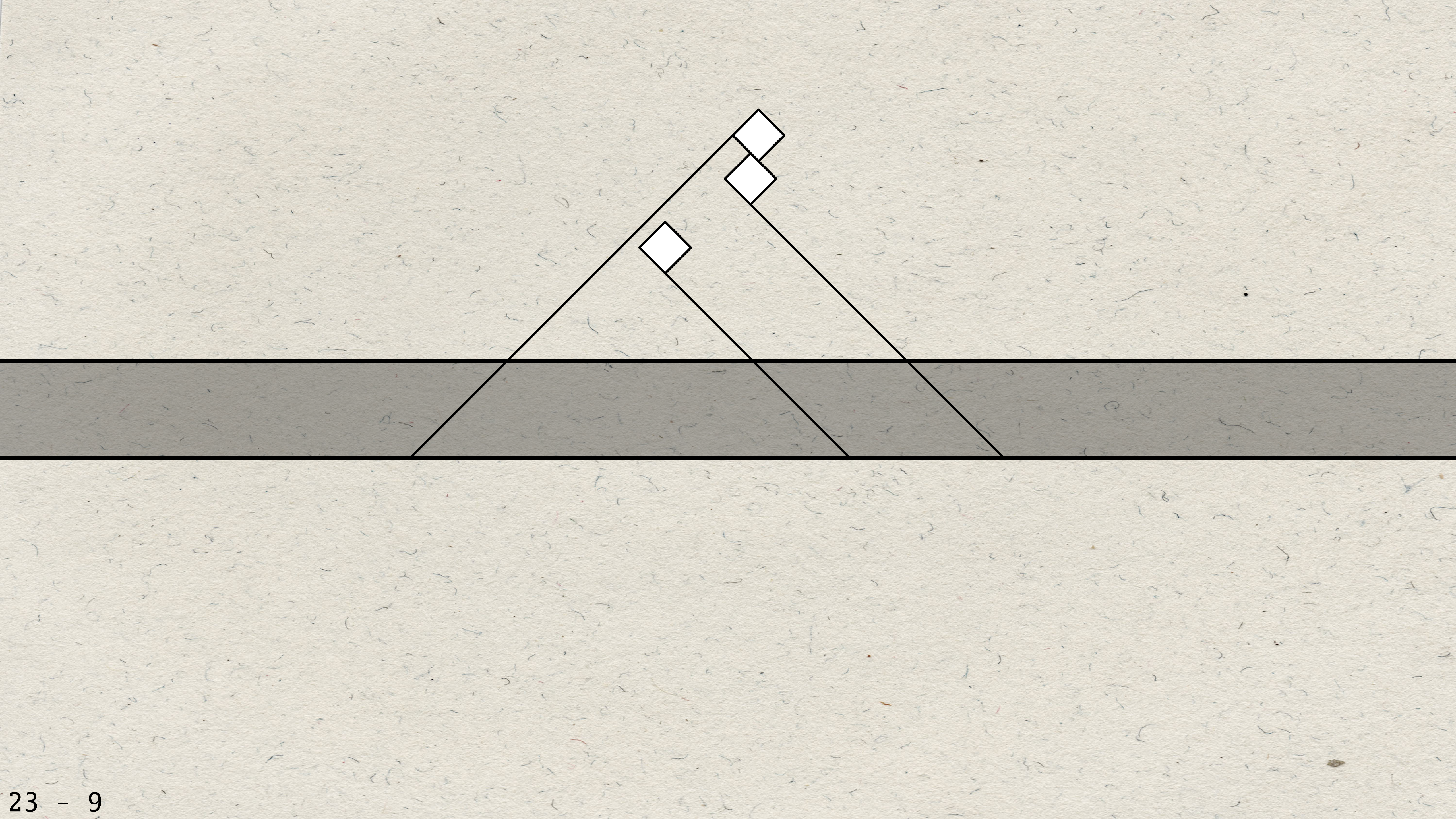


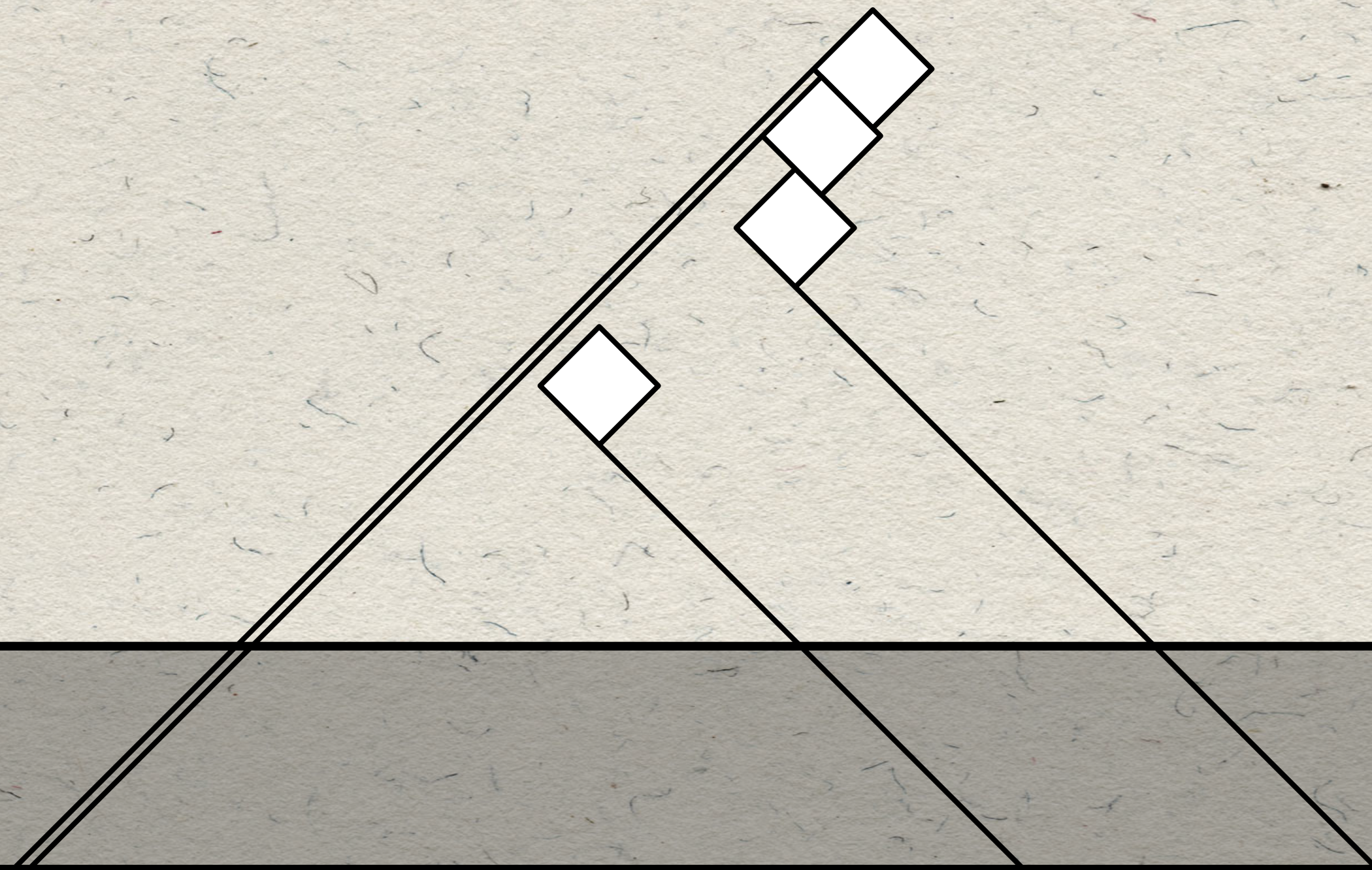


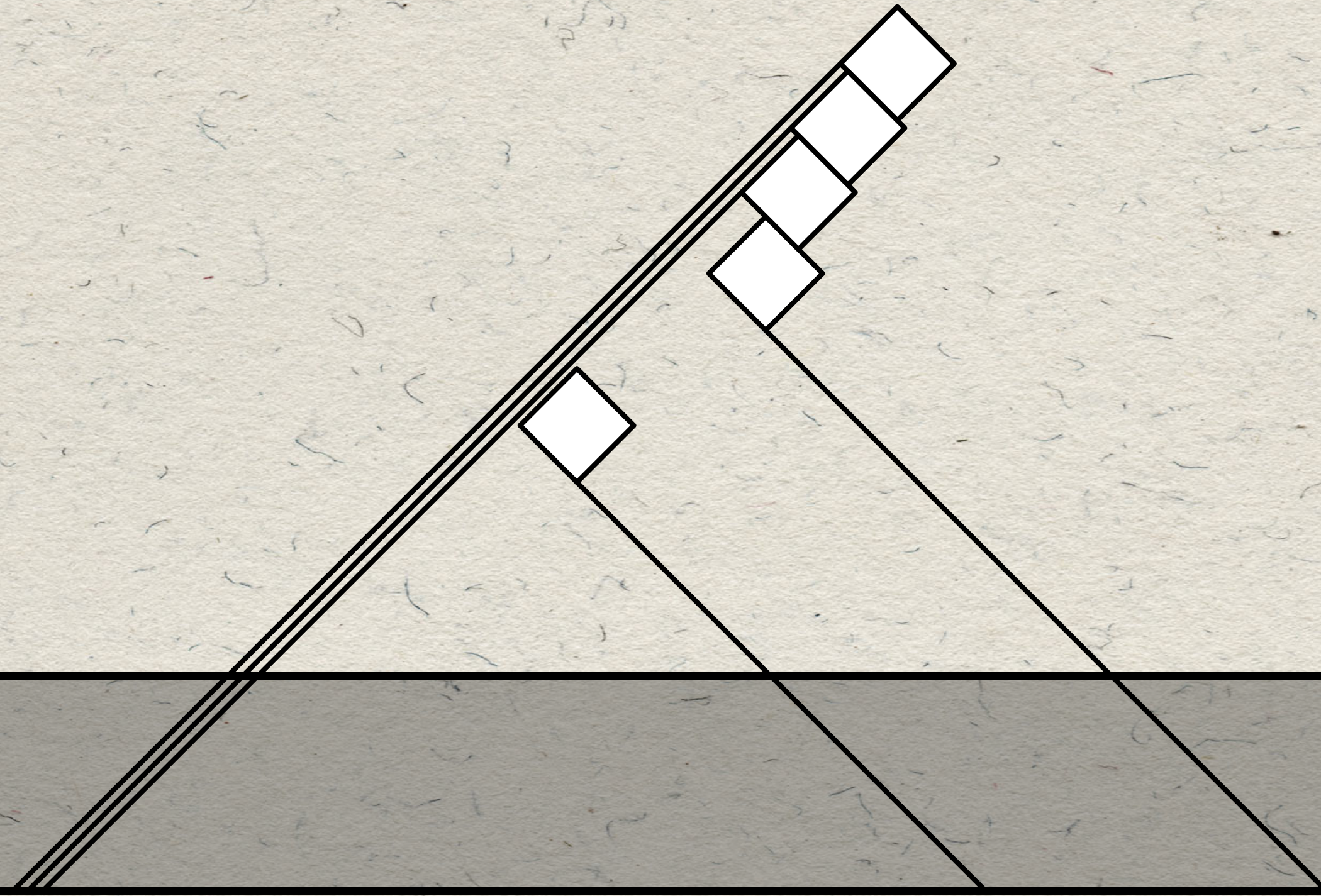


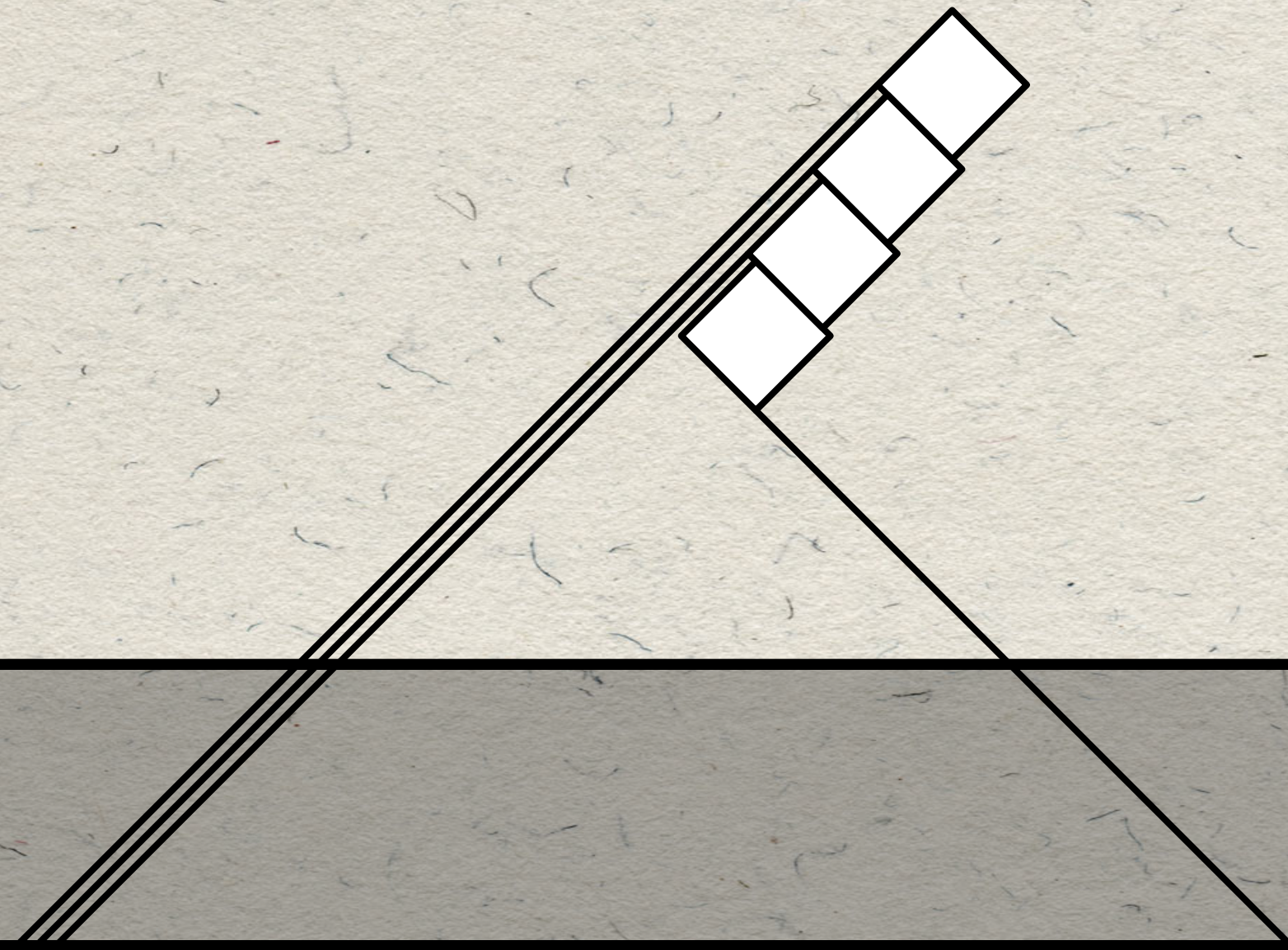












FUTURE WORK

