

# Sol LeWitt, Combinatorial Enumeration, and Rogue

Sol LeWitt, "All One, Two, & Three-Part Combinations of Three Transparent Colors", 1985

Labyrinth of Polyominoes, heptominoes

Mark Gritter (novice Artificer)  
Roguelike Celebration  
October, 2025

Rogue (PC port)

# Sol LeWitt



20th Century  
Conceptual Artist,  
1928-2007

- wall drawings
- minimalism
- lots of cubes!

← “Incomplete Open Cubes”, 1974

Photo by the speaker, at San Francisco Museum of Modern Art, 2018

# Wall drawings

Sol LeWitt, "Wall Drawing 289", 1976

A 6-inch (15 cm) grid covering each of the four black walls. White lines to points on the grids. Fourth wall: twenty-four lines from the center, twelve lines from the midpoint of each of the sides, twelve lines from each corner.

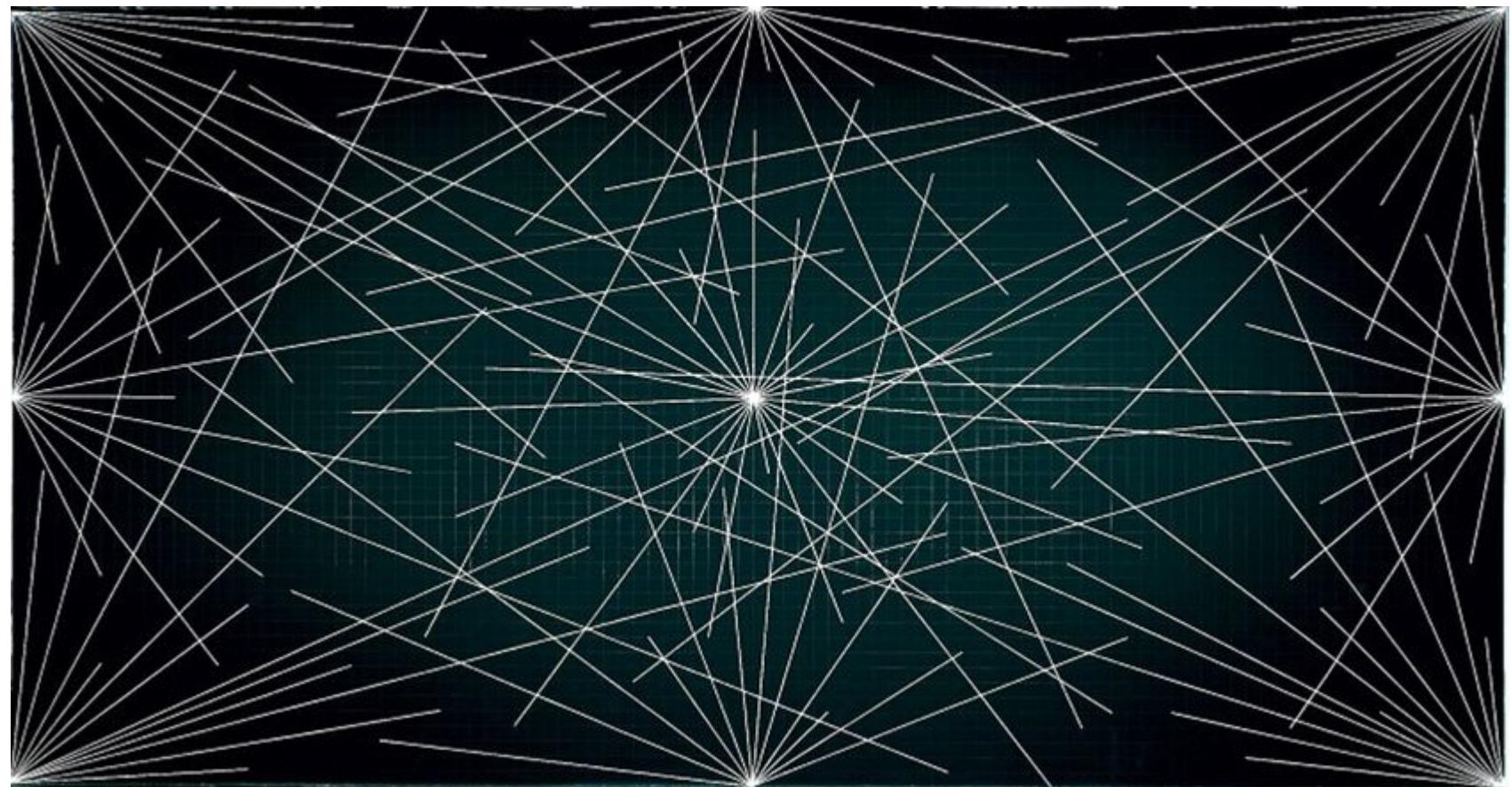


Image from <https://whitney.org/collection/works/25530>

# Wall drawings (2)

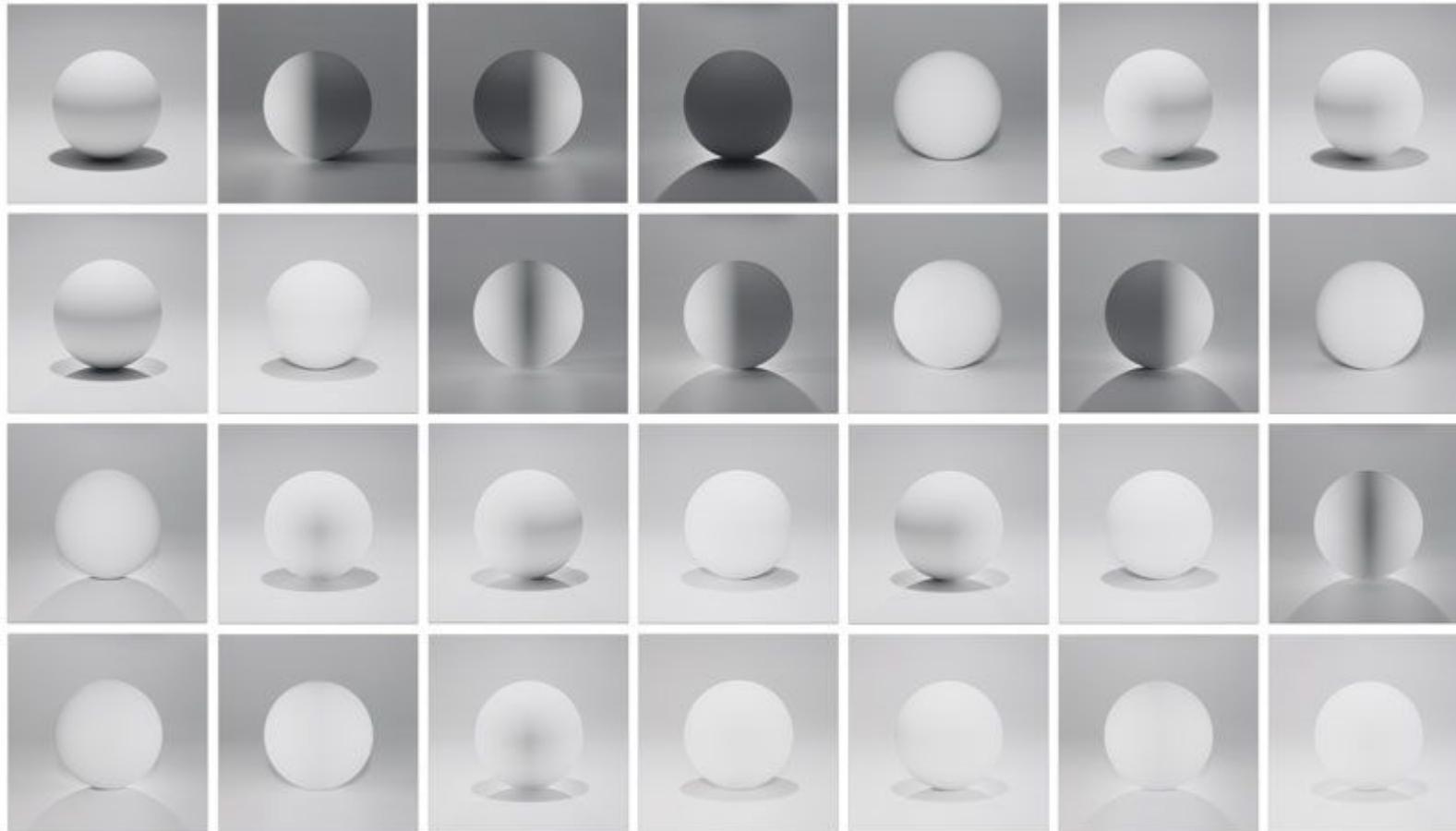
Wall drawings are generative art! They are given as **instructions** rather than as an image. (\*)

*“The only permanent, concrete form of Wall Drawing #289 is a set of typewritten guidelines and a certificate of authenticity signed by the artist... The exact angle and length of the lines are determined by those who draw them, and the work’s precise configuration and scale may be adapted to fit a variety of architectural contexts. Consequently, the wall drawing can differ significantly with each realization.”* – Whitney Museum of American Art,

<https://whitney.org/collection/works/25530>

(\*) this separation is ruined a bit by having to hire a draftsperson from the LeWitt Studio to execute an “official” drawing.

# Serial Art



Sol LeWitt, “A sphere lit from the top, four sides, and all their combinations”, 2004

Image from  
[sollewittprints.org](http://sollewittprints.org)

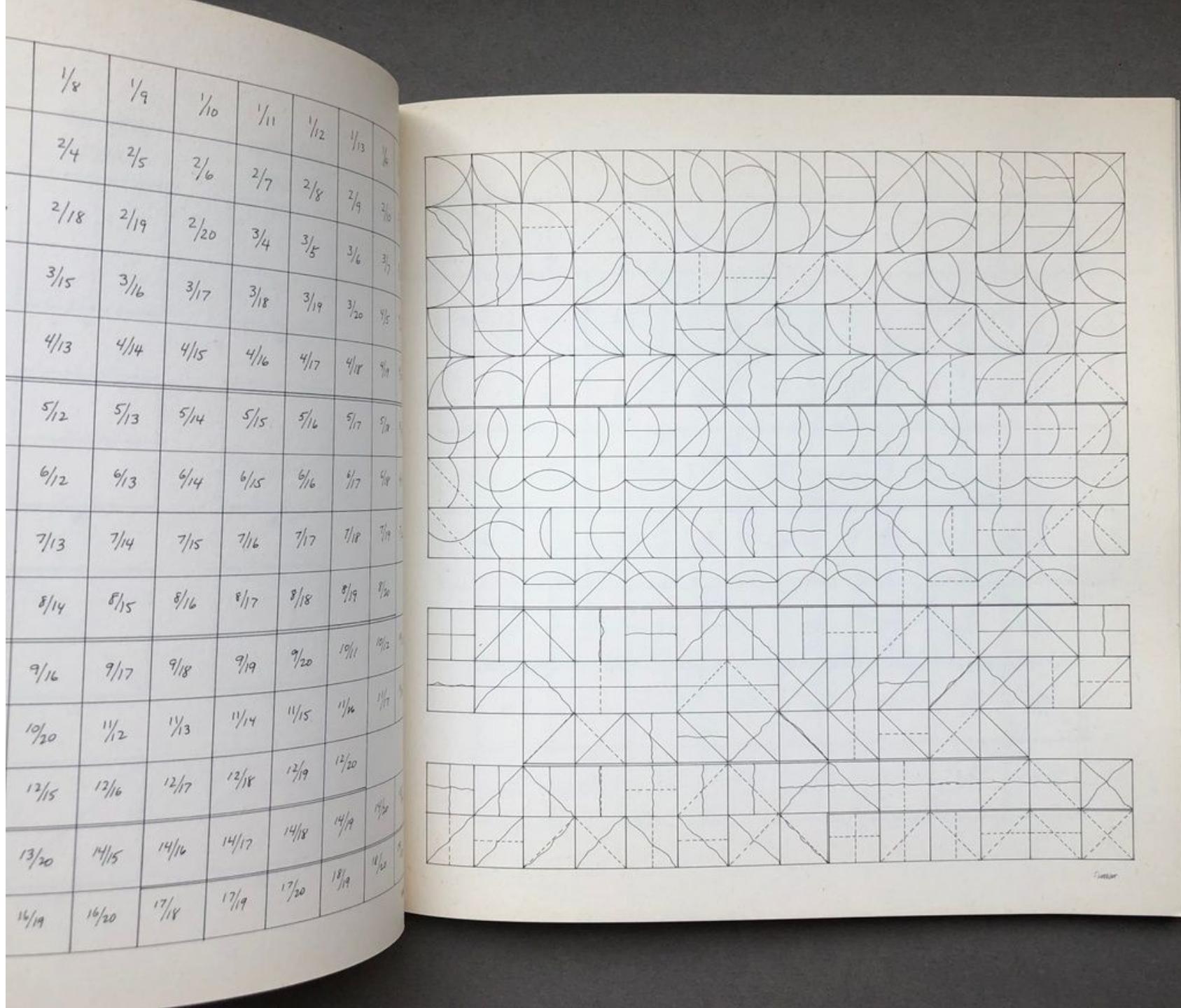
(P.S. there should be  
32, or 31, not 28?)

# “Arcs and Lines”, Sol LeWitt, 1974

“All combinations of arcs  
from four corners, arcs from  
four sides, straight lines,  
not-straight lines, and  
broken lines”

Image from  
<https://www.printedmatter.org/catalog/14771/>

(A used copy will set you back  
about \$200-\$350 so I don't, alas,  
own this.)



# Serial Art (2)

This takes a further step backwards into abstraction. The artwork is a description of a condition or axiom or law, which is then carried out to produce a tangible result. The artwork typically juxtaposes all of these results together as a finished work.

*"The serial artist does not attempt to produce a beautiful or mysterious object but functions merely as a clerk cataloguing the results of his premise."* – Sol LeWitt

(I disagree, I think these are beautiful and mysterious. – Mark)

# Serial Art (3)

*“Three basic operating assumptions separate serially ordered works from multiple variants:*

- 1. The derivation of the terms or interior divisions of the work is by means of a numerical or otherwise systematically predetermined process (permutation, progression, rotation, reversal).*
- 2. The order takes precedence over the execution.*
- 3. The completed work is fundamentally parsimonious and systematically self exhausting.”*

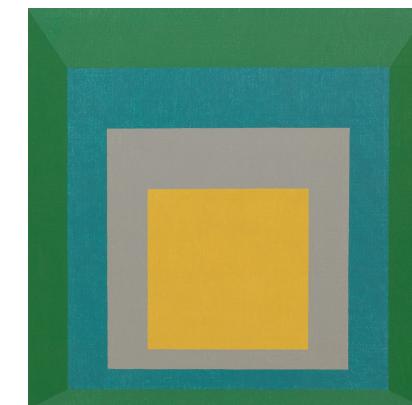
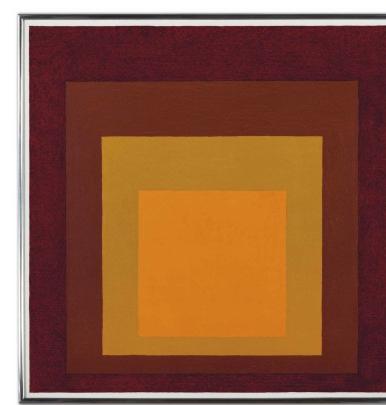
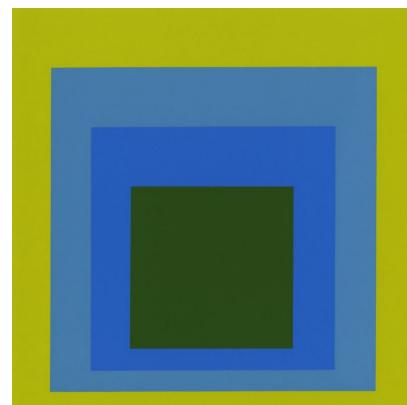
– Mel Bochner, “The Serial Attitude”, Artforum, December 1967

(Of course, not everybody agrees with this either – Mark)

Josef Albers, “Homage to the Square”. Photo by Selena N. B. H. (cc-by-2.0) at the Tate Modern.



“a rigorously formulaic project comprising more than one hundred paintings and prints and developed over twenty-five years.” – MOMA.org



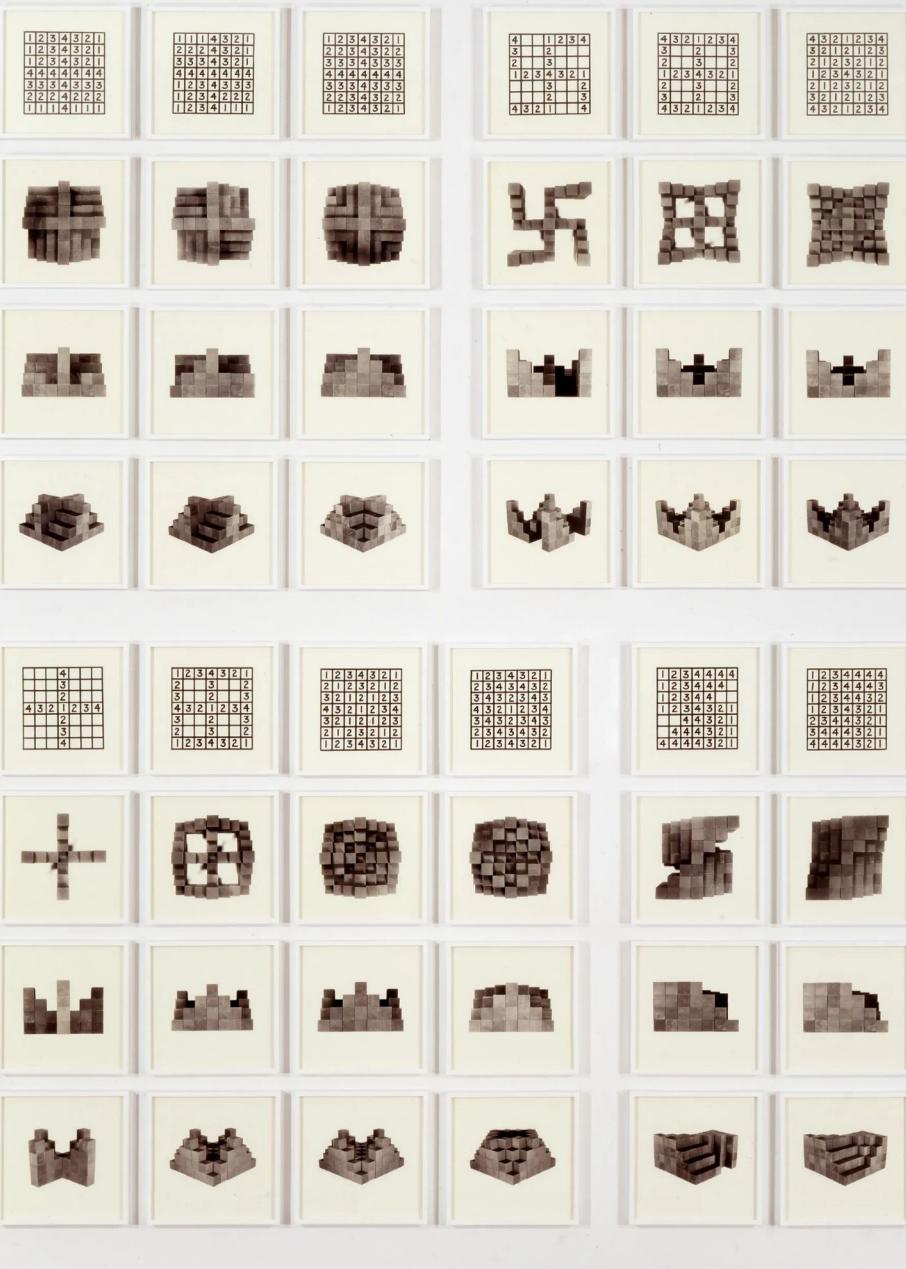


**Yayoi Kusama,  
“Infinity  
Mirrored  
Room -  
Aftermath of  
Obliteration  
of Eternity”,  
2009**  
from  
[https://hirshhorn.si.  
edu/kusama/infini  
ty-rooms/#aftermat  
h](https://hirshhorn.si.edu/kusama/infinity-rooms/#aftermath)

# Hanne Darboven, (Süd-) Koreanischer Kalender, 1991

<https://spruethmagers.com/artists/hanne-darboven/>





Mel Bochner, “36 Photographs and 12 Diagrams”, 1966,  
<https://www.glenstone.org/artworks/36-photographs-and-12-diagrams>

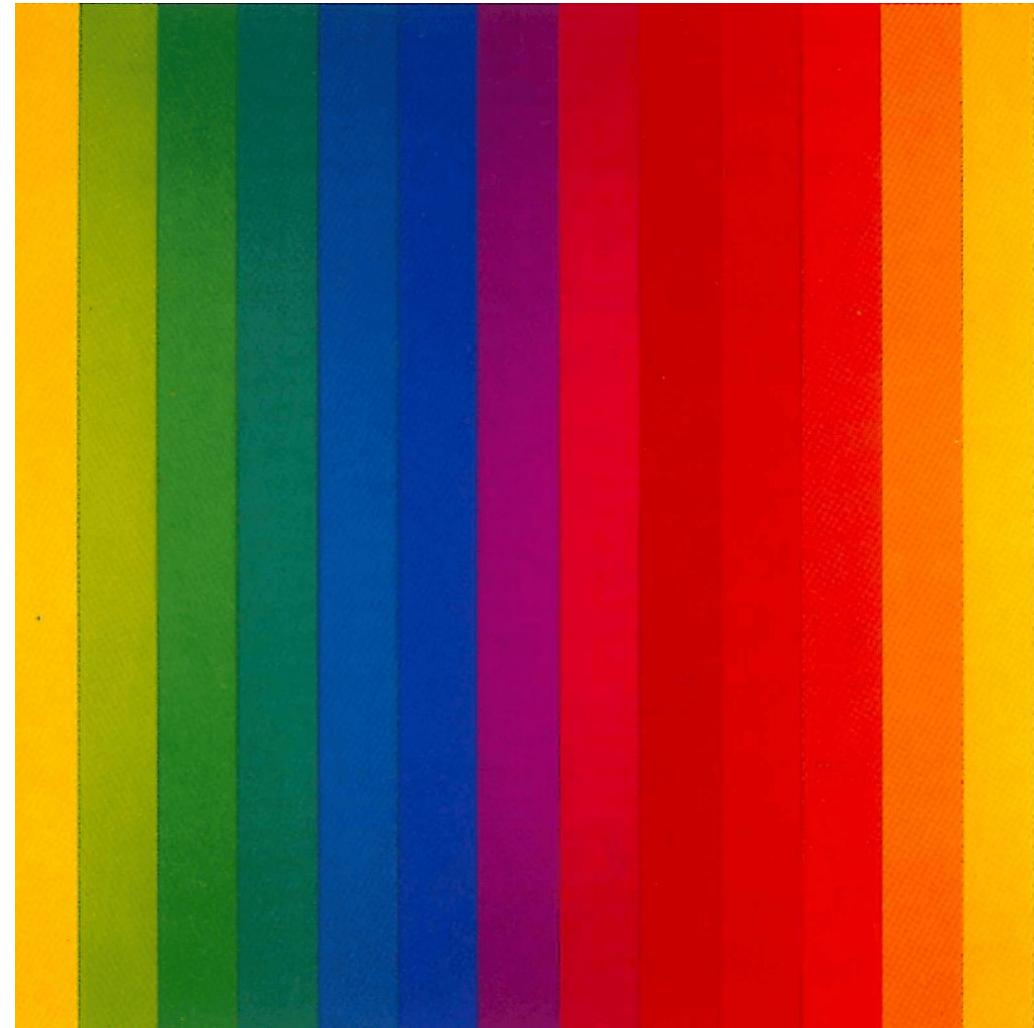
01 2 3 4 5 6 7 8 9 10 11 12 0 1 5 Σ Υ 2 2 Γ 8 Ρ 0 1 11  
 13 14 15 16 17 18 19 20 21 22 1 5 0 5 Ρ 1 8 1 Ι 1 2 1 Υ 1 Ε 1 Σ 1  
 23 24 25 26 27 28 29 30 31 5 5 Ε 5 Υ 5 2 5 2 5 1 5 8 5 Ρ 5 0 5  
 32 33 34 35 36 37 38 39 40 Ρ Ε 8 Ε Γ Ε 2 Ε 2 Ε Υ Ε Ε 5 5 1 5  
 41 42 43 44 45 46 47 48 49 0 4 1 4 5 4 Ε 4 Υ 2 4 2 4 8 4  
 50 51 52 53 54 55 56 57 58 Γ 2 2 2 2 2 2 2 2 2 1 2 0 2 4 4  
 59 60 61 62 63 64 65 66 67 8 2 Ρ 2 0 2 1 2 5 2 2 2 2 2 2 2 2 2 2  
 68 69 70 71 72 73 74 75 76 2 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
 77 78 79 80 81 82 83 84 85 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
 86 87 88 89 90 91 92 93 94 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
 95 96 97 98 99 100 101 102  
 103 104 105 106 107 108 109 0 1  
 110 111 112 113 114 115 116 117 1  
 118 119 120 121 122 123 124 125 2  
 126 127 128 129 130 131 132 133 2

Mel Bochner, “Counting (Double Over)”, 1995,  
<https://www.melbochnerprints.org/krakow-1994-01/#198>

Richard Paul Lohse, “Dreissig vertikale systematische Farbreihen in gelber Rautenform”, 1943/1970  
from [https://www.lohse.ch/works\\_paintings3\\_e.html](https://www.lohse.ch/works_paintings3_e.html)



Ellsworth Kelly, “Spectrum IV”, 1967  
<https://www.miandn.com/exhibitions/ellsworth-kelly/installation-views?view=thumbnails>

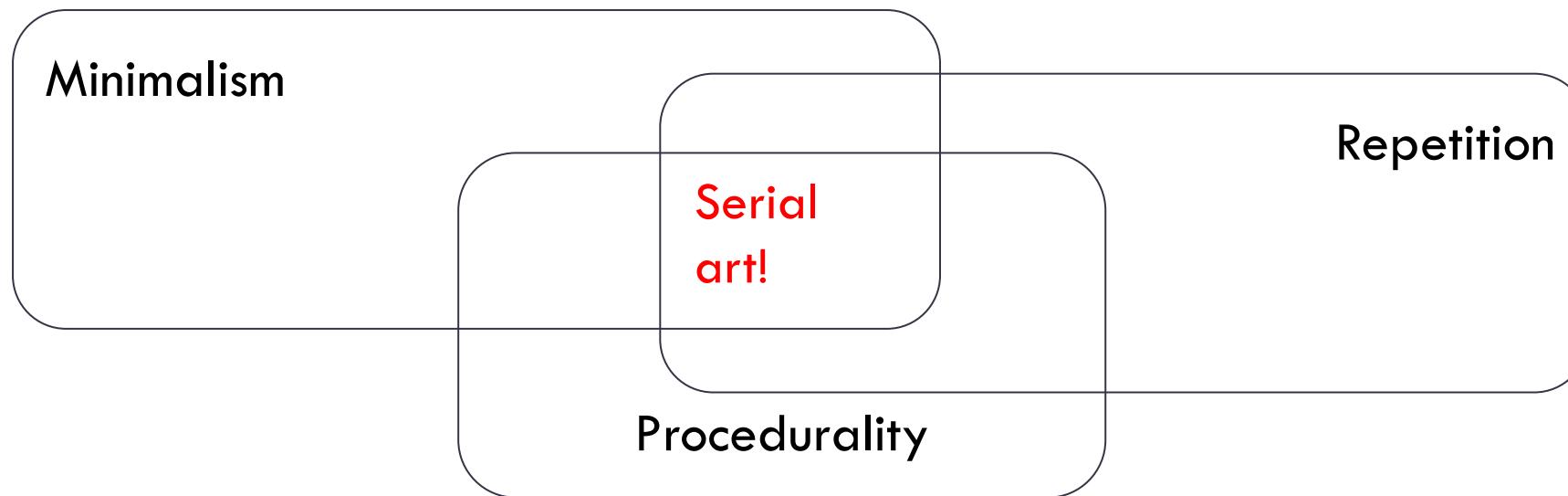


# Other Serial Art

- “Change Ringing” – playing all permutations of a set of bells

# The idea becomes a machine that makes the art.

– Sol LeWitt, “Paragraphs on Conceptual Art”, 1967



heptominoes ( $n = 7$ )

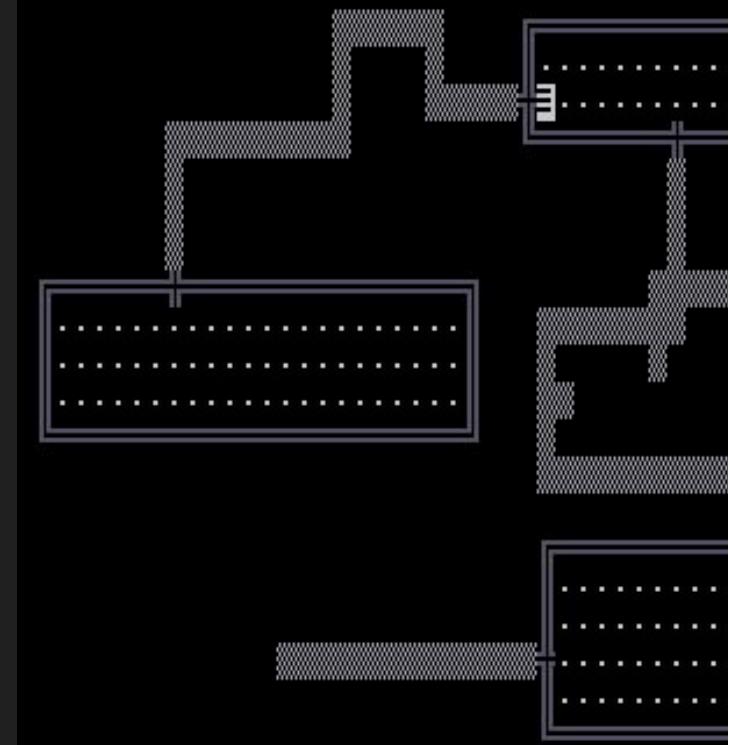
108



Level:1

Hits:12(12)

Str:16(16)



# Enumerating All the Things

# The Labyrinth of Polyominoes

By Roguelike  
Celebration  
2024 speaker  
@tesseralis!

All polyominoes  
(up to  $n=8$ ) here  
arranged by  
“genealogy”.

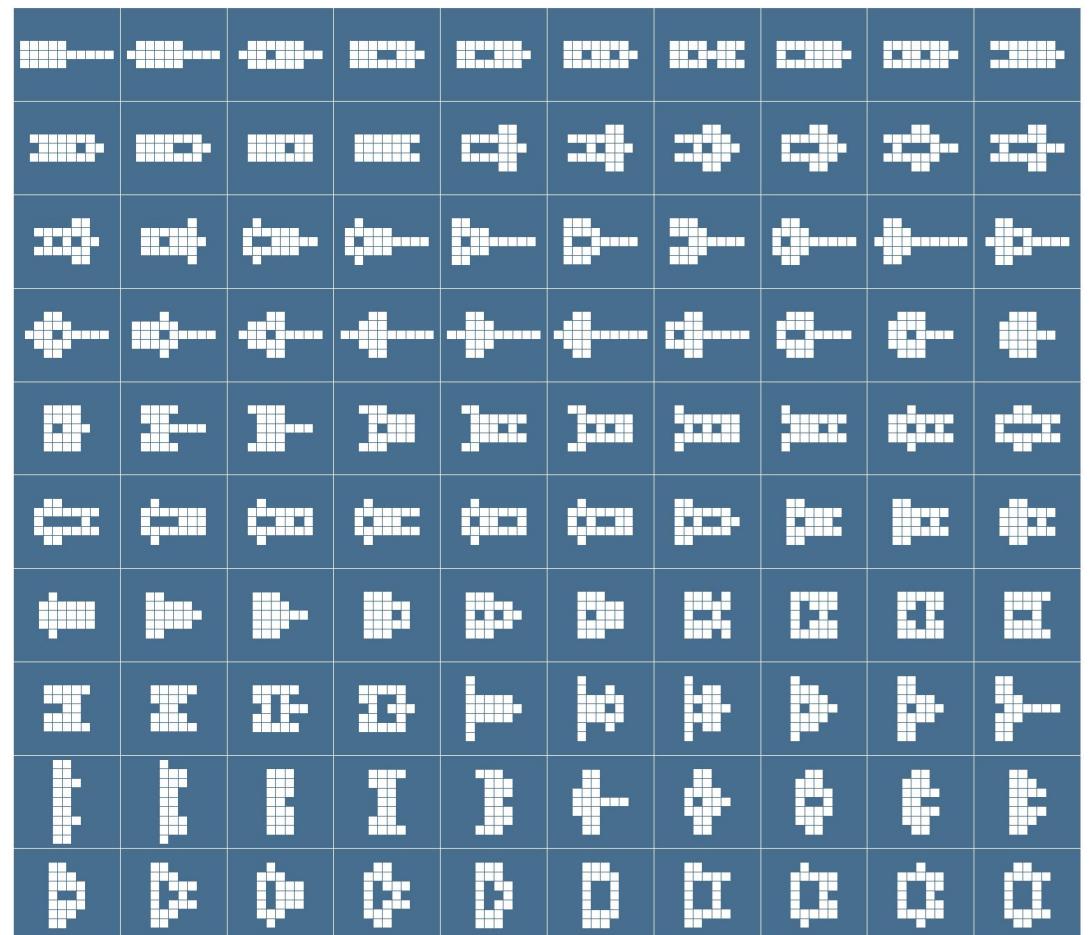
<https://minos.tessera.li/>



# All symmetric shapes constructible from the tetrominoes

Jacob Siehler, 2025

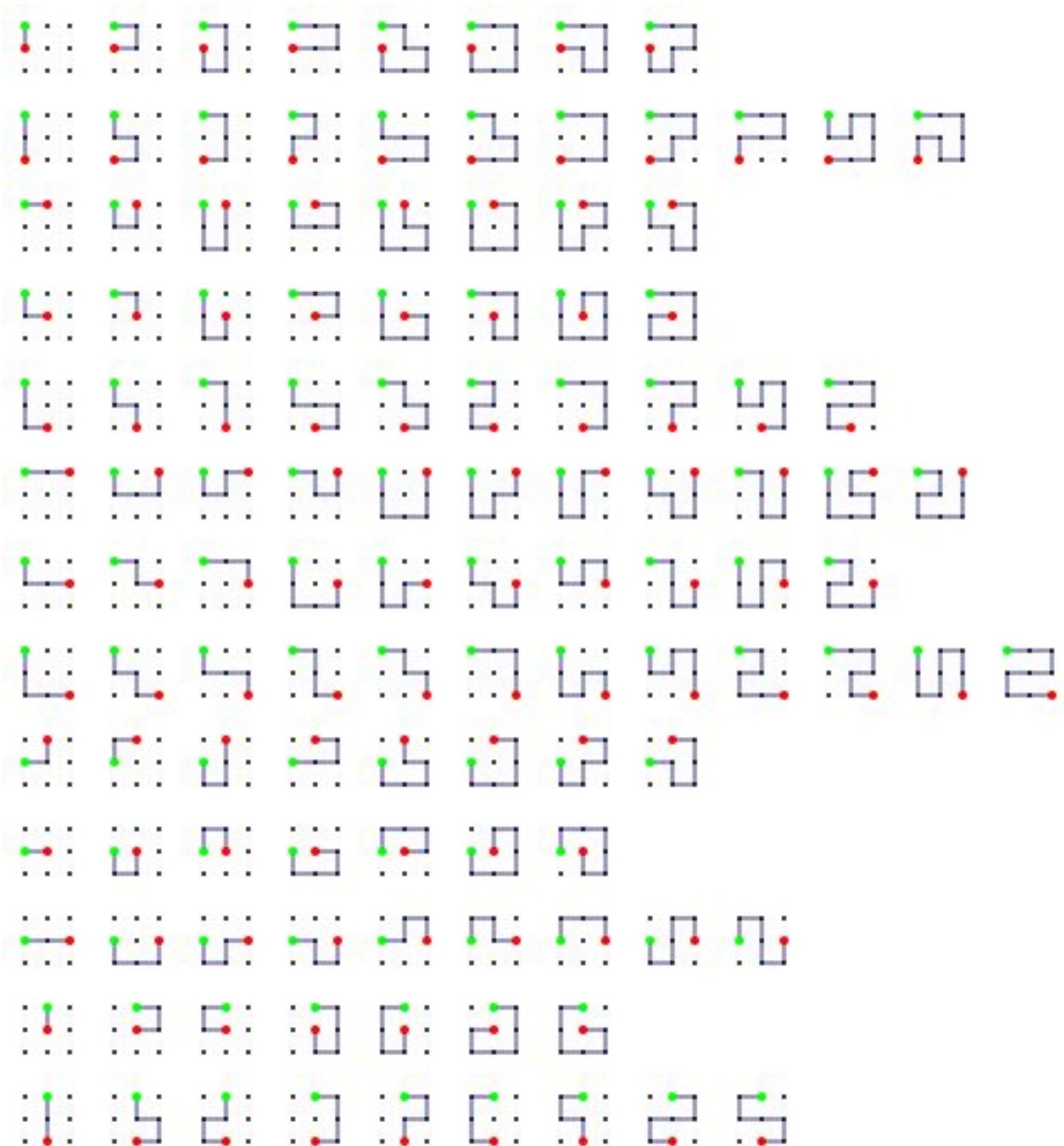
<https://mathstodon.xyz/@jsiehler/114744887537037820>



# All simple paths on a 3x3 lattice

Visualization by the speaker

These show directed paths from the green point to the red point, up to symmetries of the rectangle (so, flips but not rotations) and interchange of start and end.



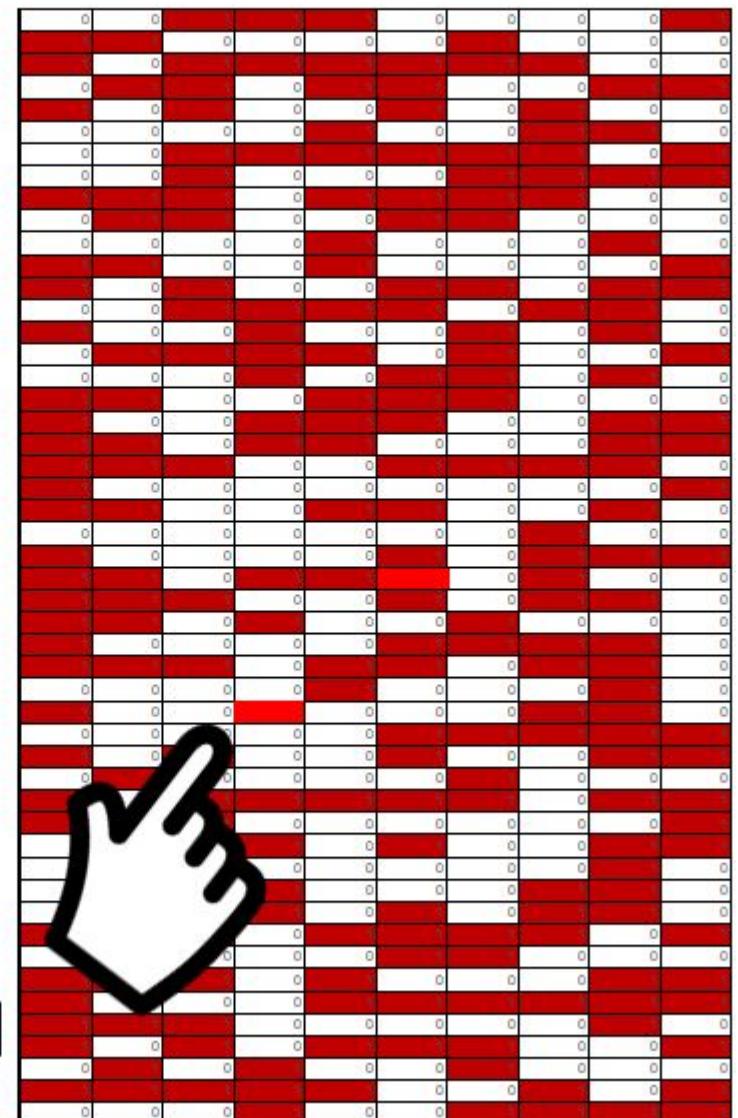
## Finite Melodic Dataset

# All the Music

Damien Riehl and Noah Rubin,  
<https://allthemusic.info/>

1. Generate every possible melody, up to 10 or 12 notes long.
2. Copyright them all.
3. Stop “substantially similar” music lawsuits forever?

Argument: either Riehl has copyrighted every unused melody, and they are free to use, or melodies aren’t copyrightable.

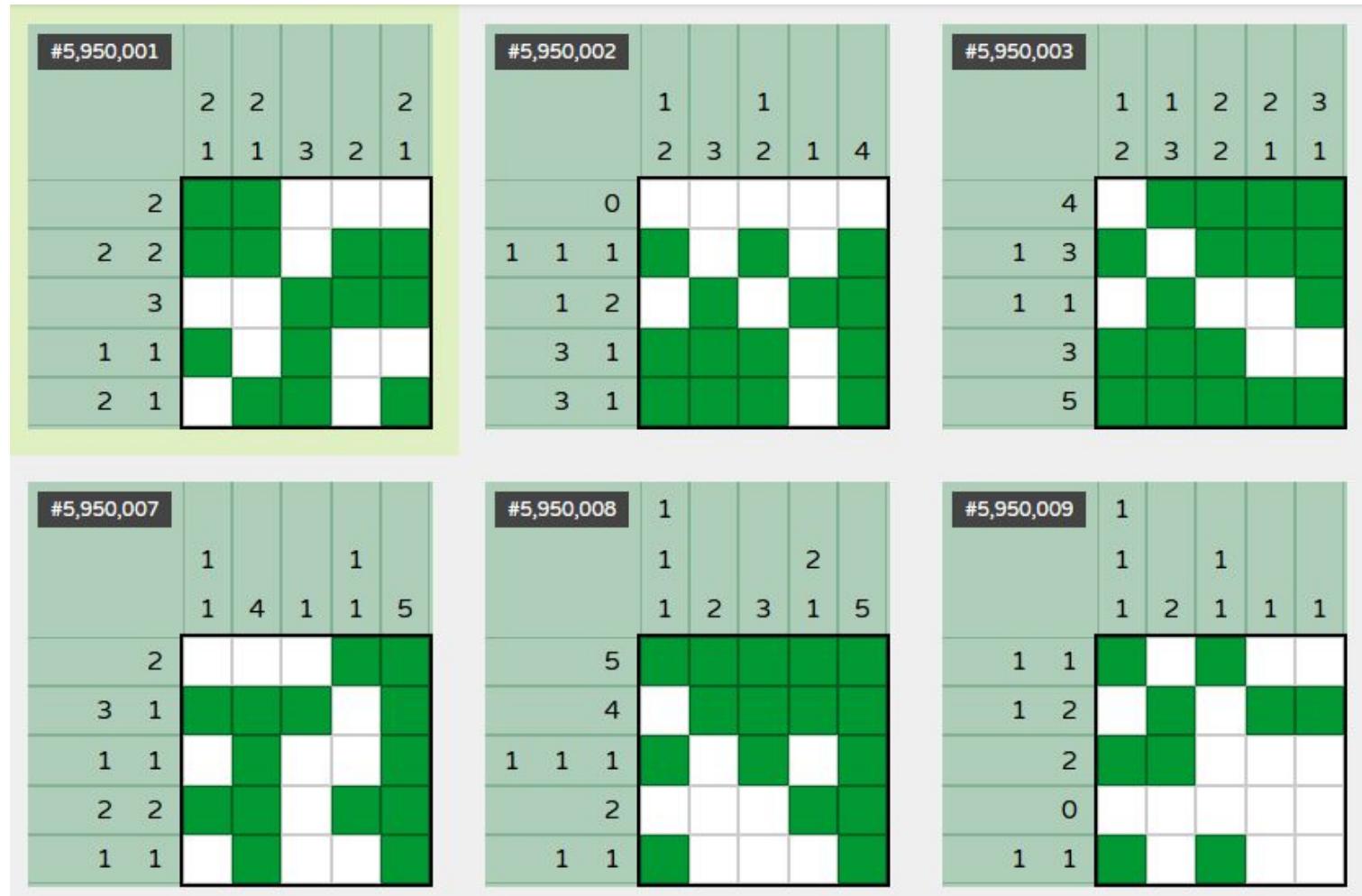


P.S. in the age of generative AI you should be able to find the unfortunate flaw in this argument.

# Every 5x5 Nonogram

Okayest Studio,  
[https://pixelogic.app/  
every-5x5-nonogram](https://pixelogic.app/every-5x5-nonogram)

24,976,511  
possibilities, all solved  
by humans via a  
collaborative effort.



# SmallCategories

Ben Spitz,  
<https://smallcats.info/>  
("in beta")

A “category” is a mathematical object consisting of dots with arrows in between them, obeying a couple simple rules.

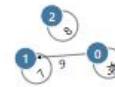
164,975 categories each with properties listed, the “multiplication table” of composition, and a visualization.

## Quick Reference

Web ID	6ca55ed1-f425-4004-a199-29def9560838
Morphisms	9
Objects	3
Index	59248
Name	N/A
Description	N/A

## Visualization

This feature is still janky! Please excuse the mess.



Morphisms 0 through 2 are identities, shown as objects in this visualization.

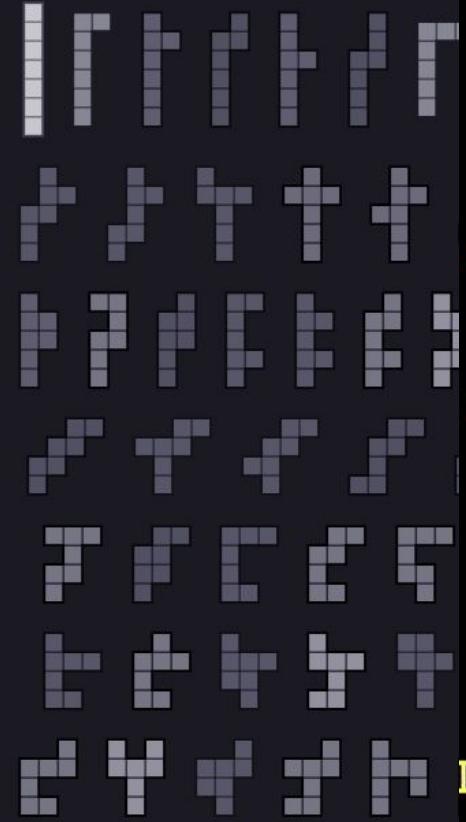
## Table

row ∘ col	0	1	2	3	4	5	6	7	8
0	0	/	/	3	4	5	/	/	/
1	/	1	/	/	/	/	6	7	/
2	/	/	2	/	/	/	/	/	8
3	3	/	/	3	4	5	/	/	/
4	4	/	/	3	4	5	/	/	/
5	5	/	/	3	4	5	/	/	/
6	6	/	/	6	6	6	/	/	/
7	/	7	/	/	/	/	6	7	/
8	/	/	8	/	/	/	/	/	2

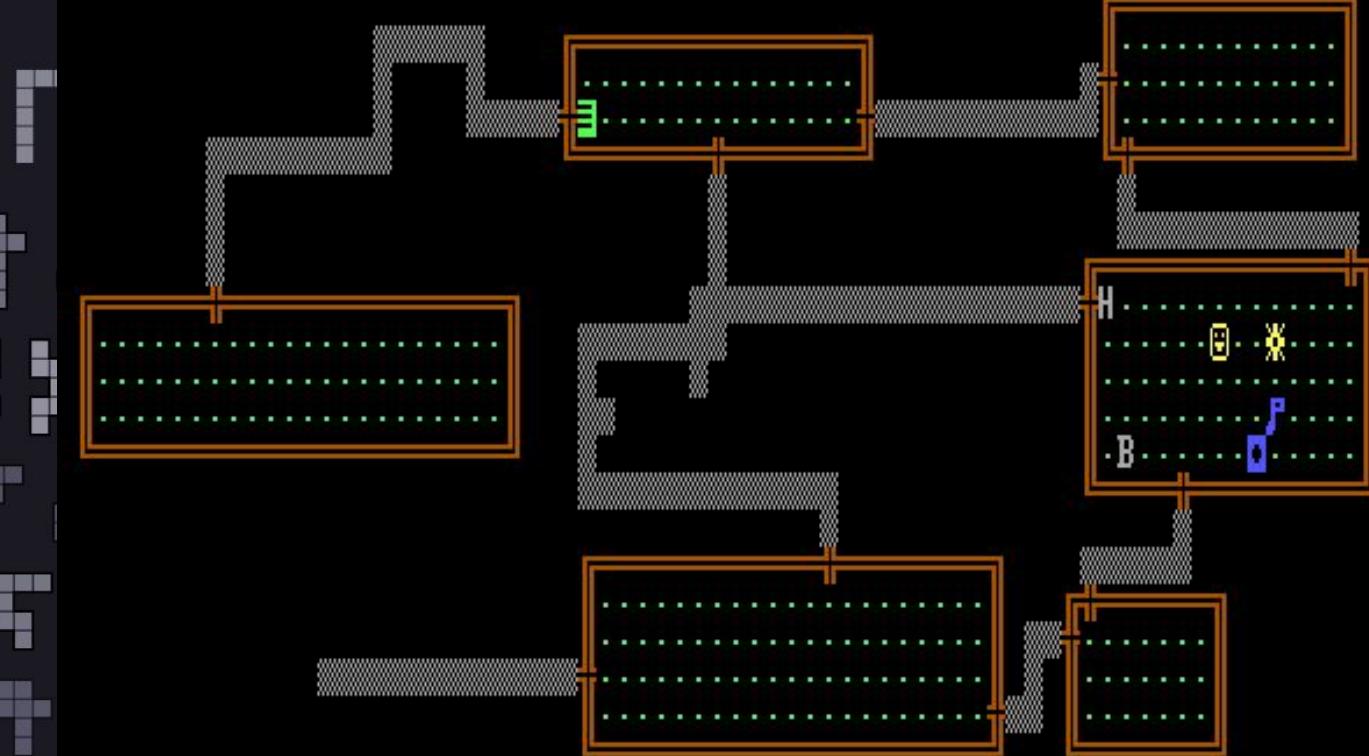
## Facts

- has\_terminal\_object  has\_binary\_products
- is\_discrete  is\_terminal  has\_initial\_object
- has\_binary\_coproducts  is\_groupoid  is\_monoid
- is\_connected  has\_finite\_products  is\_preorder
- is\_skeletal  has\_equalizers  has\_finite\_coproducts
- is\_complete  is\_cocomplete  has\_coequalizers
- is\_initial

heptominoes ( $n = 7$ )



Yellow	210	Red
Green	210	Blue
SELECT RED/BLUE		



Level:1

Hits:12(12)

Str:16(16)

Gold:75

Armor:5

Exp:1/5

10:23

... and Rogue

# Rogue Level Generator

9 rooms

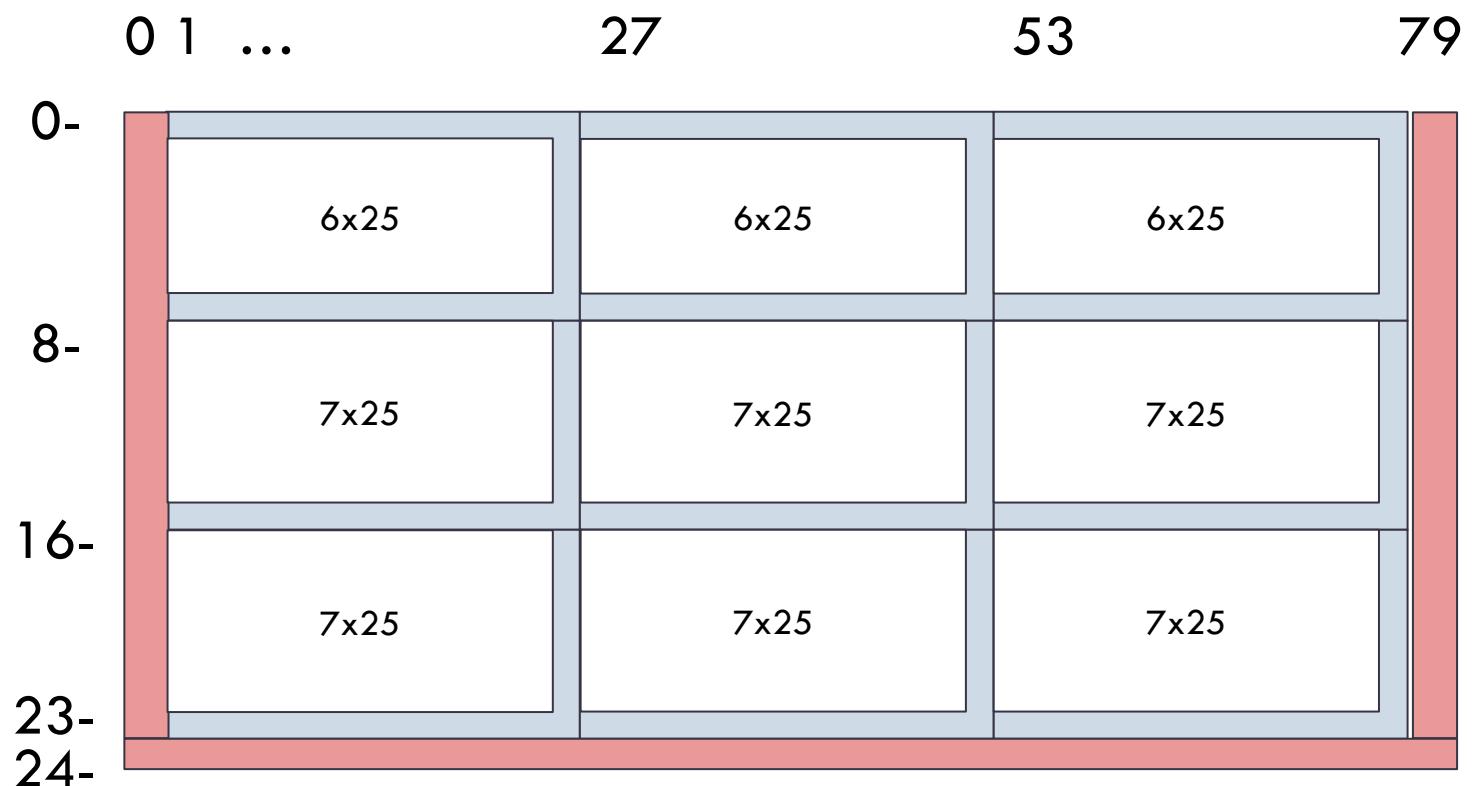
Up to 3 can be missing

For each room:

- Pick a width [4,25]
- Pick a height [4,7]

Pick a location within the “cell” that fits.

Start over if the room would be on the first row (so top rooms can only be 4-6.)



# Counting the Possibilities

Width 25	Width 24	....	Width 5	Width 4	
1 horizontal position	2 horizontal positions	...	21 horizontal positions	22 horizontal positions	=253 possibilities

Same calculation for height gives 6 (top row) or 10 (other two rows) possibilities.

So, there are about  $(253 \times 6)^3 \times (253 \times 10)^6$  possible combinations of room sizes, if no rooms are absent. That's about  $2^{(99.5)}$  possibilities.

- P.S. this number doesn't change much if we account for "gone" rooms

# Rogue as Serial Art

Rogue uses a PRNG (a linear congruential generator) with a 32-bit seed value.

There are only  $2^{32}$  possible versions of a level!

We could enumerate all of them.

- As an art project?
- To better understand the generator?
- I just thought it was cool that not only are there room configurations that are impossible, they vastly outnumber the ones that are valid.

# The Rogue RNG(s)

## **Original rogue:**

```
((seed = seed*11109+13849) >> 16) & 0xffff)
```

This is a standard Linear Congruential generator, but using bits 16-32 of the seed avoids some of its worst behaviors.

## **MS-DOS port:**

```
seed *= 125;  
seed -= (seed/2796203) * 2796203;  
return ((ran() + ran()) & 0x7fffffff)
```

“This is adapted from the FORTRAN version in ‘Software Manual for the Elementary Functions’ by W.J. Cody, Jr and William Waite.”

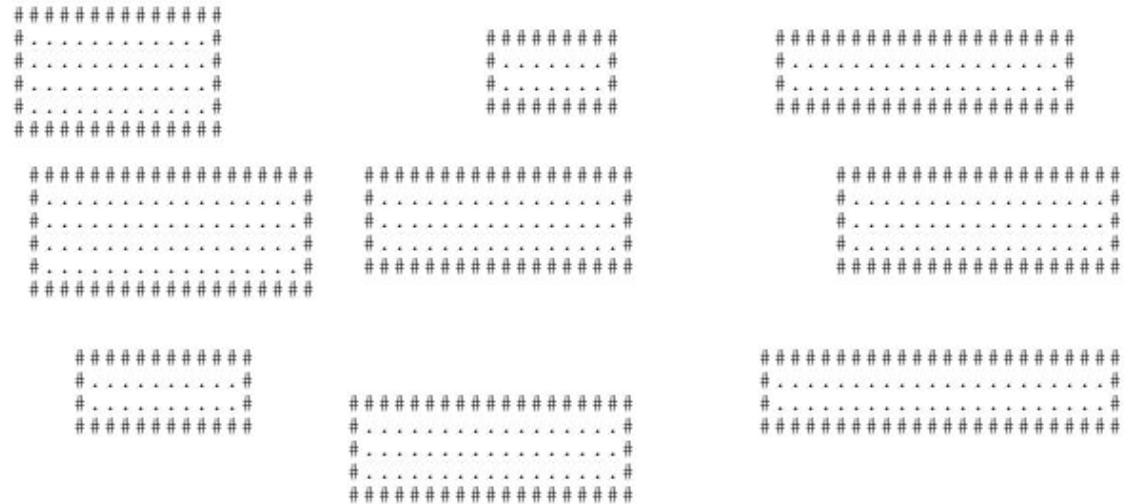
# It's not even hard!

I thought that this would take some nontrivial engineering work.

- Parallelize, implement on a GPU, ...?
- Implement an index for reverse lookup ...?

Nope, I copied the implementation into a C++ program (\*) and it only takes 11 minutes to run through every possible seed.

(\*) = turns out it is not possible to copy Rogue's room generation faithfully without also copying its monster and item generation, oops



first 100 seeds

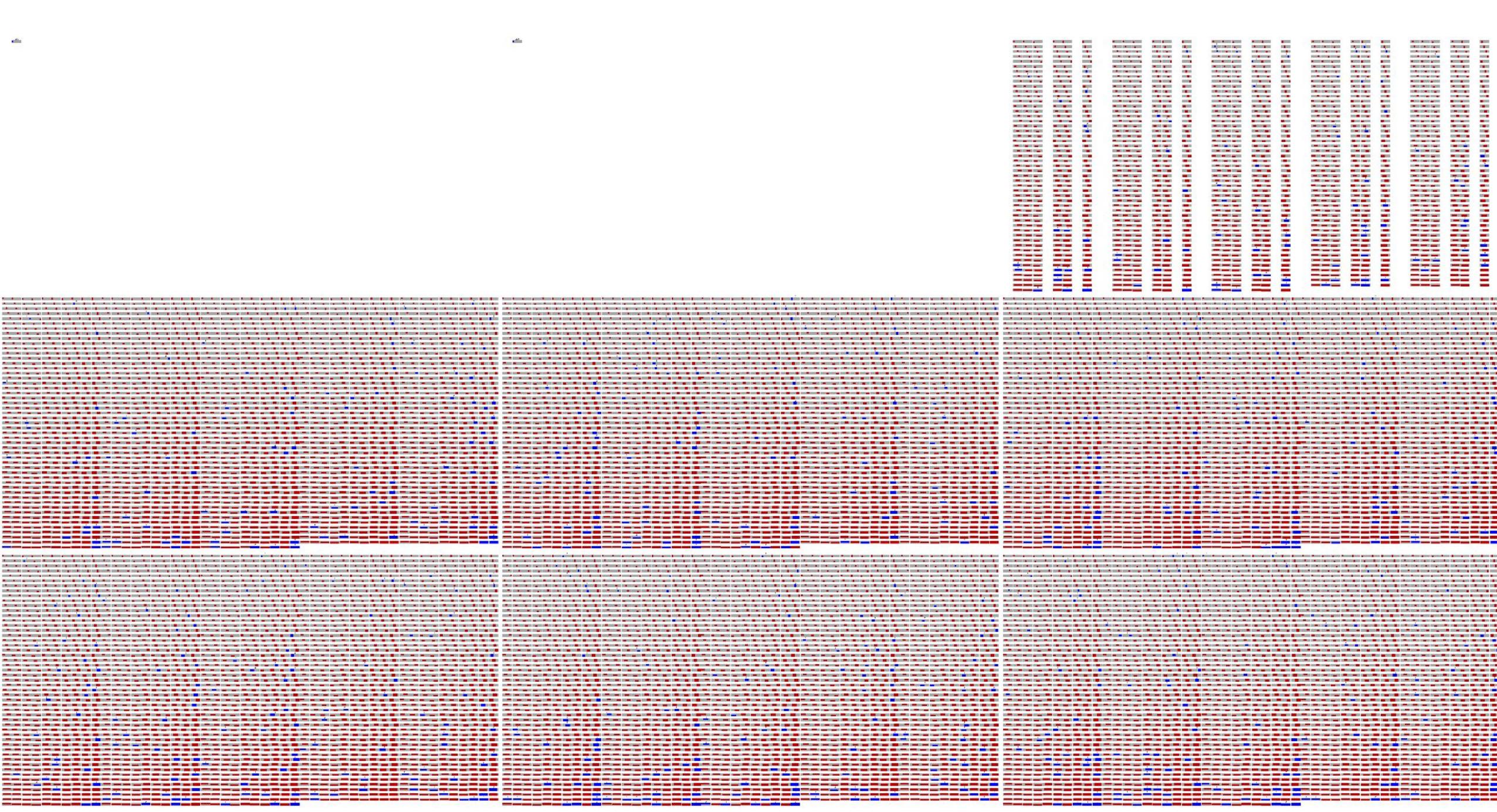
# Postselection

What if the top-left room is 4x4?

This happens 63,579,000 times. The distribution of the remaining rooms is about even.

(This plot shows how what % of time the given square is occupied.)

63579000 samples:



3589803	3591717
3566627	3568823
4042047	4034249
4750696	4748728
5656146	5656876
7057647	7056938
9426105	9423424

5308633
5239632
6076503
7058640
8506325
10645707
14182544

room 1

2347860	2367398	2374037
2647315	2615351	2588654
3026008	3034242	3053260
3022112	3036949	3052303
3536821	3479543	3453428
4227867	4250305	4272659
5438545	5182044	5357402
10765699	10405684	10725149

3147232	3143174
3547064	3543833
4035222	4039803
4040036	4036990
4730879	4725372
5653166	5652050
7217430	7220521
14331988	14335697

room 2

room 5

60	611439	614475	939278	920230	1844920
14	617531	618462	931147	904785	1849080
64	613442	612841	939391	919883	1842313
02	618072	619800	933088	905786	1848481
06	644145	645809	970803	967748	1924625
91	644248	645240	972214	966417	1929769

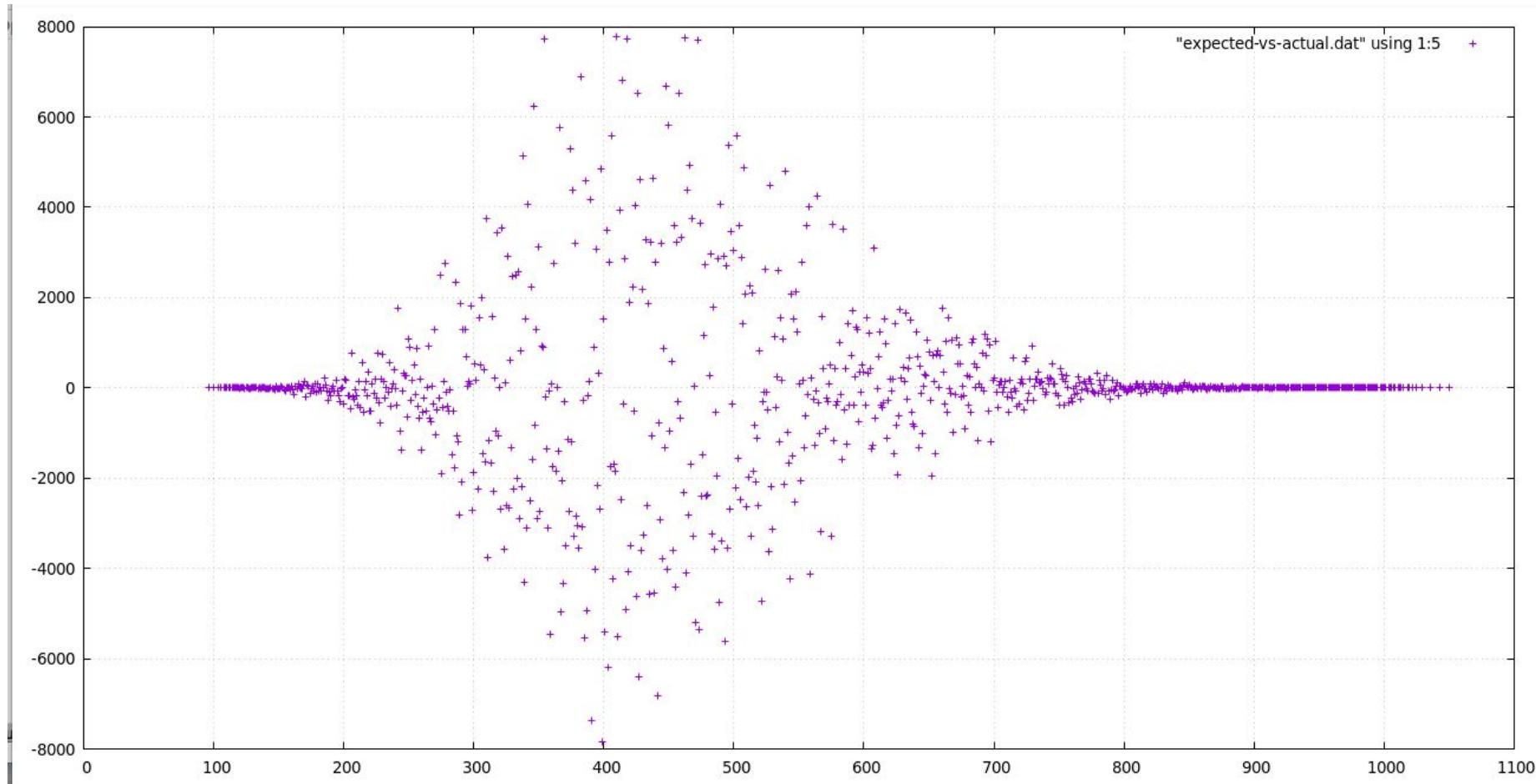
452167	469961	453436	471620	612723	611388	612952	935
465916	471365	461745	463226	609401	609582	609381	936
452854	468458	453127	471789	612599	612508	613384	934
466047	470498	460867	461442	608245	611542	610138	940
452717	468739	454236	473177	612034	613542	613231	935
485361	482926	485817	485267	650069	647611	649823	985

# Finding extremal seeds

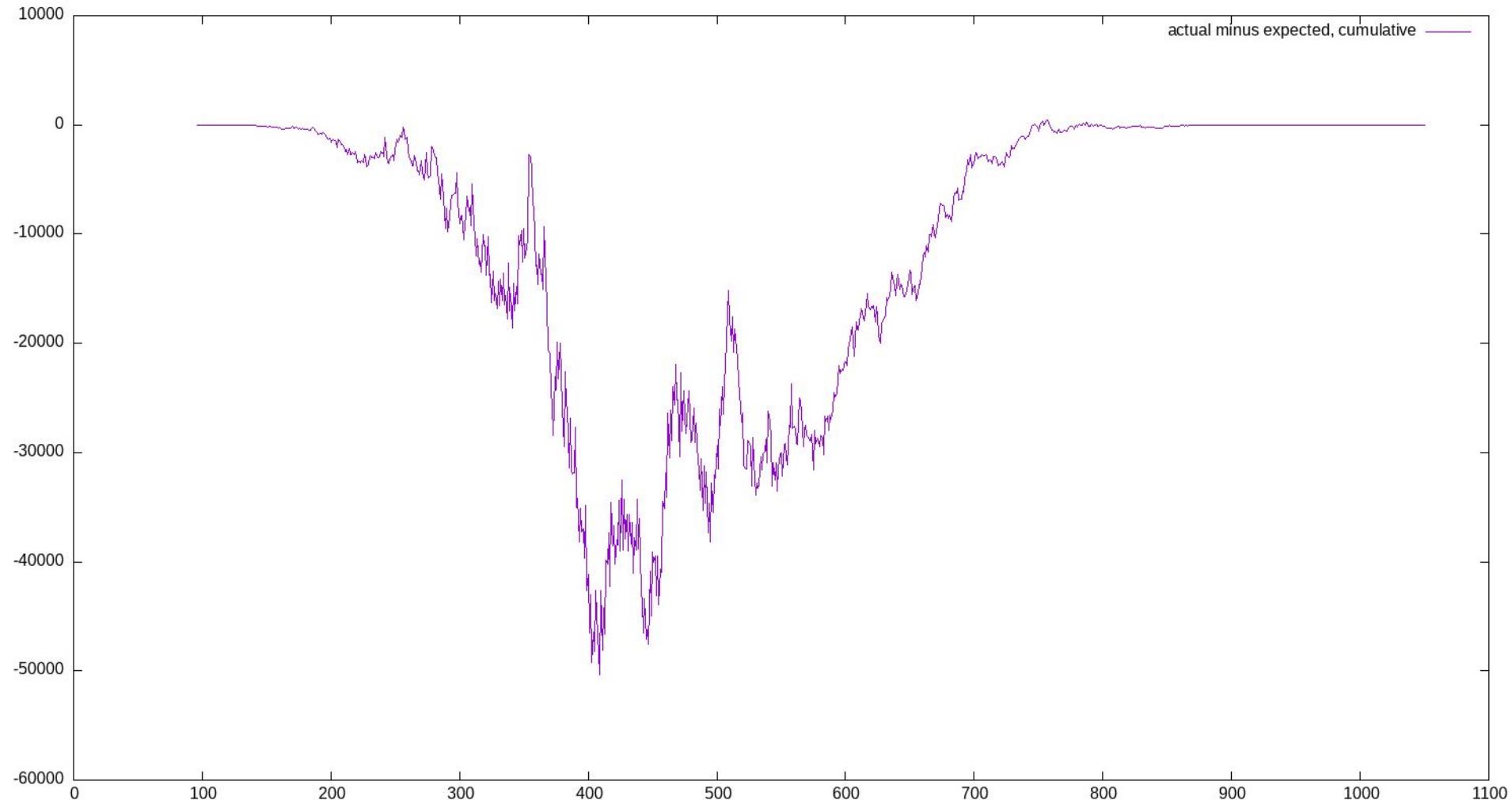
total room area	actual / cumulative	expected cumulative
96-110	0	2
112	2	7
113	4 (6)	9
116	7 (13)	21
117	6 (19)	26
118	5 (24)	30
120	18 (42)	55
121	16 (58)	69
122	11 (69)	82

There seems to be a deficit in how often the smallest areas appear, compared to their expected distribution (in the case where three rooms are gone, so 25% of the sample space.)

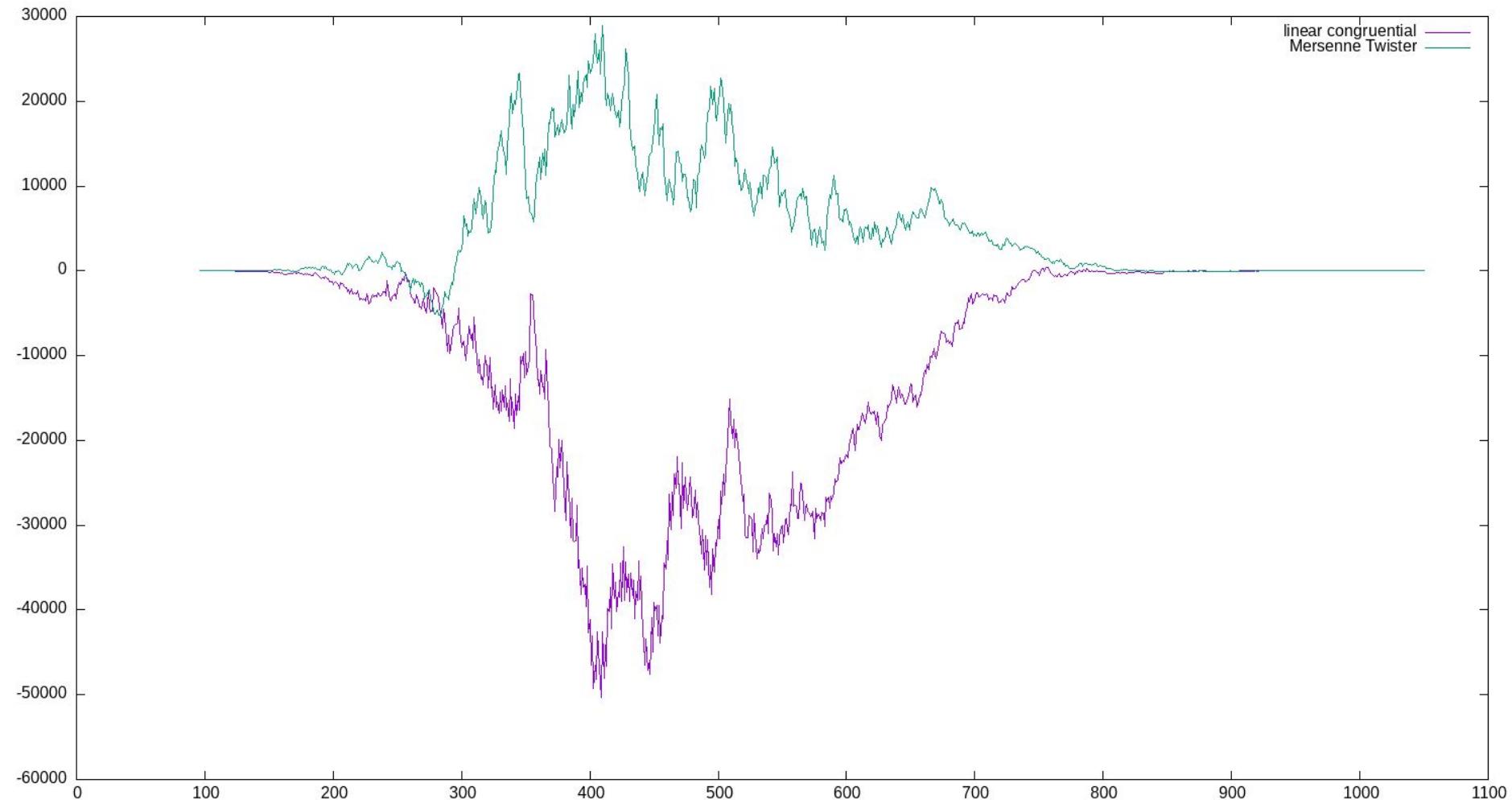
# Expected vs. actual (per size)



# Expected vs. actual (cumulative)



# Expected vs actual (comparison)



# What I'd like to find (but haven't)

- Some artifact of the room distribution that I can point to and say “this is because Rogue uses a linear congruential generator.”
- Some better visualization of the 36-dimensional space of room configurations

# Things to think about

Generators that use 32-bit seeds can be *fully explored*.

- As science!
- As serial art!
- As CYA? !?
- To completely solve a nontrivial roguelike game?

What are other connections between Serial Art and procedural generation?

- I'd love to hear an actual art historian or critic weigh in!

Are there natural examples of games with seeds in the hundreds or millions rather than in the billions?

# THANK YOU!



2020 Roguelike  
Celebration speaker

**On Mastodon:** @markgritter@mathstodon.xyz

**On Bluesky:** @markgritter.bsky.social

**On Github:** mgritter/rogue-room-generation

**In Real Life:** Principal Engineer at [thirdlaw.io](https://thirdlaw.io)



More Sol LeWitt!  
at NYMOMA and  
SFMOMA

# Things to think about (2)

Building a solver for real roguelike games— could easily “reconstruct” the seed by mistake. :(

- Once we have seen as little as 3 rooms, the value of the initial random seed can be determined, which turns the game from a random one into a deterministic one.

# S3 storage – about \$5/month

Amazon S3 > Buckets > rogue-star-bitmaps

**Objects (999+)**

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permission.

[Find objects by prefix](#)

[Actions ▾](#) [Create folder](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">00000000-room0-height4</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-height5</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-height6</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-width10</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-width11</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-width12</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-width13</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-width14</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-width15</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-width16</a>	-	July 12, 2025, 15:15:39 (UTC-05:00)	4.0 MB	Standard
<input type="checkbox"/>	<a href="#">00000000-room0-width17</a>	-	July 12, 2025, 15:15:41 (UTC-05:00)	4.0 MB	Standard

1 2 3 4 5