Conway's Game of Primes: Yet Another Twin Prime Sieve

Otto Dydakt
j.w.helkenberg@gmail.com

January 2025

Abstract

John Conway's conjecture that composite numbers exist in one of two states, trivial or nontrivial, evidently requires the existence of two types of prime number operator, trivial and nontrivial. One approach to this conjecture is to observe that the "Conway Nontrivial Numbers" exist as 24 classes of digital root and last digit preserving sequences. Conway's "Game of Primes" is then a sieving platform built for manipulating these classes. It is shown that Sloane's A142317 (1/24th of the Conway Nontrivial Primes) is equivalent to A201804 and that the complement to A201804 is rendered by 12 quadratic sequences. Following this, the twin primes are shown as equivalent to 9 OEIS sequences which are the union sets of pairs of Conway Nontrivial Prime sequences. Using these insights the twin prime conjecture is proved true for the union set of A201804 and A201816 (see: A224854) and expanded (as a relation) to cover any pair of the 24 Conway Nontrivial Prime Sequences.

Contents

1 PRIMEGAME 2.0: Twin Prime Sieve

The chibi-styled algorithm:
Address takes all values A001477, the non-negative integers 0,1,2,3... .
x takes all values of A000027, the positive counting numbers 1,2,3,4. . . .
WHERE $address - y = 0 = Composite$
ELSE
WHERE $address - y > 0$
WHERE $address - y \mod p \neq 0$
THEN number is a twin prime in A224854

To produce the members of A224854 feed the values of A001477 (address) and A000027 (x) into the sieve until $y \geq address$:

$$\text{address} - y \mod p \neq 0 \implies \text{Prime}(True)$$

$$\text{address} - (90x^2 - 120x + 34) \mod (7 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 94x + 10) \mod (7 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 86x + 6) \mod (11 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 78x - 1) \mod (11 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 76x - 1) \mod (13 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 90x + 11) \mod (13 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 120x + 38) \mod (17 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 104x + 25) \mod (17 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 132x + 48) \mod (19 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 94x + 18) \mod (19 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 90x + 17) \mod (23 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 86x + 14) \mod (23 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 132x + 48) \mod (29 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 104x + 29) \mod (29 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 108x + 32) \mod (31 + (90 \cdot (x - 1))) \neq 0$$

$$\text{address} - (90x^2 - 76x + 11) \mod (31 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 94x + 24) \mod (37 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 60x + 4) \mod (37 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 108x + 32) \mod (41 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 86x + 20) \mod (41 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 120x + 38) \mod (43 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 76x + 15) \mod (43 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 60x + 8) \mod (47 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 104x + 29) \mod (47 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 72x + 14) \mod (49 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 94x + 24) \mod (49 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 120x + 34) \mod (53 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 86x + 20) \mod (53 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 72x + 14) \mod (59 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 104x + 25) \mod (59 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 48x + 6) \mod (61 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 76x + 15) \mod (61 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 90x + 17) \mod (67 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 94x + 18) \mod (67 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 48x + 6) \mod (71 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 86x + 14) \mod (71 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 60x + 8) \mod (73 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 76x + 11) \mod (73 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 90x + 11) \mod (77 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 14x + 0) \mod (77 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 12x + 0) \mod (79 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 94x + 10) \mod (79 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 60x + 4) \mod (83 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 86x + 6) \mod (83 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 12x + 0) \mod (89 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 14x + 0) \mod (89 + (90 \cdot (x-1))) \neq 0$$

$$\text{address} - (90x^2 - 76x - 1) \mod (91 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 78x - 1) \mod (91 + (90 \cdot (x-1))) \neq 0$$

Any address which survives all 48 tests is a twin prime in A224854. To recover the base10 value: $(address * 90) + (11, 13)$

The value of x increments +1 and imposes a limit to the scope of canceled terms. The number of twin primes for the first 26 values of x for A224854 are 20, 54, 103, 167, 233, 325, 422, 522, 641, 756, 898, 1036, 1191, 1364, 1544, 1724, 1896, 2077, 2274, 2483, 2715, 2933, 3178, 3419, 3658, 3897.

For 1 to 100x average rate of change = 409.61 per step.

100 to 200 average rate of change = 982.64

200 to 300 average rate of change = 1471.15

300 to 400 average rate of change = 1932.96

400 to 500 average rate of change = 2370.97

500 to 600 average rate of change = 2794.23

600 to 700 average rate of change = 3196.37

700 to 800 average rate of change = 3602.95

800 to 900 average rate of change = 3996.97

900 to 1000 average rate of change = 4374.57

At x=1000 the number 8098920101 is the base-10 limit for the range of the sieve.

Proof of infinity: Absurdity: Assume that every value of A000027 beyond some limit x=k must be evenly divisible by a multiple of a finite pool of Conway Nontrivial Numbers. [For the complete proof, read on.]

2 Introduction

Any departure from the norms of prime number analysis must be approached with great skepticism. New sieve methods or arithmetic progressions of primes seem to emerge almost daily [see: DHJ Polymath, arxiv.org/pdf/1407.4897.pdf]. Worse, even when some new algorithm emerges from the darkness of possibility it is usually neither faster nor more efficient than currently existing alternatives [see: Dudley, U. (1983). Formulas for Primes. Mathematics Magazine, 56(1), 17–22. https://doi.org/10.1080/0025570X.1983.11977009].

The Online Encyclopedia of Integer Sequences (OEIS) contains many important entries. One of these suggests partitioning the composites into two separate classes:

A038510 Composite numbers with smallest prime factor ≥ 7.

> "John [Conway] recommends the more refined partition [of the positive numbers]: 1, prime, trivially composite, or nontrivially composite. Here, a composite number is trivially composite if it is divisible by 2, 3, or 5." See link to (van der Poorten, Thomsen, and Wiebe; 2006) pp. 73-74. - Daniel Forgues, Jan 30 2015, Feb 04 2015

This property of numbers implies the existence of a “Conway Number Universe.” It is proposed that this number universe is regulated by a “standard model of number-particle physics.”

3 What Is The Conway Number Universe?

Hint: The counting numbers (plus zero). The Conway Number Universe is just the ordinary counting numbers distributed onto a “gameboard” (aka, a ConBoard). What are some of the rules or framework(s) for constructing the Conway Number Universe?

1) “Conway numbers” are the ordinary counting numbers considered as objects or particles.

What does it mean that a Conway Number is an object or a particle?

a. ‘Pauli Exclusion’ and an ‘Impossibility Operator’ applies. For example, all number objects are individuals and no two distinct numbers can occupy the same address on a ConBoard; it is impossible for two unique number objects to possess the same order and number of digits.

b. Number objects have associated measureables/observables and states. These include certain “tangible measurables” (such as a last and leading digit) and certain properties or states (such as “primeness” or “evenness”).

c. Number objects have “degrees of certainty or uncertainty” associated with their states. For example, as regards the probability that a number particle will evaluate to prime any uncertainty regarding the state disappears once all possible factors are tested. Or, the probability a number is prime converges to 1 (certainty) as we eliminate possible factors via testing/sieving.

2) All Conway Numbers can be represented as a sequence of digits whose possible values are (0,1,2,3,4,5,6,7,8,9). For our purposes the construction of the Conway Universe and Gameboard requires the incorporation of the following two OEIS sequences:

A001477: The nonnegative integers. This sequence is included due to needing a zero indexed list to enumerate the Conway Universe.

A000027: The positive integers. Also called the natural numbers, the whole numbers or the counting numbers, but these terms are ambiguous.

3) All Conway Numbers must have a measurable associated with a digital root and it must take one of nine possible values: (1,2,3,4,5,6,7,8,9). The Conway Universe (our framework for considering the ordinary counting numbers) initially consists of 9 distinct partitions. Sloane enumerates these partitions:

a. A010888 Digital root of n (repeatedly add digits of n until a single digit is reached): [0,] 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, 5, 6, 7, 8, 9 . . .

b. Any object in A000027 is a member of one and only one of the following sequences:

Digital Root9 A008591 a(n) = 9*n.

Digital Root8 A017257 a(n) = 9*n+8.

Digital Root7 A017245 a(n) = 9*n+7.

Digital Root6 A017233 a(n) = 9*n+6.
Digital Root5 A017221 a(n) = 9*n+5.
Digital Root4 A017209 a(n) = 9*n+4.
Digital Root3 A017197 a(n) = 9*n+3.
Digital Root2 A017185 a(n) = 9*n+2.
Digital Root1 A017173 a(n) = 9*n+1.

We now assert that the Conway Universe consists of 9 classes of digital root preserving sequences. These can be represented in an nx9 list.

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
19	20	21	22	23	24	25	26	27
28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45
46	47	48	49	50	51	52	53	54
55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81
82	83	84	85	86	87	88	89	90

Figure 1: Digital Root Columns

Evidently arranging the counting numbers in an nx9 list reveals that numbers divisible by 3 exist only within digital root classes or columns 3,6,9.

1	2	3	4	5	6	7	8	9
10	11	3*4	13	14	3*5	16	17	3*6
19	20	3*7	22	23	3*8	25	26	3*9
28	29	3*10	31	32	3*11	34	35	3*12
37	38	3*13	40	41	3*14	43	44	3*15
46	47	3*16	49	50	3*17	52	53	3*18
55	56	3*19	58	59	3*20	61	62	3*21
64	65	3*22	67	68	3*23	70	71	3*24
73	74	3*25	76	77	3*26	79	80	3*27
82	83	3*28	85	86	3*29	88	89	3*30

Figure 2: Digital Root Columns 3,6,9

This is a form of "Impossibility Operator" which applies to all Conway Numbers. The statement, "There exists a prime digital root 3, 6 or 9 number larger than 3" is false due to the existence of an Impossibility Operator projected by the prime operator 3 onto the map; f.(3) maps observables to associated composite number objects and these are arrayed into columns. The columns can be thought of as part of a "family tree" or a "treasure map" depending on how we want to think about playing the Game of Primes. As a result of the 'genetic characteristics' transmitted by prime operator 3 the measurable quantity "digital root 6" indicates it is impossible for the associated number object to be prime. This impossibility operator applies to the infinity of objects that lie outside the family of digital root 3, 6 and 9 numbers. This is an instantaneous property of the continuum/column/gameboard as well as a discoverable property of an individual number object. [NOTE: There are a large number of such rules, however we will be restricting ourselves to a small subset associated with this particular configuration of the counting numbers.]

Insofar as certain characteristics or measurable properties are evidently regularly distributed we can say, "The Conway number objects snap to the grid/map based on their characteristic function(s)." That is, as we regard the Conway Numbers in an nx9 "neighborhood of objects / map of objects" we cannot unsee the patterns. Let us now unpack some of the "particle dynamics" available to number objects in the Conway Number Universe.

4 What Are Number Particle Dynamics?

Hint: Just the ordinary mathematical properties of numbers taken in unison. The crux of John Conway's conjecture is there exist two (and only two) possibilities for "composite number particles":

1) The number object is a "trivial composite particle" (divisible by 2,3,5).

2) The number object is a "non-trivial composite particle" (not divisible by 2,3,5).

The non-trivial composite numbers are resolved in OEIS as: A038510 Composite numbers with smallest prime factor ≥ 7. We also note that the trivial composite numbers include: A051037 5-smooth numbers, i.e., numbers whose prime divisors are all ≤ 5.

This implies (or actually insists) on the following:

1) There exist trivial prime (frequency) operators which define or build trivial composite objects (yield a trivial-to-produce composite certificate/class).

2) There exist non-trivial prime operators which define or build the non-trivial composite objects (yield a non-trivial composite certificate/class). NOTE: The Conway Trivial Prime Operators = [2,3,5] It is not apparent that there exists a "trivial prime" sequence in OEIS. We think of primes generally as A000040.

The rules for constructing the "Conway Trivial Composite Partition" are then as follows:

Laws of trivial composite numbers:

1.0 ALL digital root 3,6,9 numbers are composite (save 3) (test = approximately len(n) steps; solve for the digital root) *Two methods: divide by 9 and the remainder determines the class OR add the digits together until you return a single digit (in the latter method one does not need to add 9's and can drop these instantly with no loss of accuracy).*

2.0 ALL numbers ending in 0,2,4,6,8 are composite (save 2) (test == 1 step; measure the last digit)

3.0 ALL numbers ending in [0],5 are composite (save 5) (test == 1 step; measure the last digit)

1	2	3	4	5	6	7	8	9
1	2	3	co	5	co	7	co	co
co	11	co	13	co	co	co	17	co
19	co	co	co	23	co	co	co	co
co	29	co	31	co	co	co	co	co
37	co	co	co	41	co	43	co	co
co	47	co	49	co	co	co	53	co
co	co	co	co	59	co	61	co	co
co	co	co	67	co	co	co	71	co
73	co	co	co	77	co	79	co	co
co	83	co	co	co	co	co	89	co

Figure 3: Map of Conway Trivial Composite Space

TRUE: "co" is the map of composite numbers ending in 0,2,4,5,6,8 and/or having digital root 3,6,9; these are the Conway Trivial Composite numbers and take best case len(1) and worst case approximately len(n) operations to yield a composite certificate. When placed in an nx9 list one can immediately assign a composite certificate to the trivial composites.

TRUE: Evidently all Conway Numbers or number objects possess a digital root and a last digit. These are measureables and observables respectively. We see in the nx9 list that it is trivial to identify a pattern associated with the trivial composites. The regularity of the trivial composite distribution implies the relative triviality of establishing their composite certificates.

Let us add an additional column to the nx9 list, giving us an nx10 list. We now have a Conway Game of Primes Gameboard [ConBoard]). This is where we incorporate A001477 into the Conway Universe. In the leftmost column we now have an *address* which can be associated with a row of nine values; each term in A000027 exists within a row and a column. So, for numbers 1-9 we have row 0 and for 10-thru-18 we have 1, etc. We can represent this as a list: (0,(1 ... 9)), (1,(10 ... 18)), (2,(19 ... 27))... . We can determine the row and column location of any element of A000027 by dividing the number by 9. Examples:

23/9 = 2.555555... thus 23 is row 2 column 5

56/9 = 6.222222... thus 56 is row 6 column 2

99/9 = 11 In the case of multiples of 9 you get 0 as the remainder. Thus if you subtract 1 from the whole-number part of the result you will get the row address. All column 9 numbers are divisible by 9; no number divisible by 9 exists outside column 9. This is another form of impossibility operator which applies to numbers in the Conway Universe.

We can now associate a row address from A001477 with a digital root column and a last digit for a span of 9 values taken from A000027. We can now map these values onto the ConBoard.

All values in A000027 have a (row) address on the ConBoard

0	1	TP	TP	co	TP	co	7	co	co
1	co	11	co	13	co	co	co	17	co
2	19	co	co	co	23	co	co	co	co
3	co	29	co	31	co	co	co	co	co
4	37	co	co	co	41	co	43	co	co
5	co	47	co	49	co	co	co	53	co
6	co	co	co	co	59	co	61	co	co
7	co	co	co	67	co	co	co	71	co
8	73	co	co	co	77	co	79	co	co
9	co	83	co	co	co	co	co	89	co

Figure 4: Map of Conway Nontrivial Numbers

co = trivial composite

TP = trivial prime

All remaining numbers (in green) are Conway Nontrivial Numbers.

Evidently the number of prime operators required to build the Conway Trivial Composite Number partition is 3. *[Note: As an analogy think of this as a genetic lineage of numbers. A "genetic trait" is carried by the operator and this manifests as an observable in all "related" composites. Trivial composite tests (mod3=0) indicate the relative "superficiality" of the trait as it "appears" in the state representation of the number object. Approximately 1/2 of all counting numbers are divisible by 2 and this manifests as the last digits 0,2,4,6,8. We would say some number objects possess "surface characteristics" which trivialize composite certificate testing. When compared with the surface of a large semiprime number whose series of digits contains no obvious "signs of lineage" the trivial composites are obvious. It is the lack of superficial correlated observables that enforces the difficulty of semiprime factorization.]*

5 What Are Conway Nontrivial Composites?

Hint: It has something (or nothing) to do with Conway Nontrivial Primes.

TRUE: Evidently the "Conway Non-Trivial Number Universe" within a Con-Board consists of digital root 1,2,4,5,7,8 and last digit 1,3,7,9 numbers.

Conjecture: One mode for configuring the Conway Nontrivial Numbers is to assemble them as 24 classes of digital root and last digit preserving sequences. [NOTE: There are other/alternative complete descriptions for this domain of numbers]. Let us construct these classes.

The 24 classes must have zeroth elements. The 24 "smallest numbers" or "primitives" are denoted as the Conway Primitives (p):

p = (7, 11, 13, 17, 23, 29, 31, 37, 41, 43, 47, 49, 53, 57, 59, 61, 67, 71, 73, 77, 79, 83, 89, [91,1]) *[NOTE: [91,1] is a special case which relates to the operation of the 'molecular sieve' detailed later. For our immediate purposes we are treating the number 91 as the primitive element rather than the number 1. The value 1 is interchangeable with / can be replaced by the value 91 as with A181732 and the generation of A255491 and A224889 and its complement. For our immediate purposes the Conway primitive is 91.]*

We arrive at an equation for producing a digital root and last digit preserving Conway Nontrivial number sequence:

$$p + (90 * n)$$

Why 90? Necessarily all nontrivial numbers less their primitive are divisible by 90 with no remainder; if you take the multiples of 90 (ex: 90, 180, 270, ...9000, ...) and add any Conway Primitive to them, the returned number will have the same digital root and last digit as the Conway Primitive and be a member of that class of nontrivial numbers. All nontrivial numbers are built from the above relationship. Thus digital root and last digit preserving sequences can be considered a fundamental architecture for rendering Conway Nontrivial Number Space or the class of nontrivial numbers generally.

TRUE: By definition both the Conway Nontrivial Composites and the Conway Nontrivial Prime Operators (or factors of Conway Nontrivial Composites) must have digital root 1,2,4,5,7,8 and last digit 1,3,7,9 measureables and observables.

The 5-smooth numbers as a class of composites are restricted to 3 operators (2,3,5), yet there exist an infinite number of prime numbers following 2,3,5 or prime operators antecedent to 2,3,5. Thus we now assert that it takes infinity-3 Conway Nontrivial Prime frequency operators (or factors) to build a sieve for generating a list of Conway Nontrivial Composite Numbers see:A038510.

The 24 classes of Conway Nontrivial Primes (and their associated Conway Primitives) are resolved in OEIS. Sloane provides the following:

7 - A142315 Primes congruent to 7 mod 45.
11 - A142317 Primes congruent to 11 mod 45.
13 - A142318 Primes congruent to 13 mod 45.
17 - A142321 Primes congruent to 17 mod 45.
19 - A142322 Primes congruent to 19 mod 45.
23 - A142324 Primes congruent to 23 mod 45.
29 - A142327 Primes congruent to 29 mod 45.
31 - A142328 Primes congruent to 31 mod 45.
37 - A142331 Primes congruent to 37 mod 45.
41 - A142333 Primes congruent to 41 mod 45.
43 - A142334 Primes congruent to 43 mod 45.
47 - A142313 Primes congruent to 2 mod 45. -excluding 2, the zeroth element
49 - A142314 Primes congruent to 4 mod 45.
53 - A142316 Primes congruent to 8 mod 45.
59 - A142319 Primes congruent to 14 mod 45.
61 - A142320 Primes congruent to 16 mod 45.
67 - A142323 Primes congruent to 22 mod 45.
71 - A142325 Primes congruent to 26 mod 45.
73 - A142326 Primes congruent to 28 mod 45.
77 - A142329 Primes congruent to 32 mod 45.
79 - A142330 Primes congruent to 34 mod 45.
83 - A142332 Primes congruent to 38 mod 45.

89 - A142335 Primes congruent to 44 mod 45.

[1,91] - A142312 Primes congruent to 1 mod 45.

As evidently only primes of a non-trivial type can sieve for nontrivial composites or be factors of nontrivial composites we propose to build a Conway Nontrivial Sieve from Conway Nontrivial Numbers.

6 Playing The Game Of Primes

Hint: The Conway Nontrivial Sieve is of two types, non-self-referential and self-referential.

6.1 The Non-Self-Referential Conway Sieve

...in quasi-pseudocode:

We must declare our constants (Conway Primitives):

p = (7,11,13,17,23,29,31,37,41,43,47,49,53,57,59,61,67,71,73,77,79,83,89,91)

We must declare a limit.

test = int(input(Your number here))

We must divide the limit by 90. The generator is multiples of 90.

limit = int(test/90)

We must generate an empty list.

ConSieve = [ ]

We must use the limit to define some number of operations for our sieve.

for x in range(0, limit):

We must generate a Conway Nontrivial Number.

y = p+(90*(x))

We must append the Conway Nontrivial Number to the list.

ConSieve.extend(y)

As with the Sieve of Eratosthenes, we now want to add multiples of our number to the list. However, we only want to map the multiples that correspond to the Conway Nontrivial Numbers. Thus we do not add numbers whose factors include 2,3,5 by implementing the $n \mod 5$ and $n \mod 3$ tests below. Python lacks the ability to use a pattern value within range as with the initial line:

for n in range (y,int((test/y, [4,2,4,2,4,6,2,6])

example: 7, 11, 13, 17, 19, 23, 29, 31, 37, repeat

for n in range (y, int((test/y)), 2):

example: 7, (test/7), +2) output=7,9,11,13...

if $n \mod 5 \neq 0:$

if $n \mod 3 \neq 0:$

ConSieve.extend(y*n)

We thus:

for x in range(1, limit+1):

ConSeq(p)

We repeat this process until the limit is reached. OUTPUT: Any number appearing exactly one time (frequency=1 OR amplitude=1) in the output (list) is prime. Note: In the spirit of John Conway's PRIMEGAME this sieve is admittedly inefficient. It is easy to implement this in such a way you have a tail of false-positives. Make sure all operators print to the actual limit.

6.2 Self-Referential Conway Sieve (Modified Sieve of Eratosthenes)

We must declare our frequency operators (Conway Primitives):

p = (7,11,13,17,23,29,31,37,41,43,47,49,53,57,59,61,67,71,73,77,79,83,89,91)

We must create 24 empty lists for aggregating the 24 classes of Conway Nontrivial Primes:

primelist = (A142315, A142317, A142318, A142321, A142322, A142324, A142327, A142328, A142331, A142333, A142334, A142313, A142314, A142316, A142319, A142320, A142323, A142325, A142326, A142329, A142330, A142332, A142335, A142312, A142315, A142317, A142318, A142321, A142322, A142324, A142327, A142328, A142331, A142333, A142334, A142313, A142314, A142316, A142319, A142320, A142323, A142325, A142326, A142329, A142330, A142332, A142335, A142312) as A142315 = [ ], A142317 = [ ], ...

We must declare a test value for our Base-10 range.

test = int(input(Your number here))

We must calculate a limit for the number of iterations of our function(s).

limit = int(test/90)

We must provide a ConBoard for storing the outputs from our frequency operators. We do this by generating a list of 0's whose quantity of elements is equal to the test number.

addressspace = [0]*int(test)

We now populate the values for our function.

for x in range(0, limit):

We generate a term (y) from a Conway Primitive.

$y = p + (90 * (x))$

In the Style of the Sieve of Eratosthenes we now check the addressspace list to see if this y address contains a value > 0.

if addressspace[y] > 0:

return

If the address is occupied we return and do not continue to process further steps; the number is not a prime frequency operator. The only alternative is that the value equals zero, then:

We record the value of y into the addressspace.

addressspace[y] = y

We record the value of y into the associated Axxxxxx Prime Operator addresslist.

primelist.append(y)

We now distribute the Conway Nontrivial Composites via frequency generation using modular arithmetic tests as detailed previously.

for n in range (y, int((test/y)), 2):

example: 7, (test/7), +2) output=7,9,11,13...
if $n \mod 5 \neq 0$:
if $n \mod 3 \neq 0$:
newy = y*(n)
addressspace[newy] = addressspace[newy] + 1 (+1 in amplitude)
We now populate the values for our function.
for x in range(1, limit+1):
ConSeq(p)

In Method 1 all nontrivial numbers whose amplitude (number of occurrences) is measured as equal to 1 are prime. By design all Conway Nontrivial Numbers are used as frequency operators to produce a map of amplitudes. We must read (or sort) the list and look for numbers that appear only once. This sort is expensive. This sieve necessarily cannot produce 5-smooth numbers. In fact, by removing the modulus tests in the above algorithms the 0's in the list return the Hamming Numbers or 5-smooth number sequence A051037. So, in effect, this is a Hamming Number sieve masquerading as a prime number sieve.

TRUE: Multiples of the Conway Nontrivial Numbers cannot produce a Hamming Number.

Conjecture: By adding a constant to the members of the Conway Primitives you create a skewed class of Hamming-like Numbers.

7 Digital Root And Last Digit Preserving Composite Sequences

Hint: There are certain things whose number is unknown. If we count them by threes, we have two left over; by fives, we have three left over; and by sevens, two are left over. How many things are there? –Sunzi Suanjing

Both Non-Self-Referential and Self-Referential sieves use "number atoms" or single number objects to generate "cancellation frequencies / signals" which then distribute composite certificates to some limit. The number of times a number appears (its frequency in the list) determines its prime or composite state. The total number of factors for a composite is called Big Omega, which will be discussed in Section 10 Results.

TRUE BY CONSTRUCTION: Evidently a digital root and last digit preserving sequence of numbers can be ordered as a zero-indexed list.

TRUE BY DEFINITION: Evidently a digital root and last digit preserving sequence of nontrivial numbers can be placed into a 1:1 correspondence with A001477 (0,1,2,3...) the non-negative numbers. OEIS resolves this address schema for the Conway Nontrivial Primes.

From A000040 - The Prime Numbers:

> Reading the primes (excluding 2,3,5) mod 90 divides them into 24 classes, which are described by A181732, A195993, A198382, A196000, A201804, A196007, A201734, A201739, A201819, A201816, A201817, A201818, A202104,

A201820, A201822, A201101, A202113, A202105, A202110, A202112, A202129, A202114, A202115 and A202116. J. W. Helkenberg, Jul 24 2013

Why mod 90? Per A201804 this is an application of the Chinese Remainder Theorem. Division by 90 removes or strips the digital root measurable and last digit observable from a Conway Nontrivial Number and thereby generates its corresponding "class-address." The reduced number remains divisible by its smallest factor, as will be demonstrated. We know from earlier that ALL nontrivial composites and primes are built from (90*n)+p. Thus we know that the list of primes A142317 correlates to a list of n's in A201804:

7	97	187	277	367	457	547	637	727
0	1	2	3	4	5	6	7	8

Figure 5: 0-indexed list of nontrivial numbers

A142317 Primes congruent to 11 mod 45.

A201804 Numbers n such that 90*n + 11 is prime.

The above sequences can be placed into 1:1 correspondence.

Conjecture: There exist 24 permissible digital root and last digit preserving multiplicative arrangements for the 24 Conway Primitives. These multiplicative arrangements or "bindings" can be thought of as generating "number molecules" from a collection of "number atoms." For example, in the Conway non-self-referential sieve when we describe the Conway Primitives operating as "single atoms" we mean that the atom 7 generates a composite print statement "signal" that maps a composite certificate to 1/7th of the remaining counting numbers; mod7=0 is a trait which can be revealed if division of that number by 7 leaves no remainder. We then sort the contributions of the various "atoms" (Conway nontrivial numbers) and look for objects that exist with frequency=1. For the purpose of rendering a composite signal against or upon the static space of A000027 we say that the Conway non-self-referential sieve uses only Conway Nontrivial Numbers as operators/generators.

8 The Molecular Sieve

Hint: Bohr Model of Atomic Orbitals meets Minkowski Space In the Molecular Sieve for A201804 the 24 Conway Primitives collapse (or fold) into "legal pairs" which become the new primitives for generating digital root and last digit preserving composite number outputs. This necessitates that every nontrivial composite number exists as a multiplied pair of Conway nontrivial numbers. While a given number may have numerous factors (Big Omega and Small Omega) a molecular sieve always determines two factors simultaneously.

Conjecture: For A201804 all non-trivial composite numbers are located within the solutions to 12 Diophantine quadratic sequences.

For example, the "binding configuration" for a digital root 2 last digit 1 sequence of composites requires the underlying primitives/factors to be configured as follows: $\neg A142317$ = (7,53),(19,29),(17,43),(13,77),(11,91),(31,41), (23,67),(49,59),(37,83),(47,73),(61,71),(79,89) [Note: 7+(90*n) * 53+(90*k) for n,k=A001477 is approximately 1/12th of $\neg A142317$] When factors are bound into discrete arrangements they produce composite numbers bound to a specific class.

The functions which generate digital root and last digit preserving composite sequences must themselves conserve the last digit and digital root of the cancellation operators. This imposes an impossibility condition on potential configurations of prime factors; the insertion location for all nontrivial-number cancellation operators is then quantized. We now present two implementations.

8.1 Modular Ordinary Sieve

For A201804 there are 12 unique quadratic sequences each having two Conway Nontrivial primitives attached as cancellation operators. In a standard 'cancellation' method the whole number values generated by the quadratic sequence are the 'insertion locations' for the Conway nontrivial cancellation operators. This algorithm is essentially stating that by subtracting the quadratic value from the tested number (address of the base-10 number) it becomes divisible by its smallest Conway nontrivial factor.

The prime address generating function is built from the following 24 inequalities. For all Digital Root 2 Last Digit 1 (A201804 = DR2LD1) prime number addresses the following must be true:

The chibi-styled algorithm:

Address takes all values A001477, the non-negative integers 0,1,2,3... .

x takes all values of A000027, the positive counting numbers 1,2,3,4. . . .

WHERE $address - y = 0 = Composite$

ELSE

WHERE $address - y > 0$

WHERE $address - y \mod p \neq 0$

THEN address is a prime value in A201804.

To produce the members of A201804 operate the sieve until $y \geq address$:

$$\text{address} - y \mod p \neq 0 \implies \text{Prime}(True)$$

$$\text{address} - (90x^2 - 120x + 34) \mod (7 + (90 \cdot (x-1))) \neq 0$$

$$\text{address} - (90x^2 - 78x - 1) \mod (11 + (90 \cdot (x-1))) \neq 0$$

$$\text{address} - (90x^2 - 90x + 11) \mod (13 + (90 \cdot (x-1))) \neq 0$$

$$\text{address} - (90x^2 - 120x + 38) \mod (17 + (90 \cdot (x-1))) \neq 0$$

$$\text{address} - (90x^2 - 132x + 48) \mod (19 + (90 \cdot (x-1))) \neq 0$$

$$\text{address} - (90x^2 - 90x + 17) \mod (23 + (90 \cdot (x-1))) \neq 0$$

$$\text{address} - (90x^2 - 132x + 48) \mod (29 + (90 \cdot (x-1))) \neq 0$$

$$\text{address} - (90x^2 - 108x + 32) \mod (31 + (90 \cdot (x-1))) \neq 0$$

$$\text{address} - (90x^2 - 60x + 4) \mod (37 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 108x + 32) \mod (41 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 120x + 38) \mod (43 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 60x + 8) \mod (47 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 72x + 14) \mod (49 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 120x + 34) \mod (53 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 72x + 14) \mod (59 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 48x + 6) \mod (61 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 90x + 17) \mod (67 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 48x + 6) \mod (71 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 60x + 8) \mod (73 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 90x + 11) \mod (77 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 12x + 0) \mod (79 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 60x + 4) \mod (83 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 12x + 0) \mod (89 + (90 \cdot (x-1))) \neq 0$$
$$\text{address} - (90x^2 - 78x - 1) \mod (91 + (90 \cdot (x-1))) \neq 0$$

For any address which returns TRUE for all 24 evaluations the equation (90*address)+11 returns a Digital Root 2 Last Digit 1 prime number.

The number of failed tests determines an amplitude. The amplitude determines Big Omega. For any address which fails one test the number is a semiprime of the form a*b [Note: An associated Impossibility Operator requires that A201804 contain no squared semiprimes. This is related to the global ruleset for impossibility operators which is beyond the scope of this writing.] We will address Big Omega in Section 10 Results.

The alternative to the above is to reverse the operations such that you generate y and then implement a cancellation operator for all multiples of all Conway Nontrivial numbers, similar to the original Conway "Game of Primes" algorithm. In the current implementation it is best to think of the growth of the quadratic sequences as representing "epochs" of addresses. There is a "maximum extent" of sieving for x=1 which correlates with the number of terms that can be sieved "per round of cancellation." The size of an epoch is approximately $90x^2 - 12x + 1$. Meanwhile the number of new "Conway nontrivial cancellation operators" for A201804 at the limit is 24*x or 12*x if you are only counting the unique quadratic sequences.

The epoch grows faster than the number of Conway nontrivial numbers used per round to sieve. Additionally, per the Prime Number Theorem as the range of addresses grows larger we encounter fewer and fewer prime number cancellation operators. The Conway Nontrivial numbers that are themselves composite add amplitude to already “canceled” addresses, but do not deduct addresses from the range. This further guarantees an infinity of primes, as there are insufficient prime-valued generators per epoch to cancel all elements of A001477. So, A001477 contains 100 percent of the addresses associated with a given class of nontrivial numbers and the composite addresses become divisible by the smallest factor following the subtraction of y. The epoch model is important for diagnosing the relative growth of prime numbers as well as twin primes, cousin primes and other arrangements.

8.2 Traditional Molecular Sieve

The second version of this algorithm is presented as Python code.

```
#!/usr/bin/env python
import cmath

limit = int(input('limit value here:')) # the "epoch"
limit = int(limit)  #convert it to an int type
h = limit    #set variable h as equivalent to "limit"
epoch = 90*(h*h) - 12*h + 1 #The limit of the epoch
print('The epoch range is', epoch)
limit = epoch
base10 = (limit*90)+11
print('This is the base-10 limit:', base10)
#get number of iterations x for quadratic functions
a = 90
b = -300
c = 250 - limit
# calculate the discriminant
d = (b**2) - (4*a*c)
# find two solutions
sol1 = (-b-cmath.sqrt(d))/(2*a)
sol2 = (-b+cmath.sqrt(d))/(2*a)
print('The solution are {0} and {1}'.format(sol1,sol2))
new_limit = sol2 #integer REAL part for RANGE
A201804 = [0]*int(limit+10) #pad we will dropat the end

#x=increment, l=quadratic term, m = quadratic term,
z = primitive, o = primitive
def drLD(x, l, m, z, o, listvar, primitive):
"This is a composite generating function"
  y = 90*(x*x) - l*x + m
  listvar[y] = listvar[y]+1
  p = z+(90*(x-1))
  q = o+(90*(x-1))
```

```
    for n in range (1, int(((limit-y)/p)+1)):
      listvar[y+(p*n)] = listvar[y+(p*n)]+1
    for n in range (1, int(((limit-y)/q)+1)):
      listvar[y+(q*n)] = listvar[y+(q*n)]+1

for x in range(1, int(new_limit.real)):
#A201804
    drLD(x, 120, 34, 7, 53, A201804, 11)  #7,53 @4, 154
    drLD(x, 132, 48, 19, 29, A201804, 11) #19,29 @6, 144
    drLD(x, 120, 38, 17, 43, A201804, 11) #17,43 @8, 158
    drLD(x, 90, 11, 13, 77, A201804, 11)  #13,77 @11, 191
    drLD(x, 78, -1, 11, 91, A201804, 11)  #11,91 @11, 203
    drLD(x, 108, 32, 31, 41, A201804, 11) #31,41 @14, 176
    drLD(x, 90, 17, 23, 67, A201804, 11)  #23,67 @17, 197
    drLD(x, 72, 14, 49, 59, A201804, 11)  #49,59 @32, 230
    drLD(x, 60, 4, 37, 83, A201804, 11)   #37,83 @34, 244
    drLD(x, 60, 8, 47, 73, A201804, 11)   #47,73 @38, 248
    drLD(x, 48, 6, 61, 71, A201804, 11)   #61,71 @48, 270
    drLD(x, 12, 0, 79, 89, A201804, 11)   #79,89 @78, 336

A201804 = A201804[:-10] #remove the padding
print('Count', A201804.count(0), 'primes at limit', len(A201804))
A201804_enumerated=[i for i,x in enumerate(A201804) if x == 0]
print('This is A201804:', A201804_enumerated)
```

For A201804 as x increments +1:
Epoch increments $90x^2 - 12x + 1$
New addresses added per epoch = $180x + 78$
Unique Quadratics = $12 * x$
Frequency operators = $24 * x$
base-10 limit = $8100x^2 - 1080x + 101$

The possible factors (operators) for numbers of a given length is determined by a step function based on the number of epochs in a given length of number. Up to 4 digit addresses can be sieved by one pass through the 12 generators. It take 2 passes to sieve 5-digit space, 8 for 6 digit. The pattern:

there exist 1 4-digit value
there exist 2 5-digit values
there exist 8 6-digit values
there exist 24 7-digit values
there exist 76 8-digit values
there exist 242 9-digit values
there exist 764 10-digit values
there exist 2416 11-digit values

This indicates that the number of possible factors is increasing as we traverse the n-digit addresses. The amplitude relative to the leading digit is beyond the scope of this paper. Anecdotally when going from 50000 to 60000 or 500000 to 600000 in address space there is chance for leading digit 5 partition to have more amplitude than the leading 6 digit partition; the higher the amplitude in a given finite span the greater the observed number of primes.

9 The Twin Prime Sieve

Hint: Unfortunately, one-parameter patterns, such as twins n, n+2, remain stubbornly beyond current technology. There is still much to be done in the subject! –T. Tao

The twin primes are equivalent to 9 classes:

A224855 Numbers n such that 90*n + 17 and 90*n + 19 are twin primes.

A224856 Numbers n such that 90*n + 29 and 90*n + 31 are twin primes.

A224859 Numbers n such that 90*n + 47 and 90*n + 49 are twin primes.

A224854 Numbers n such that 90*n + 11 and 90*n + 13 are twin prime.

A224857 Numbers n such that 90n + 41 and 90n + 43 are twin primes.

A224860 Numbers n such that 90*n + 59 and 90*n + 61 are twin prime.

A224862 Numbers n such that 90*n + 71 and 90*n + 73 are twin primes.

A224864 Numbers n such that 90*n + 77 and 90*n + 79 are twin primes.

A224865 Numbers n such that 90*n + 89 and 90*n + 91 are twin primes.

A sieve for A224854 was detailed in Section 1. When taking all 9 sequences into account we want to analyze the total growth of twin primes per epoch. We find the following:

a(n) to a(n+1) = a(n+1)/a(n) = ratio of terms

100x to 200x = 3.422 (there are 3.422 times as many twin primes from 100 to 200 as there are from 0 to 100)

200 to 300 = 2.065

300 to 400 = 1.676

400 to 500 = 1.495

500 to 600 = 1.389

600 to 700 = 1.321

700 to 800 = 1.273

800 to 900 = 1.238

900 to 1000 = 1.211

1000 to 1100 = 1.189

1100 to 1200 = 1.171

1200 to 1300 = 1.157

1300 to 1400 = 1.144

1400 to 1500 = 1.134

1500 to 1600 = 1.125

1600 to 1700 = 1.117

1700 to 1800 = 1.110

1800 to 1900 = 1.104

1900 to 2000 = 1.098

2000 to 2100 = 1.093

2100 to 2200 = 1.089

2200 to 2300 = 1.085
2300 to 2400 = 1.081
2400 to 2500 = 1.078
2500 to 2600 = 1.074
2600 to 2700 = 1.072
2700 to 2800 = 1.070
2800 to 2900 = 1.066
2900 to 3000 = 1.064
3000 to 3100 = 1.062
3100 to 3200 – 1.060
3200 to 3300 = 1.058
3300 to 3400 = 1.056
3400 to 3500 = 1.055 or (222771992 / 211237541)
Conjecture: a(n+1)/a(n) is always greater than 1.

If we can prove that A224854 contains an infinite number of terms we have proven the Twin Prime Conjecture. As we examine the cancellation operators associated with $\neg A224854$ we can immediately recognize that it is functionally impossible to eliminate all counting numbers beyond a given limit. This is reinforced by the observed reality that as the epoch grows so does the number of twin primes.

Let's assume that the generators in A224854 are capable of eliminating all counting numbers beyond some limit and show how this leads to a contradiction.

A201804(11): y=4(7,53), 154(97,143), 484(187,233)
A201816(13): y=6(7,79), 182(97,169), 538,(187,259)
Starting at the value 4 exactly 1 in every 7 numbers is composite.
Starting at the value 6 exactly 1 in every 7 numbers is composite
4,11,18,25,32,39,46,53,60,67,74,81,88,95, ... are composite
6,13,20,27,34,41,48,55,62,69,76,83,90,97, ... are composite

The distribution of these cancellation operators is insufficient to provide coverage against all values in A001477; there can be no value beyond which all numbers are generated by the available remaining Conway Nontrivial numbers / operators. This is reinforced by the Prime Number Theorem, which indicates that the number of prime number operators on average decreases as the numbers get larger. As we know we are dealing with 48 cancellation operators per increment of x and that the range of numbers to be sequenced grows by approximately 180x+78 we can see that the number of prime cancellation operators must necessarily fail to cover all the addresses from epoch to epoch. The proof relies on stating there is a number such that all numbers beyond it are solutions in finite x. Since x grows to infinity and new prime numbers MUST exist for A201804 and A201816 the union set of these two classes cannot then be finite as new (semiprime) composites must necessarily be generated by new prime operators.

Take the number 7. The next operator available to perform cancellations is 97. The generator is insufficient to fill the spaces missed by 7. The next operator is 187, then 277, etc. The decay in the frequency coupled with the

Prime Number Theorem ensures an infinity of twin primes exist. In fact, this is true for ANY pair of the 24 classes of sequences when taking the union set.

To prove the above assertions we show that there must exist a configuration of operators that is sufficient to cancel all values. An example is now given using the operator 7:

start address=0, operator = 7
start address=1, operator = 7
start address=2, operator = 7
start address=3, operator = 7
start address=4, operator = 7
start address=5, operator = 7
start address=6, operator = 7

we then have the following cancellations:

0, 7, 14, 21, 28, ...
1, 8, 15, 22, 29, ...
2, 9, 16, 23, 30, ...
3, 10, 17, 24, 31, ...
4, 11, 18, 25, 32, ...
5, 12, 19, 26, 33, ...
6, 13, 20, 27, 34, ...

In this scenario, if we have seven sequences whose Conway nontrivial 7 has starting positions ranging from 0 to 6 then the sequence contains no surviving elements. The cancellation operators that produce A224854 lack this configuration and therefore lack sufficient density to populate all terms in A001477 beyond a limit.

Is there a combination of OEIS sequences which guarantees that there are no survivors as per the description above? We can take the following:

0 = 49 (A201818) test
0 = 77 (A201822)
1 = 71 (A202129) test
1 = 43 (A202105)
1 = 29 (A201739)
2 = 79 (A202112) test
2 = 37 (A198382)
2 = 23 (A201820)
3 = 17 (A202115) test
3 = 73 (A195993)
3 = 59 (A202101)
3 = 31 (A201819)
4 = 11 (A201804)
4 = 67 (A201817) test
4 = 53 (A202114)
5 = 89 (A202116) test
5 = 61 (A202113)
5 = 47 (A201734)
5 = 19 (A196000)

6 = 13 (A201816) test

6 = 41 (A202104)

Any combination (union set) of the above sequences will suffice to eliminate all values of A001477 at infinity provided it contains a list of start positions (0,1,2,3,4,5,6). The union set labeled “test” necessarily cancels all terms.

Operator 11 will also cancel 100 percent of the terms in A001477 with the union set of start positions (0,1,2,3,4,5,6,7,8,9,10) using any combination of the following:

0 = 77 (A201822)

1 = 53 (A202114)

1 = 31 (A201819)

2 = 73 (A195993)

2 = 7 (A202110)

2 = 29 (A201739)

3 = 71 (A202129)

3 = 49 (A201818)

4 = 47 (A201734)

5 = 89 (A202116)

5 = 23 (A201820)

5 = 67 (A201817)

6 = 43 (A202105)

7 = 41 (A202104)

7 = 19 (A196000)

8 = 17 (A202115)

8 = 83 (A196007)

8 = 61 (A202113)

9 = 59 (A202101)

9 = 37 (A198382)

10 = 79 (A202112)

10 = 13 (A201816)

Additionally, start position union set of (1,2,3,4,5,6,7,8,9,10,11,12,13) for operator 13 and start position union set of (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17) for operator 17 will cancel all terms of A000027.

For A201804 we have the following addresses with the associated Conway primitive cancellation operators (p,q):

y=4, 154, 484, ... (7, 53)

y=6, 144, 462, ... (19,29)

y=8, 158, 488, ... (17, 43)

y=11, 191, 151, ... (13, 77)

y=11, 203, 575, ... (11, 91)

y=14, 176, 518, ... (31, 41)

y=17, 197, 557, ... (23, 67)

y=32, 230, 608, ... (49, 59)

y=34, 244, 634, ... (37, 83)

y=38, 248, 638, ... (47, 73)

y=48, 270, 672, ... (61, 71)

y=78, 336, 774, ... (79, 89)
For A201816:
y=6, 182, 538, ... (7,79)
y=10, 194, 558, ... (11, 83)
y=11, 177, 523, ... (17, 59)
y=13, 207, 581, ... (13,91)
y=14, 190, 546, ... (19, 67)
y=15, 181, 527, ... (29, 47)
y=18, 202, 566, ... (23, 71)
y=20, 196, 552, ... (37, 49)
y=24, 208, 572, ... (41, 53)
y=25, 219, 593, ... (31, 73)
y=29, 223, 597, ... (43, 61)
y=76, 332, 768, ... (77, 89)

Assuming the union set of these two classes of operators is sufficient to cancel all terms of A001477 beyond some point fails on its face; it is a superficial characteristic of these cancellation operators that they are incapable of generating sufficient density. Additionally, both sub-sequences A201804 and A201816 necessarily contain an infinite number of terms and use the same cancellation operators per epoch (with different start positions) and thus the probability that both infinite sequences contain no matching terms when evaluated from the perspective of the known cancellation operators evaluates to zero. This proves the infinity of the twin primes.

The global Twin Prime Sieve is then as follows:

```
#!/usr/bin/env python
import cmath #for a limit calculation

#get a value for the limit of the range to be sieved
limit = int(input("limit value here:"))
limit = int(limit) #convert it to an int type

#epoch
h = limit
epoch = 90*(h*h) - 12*h + 1
print("The epoch range is", epoch)
limit = epoch

#get RANGE for quadratic functions
a = 90
b = -300
c = 250 - limit
# calculate the discriminant
d = (b**2) - (4*a*c)
# find two solutions
```

```
sol1 = (-b-cmath.sqrt(d))/(2*a)
sol2 = (-b+cmath.sqrt(d))/(2*a)
print('The solution are {0} and {1}'.format(sol1,sol2))

new_limit = sol2

A224854 = [0]*int(limit)
A224855 = [0]*int(limit)
A224856 = [0]*int(limit)
A224857 = [0]*int(limit)
A224859 = [0]*int(limit)
A224860 = [0]*int(limit)
A224862 = [0]*int(limit)
A224864 = [0]*int(limit)
A224865 = [0]*int(limit)

def drLD(x, l, m, z, o, listvar, primitive):
  "This is a composite generating function"
  y = 90*(x*x) - l*x + m
  try:
    listvar[y] = listvar[y]+1
  except:
    pass
  p = z+(90*(x-1))
  q = o+(90*(x-1))
  for n in range (1, int(((limit-y)/p)+1)):
    try:
      listvar[y+(p*n)] = listvar[y+(p*n)]+1
    except:
      pass
  for n in range (1, int(((limit-y)/q)+1)):
    try:
      listvar[y+(q*n)] = listvar[y+(q*n)]+1
    except:
      pass

for x in range(1, int(new_limit.real)):

# 19 = A196000
    drLD(x, 70, -1, 19, 91, A224855, 19) #19,91
    drLD(x, 106, 31, 37, 37, A224855, 19) #37,73
    drLD(x, 34, 3, 73, 73, A224855, 19) #73,73
    drLD(x, 110, 27, 11, 59, A224855, 19) #11,59
    drLD(x, 110, 33, 29, 41, A224855, 19) #29,41
    drLD(x, 56, 6, 47, 77, A224855, 19) #47,77
    drLD(x, 74, 5, 23, 83, A224855, 19) #23,83
```

```
    drLD(x, 124, 40, 13, 43, A224855, 19) #13,43
    drLD(x, 70, 7, 31, 79, A224855, 19) #31,79
    drLD(x, 70, 13, 49, 61, A224855, 19) #49,61
    drLD(x, 106, 21, 7, 67, A224855, 19) #7,67
    drLD(x, 20, 0, 71, 89, A224855, 19) #71,89
    drLD(x, 74, 15, 53, 53, A224855, 19) #53,53
    drLD(x, 146, 59, 17, 17, A224855, 19) #17,17

#17 = A202115
    drLD(x, 72, -1, 17, 91, A224855, 17)   #17,91
    drLD(x, 108, 29, 19, 53, A224855, 17) #19,53
    drLD(x, 72, 11, 37, 71, A224855, 17)   #37,71
    drLD(x, 18, 0, 73, 89, A224855, 17)   #73,89
    drLD(x, 102, 20, 11, 67, A224855, 17)  #11,67
    drLD(x, 138, 52, 13, 29, A224855, 17) #13,29
    drLD(x, 102, 28, 31, 47, A224855, 17)  #31,47
    drLD(x, 48, 3, 49, 83, A224855, 17)  #49,83
    drLD(x, 78, 8, 23, 79, A224855, 17)  #23,79
    drLD(x, 132, 45, 7, 41, A224855, 17) #7,41
    drLD(x, 78, 16, 43, 59, A224855, 17)   #43,59
    drLD(x, 42, 4, 61, 77, A224855, 17) #61,77

#91 = A224889
    drLD(x, -2, -1, 91, 91, A224865, 91) #91,91
    drLD(x, 142, 55, 19, 19, A224865, 91) #19,19
    drLD(x, 70, 9, 37, 73, A224865, 91) #37, 73
    drLD(x, 128, 42, 11, 41, A224865, 91) #11, 41
    drLD(x, 92, 20, 29, 59, A224865, 91) #29,59
    drLD(x, 110, 31, 23, 47, A224865, 91) #23,47
    drLD(x, 20, 0, 77, 83, A224865, 91) #77,83
    drLD(x, 160, 70, 7, 13, A224865, 91) #7,13
    drLD(x, 88, 18, 31, 61, A224865, 91) #31,61
    drLD(x, 52, 4, 49, 79, A224865, 91) #49,79
    drLD(x, 70, 11, 43, 67, A224865, 91) #43,67
    drLD(x, 110, 29, 17, 53, A224865, 91) #17,53
    drLD(x, 38, 3, 71, 71, A224865, 91) #71,71
    drLD(x, 2, -1, 89, 89, A224865, 91) #89,89

# 89 = A202116
    drLD(x, 0, -1, 89, 91, A224865, 89) #89,91
    drLD(x, 90, 14, 19, 71, A224865, 89) #19,71
    drLD(x, 126, 42, 17, 37, A224865, 89) #17,37
    drLD(x, 54, 6, 53, 73, A224865, 89) #53,73
    drLD(x, 120, 35, 11, 49, A224865, 89) #11,49
    drLD(x, 120, 39, 29, 31, A224865, 89) #29,31
    drLD(x, 66, 10, 47, 67, A224865, 89) #47,67
```

```
    drLD(x, 84, 5, 13, 83, A224865, 89) #13,83
    drLD(x, 114, 34, 23, 43, A224865, 89) #23,43
    drLD(x, 60, 5, 41, 79, A224865, 89) #41,79
    drLD(x, 60, 9, 59, 61, A224865, 89) #59,61
    drLD(x, 96, 11, 7, 77, A224865, 89) #7,77

# 31 = A201819
    drLD(x, 58, −1, 31, 91, A224856, 31) #31,91
    drLD(x, 112, 32, 19, 49, A224856, 31) #19,49
    drLD(x, 130, 45, 13, 37, A224856, 31) #13,37
    drLD(x, 40, 4, 67, 73, A224856, 31) #67,73
    drLD(x, 158, 69, 11, 11, A224856, 31) #11,11
    drLD(x, 122, 41, 29, 29, A224856, 31) #29,29
    drLD(x, 50, 3, 47, 83, A224856, 31) #47,83
    drLD(x, 140, 54, 17, 23, A224856, 31) #17,23
    drLD(x, 68, 10, 41, 71, A224856, 31) #41,71
    drLD(x, 32, 0, 59, 89, A224856, 31) #59,89
    drLD(x, 50, 5, 53, 77, A224856, 31) #53,77
    drLD(x, 130, 43, 7, 43, A224856, 31) #7,43
    drLD(x, 58, 9, 61, 61, A224856, 31) #61,61
    drLD(x, 22, 1, 79, 79, A224856, 31) #79,79

# 29 = A201739
    drLD(x, 60, −1, 29, 91, A224856, 29) #29,91
    drLD(x, 150, 62, 11, 19, A224856, 29) #11,19
    drLD(x, 96, 25, 37, 47, A224856, 29) #37,47
    drLD(x, 24, 1, 73, 83, A224856, 29) #73,83
    drLD(x, 144, 57, 13, 23, A224856, 29) #13,23
    drLD(x, 90, 20, 31, 59, A224856, 29) #31,59
    drLD(x, 90, 22, 41, 49, A224856, 29) #41,49
    drLD(x, 36, 3, 67, 77, A224856, 29) #67,77
    drLD(x, 156, 67, 7, 17, A224856, 29) #7,17
    drLD(x, 84, 19, 43, 53, A224856, 29) #43,53
    drLD(x, 30, 0, 61, 89, A224856, 29) #61,89
    drLD(x, 30, 2, 71, 79, A224856, 29) #71,79

# 49 = A201818
    drLD(x, 40, −1, 49, 91, A224859, 49) #49,91
    drLD(x, 130, 46, 19, 31, A224859, 49) #19,31
    drLD(x, 76, 13, 37, 67, A224859, 49) #37,67
    drLD(x, 94, 14, 13, 73, A224859, 49) #13,73
    drLD(x, 140, 53, 11, 29, A224859, 49) #11,29
    drLD(x, 86, 20, 47, 47, A224859, 49) #47,47
    drLD(x, 14, 0, 83, 83, A224859, 49) #83,83
    drLD(x, 104, 27, 23, 53, A224859, 49) #23,53
    drLD(x, 50, 0, 41, 89, A224859, 49) #41,89
```

```
    drLD(x, 50, 6, 59, 71, A224859, 49) #59,71
    drLD(x, 86, 10, 17, 77, A224859, 49) #17,77
    drLD(x, 166, 76, 7, 7, A224859, 49) #7,7
    drLD(x, 94, 24, 43, 43, A224859, 49) #43,43
    drLD(x, 40, 3, 61, 79, A224859, 49) #61,79

# 47 = A201734
    drLD(x, 42, -1, 47, 91, A224859, 47) #47,91
    drLD(x, 78, 5, 19, 83, A224859, 47) #19,83
    drLD(x, 132, 46, 11, 37, A224859, 47) #11,37
    drLD(x, 78, 11, 29, 73, A224859, 47) #29,73
    drLD(x, 108, 26, 13, 59, A224859, 47) #13,59
    drLD(x, 72, 8, 31, 77, A224859, 47) #31,77
    drLD(x, 108, 30, 23, 49, A224859, 47) #23,49
    drLD(x, 102, 17, 7, 71, A224859, 47) #7,71
    drLD(x, 48, 0, 43, 89, A224859, 47) #43,89
    drLD(x, 102, 23, 17, 61, A224859, 47) #17,61
    drLD(x, 48, 4, 53, 79, A224859, 47) #53,79
    drLD(x, 72, 12, 41, 67, A224859, 47) #41,67

# 11 = A201804
    drLD(x, 120, 34, 7, 53, A224854, 11)  #7,53
    drLD(x, 132, 48, 19, 29, A224854, 11) #19,29
    drLD(x, 120, 38, 17, 43, A224854, 11) #17,43
    drLD(x, 90, 11, 13, 77, A224854, 11)  #13,77
    drLD(x, 78, -1, 11, 91, A224854, 11)  #11,91
    drLD(x, 108, 32, 31, 41, A224854, 11) #31,41
    drLD(x, 90, 17, 23, 67, A224854, 11)  #23,67
    drLD(x, 72, 14, 49, 59, A224854, 11)  #49,59
    drLD(x, 60, 4, 37, 83, A224854, 11)   #37,83
    drLD(x, 60, 8, 47, 73, A224854, 11)   #47,73
    drLD(x, 48, 6, 61, 71, A224854, 11)   #61,71
    drLD(x, 12, 0, 79, 89, A224854, 11)   #79,89

#13 = A201816
    drLD(x, 76, -1, 13, 91, A224854, 13)   #13,91
    drLD(x, 94, 18, 19, 67, A224854, 13)  #19,67
    drLD(x, 94, 24, 37, 49, A224854, 13)  #37,49
    drLD(x, 76, 11, 31, 73, A224854, 13)  #31,73
    drLD(x, 86, 6, 11, 83, A224854, 13)   #11,83
    drLD(x, 104, 29, 29, 47, A224854, 13) #29,47
    drLD(x, 86, 14, 23, 71, A224854, 13)  #23,71
    drLD(x, 86, 20, 41, 53, A224854, 13)  #41,53
    drLD(x, 104, 25, 17, 59, A224854, 13) #17,59
    drLD(x, 14, 0, 77, 89, A224854, 13)   #77,89
    drLD(x, 94, 10, 7, 79, A224854, 13)   #7,79
```

```
    drLD(x, 76, 15, 43, 61, A224854, 13) #43,61

# 79 = A202112
    drLD(x, 10, -1, 79, 91, A224864, 79) #79,91
    drLD(x, 100, 22, 19, 61, A224864, 79) #19,61
    drLD(x, 136, 48, 7, 37, A224864, 79) #7,37
    drLD(x, 64, 8, 43, 73, A224864, 79) #43,73
    drLD(x, 80, 0, 11, 89, A224864, 79) #11,89
    drLD(x, 80, 12, 29, 71, A224864, 79) #29,71
    drLD(x, 116, 34, 17, 47, A224864, 79) #17,47
    drLD(x, 44, 2, 53, 83, A224864, 79) #53,83
    drLD(x, 154, 65, 13, 13, A224864, 79) #13,13
    drLD(x, 100, 26, 31, 49, A224864, 79) #31,49
    drLD(x, 46, 5, 67, 67, A224864, 79) #67,67
    drLD(x, 134, 49, 23, 23, A224864, 79) #23,23
    drLD(x, 80, 16, 41, 59, A224864, 79) #41,59
    drLD(x, 26, 1, 77, 77, A224864, 79) #77,77

# 77 = A201822
    drLD(x, 12, -1, 77, 91, A224864, 77) #77,91
    drLD(x, 138, 52, 19, 23, A224864, 77) #19,23
    drLD(x, 102, 28, 37, 41, A224864, 77) #37,41
    drLD(x, 48, 5, 59, 73, A224864, 77) #59,73
    drLD(x, 162, 72, 7, 11, A224864, 77) #7,11
    drLD(x, 108, 31, 29, 43, A224864, 77) #29,43
    drLD(x, 72, 13, 47, 61, A224864, 77) #47,61
    drLD(x, 18, 0, 79, 83, A224864, 77) #79,83
    drLD(x, 78, 0, 13, 89, A224864, 77) #13,89
    drLD(x, 132, 47, 17, 31, A224864, 77) #17,31
    drLD(x, 78, 16, 49, 53, A224864, 77) #49,53
    drLD(x, 42, 4, 67, 71, A224864, 77) #67,71

# 43 = A202105
    drLD(x, 46, -1, 43, 91, A224857, 43) #43,91
    drLD(x, 154, 65, 7, 19, A224857, 43) #7,19
    drLD(x, 64, 6, 37, 79, A224857, 43) #37,79
    drLD(x, 46, 5, 61, 73, A224857, 43) #61,73
    drLD(x, 116, 32, 11, 53, A224857, 43) #11,53
    drLD(x, 134, 49, 17, 29, A224857, 43) #17,29
    drLD(x, 44, 0, 47, 89, A224857, 43) #47,89
    drLD(x, 26, 1, 71, 83, A224857, 43) #71,83
    drLD(x, 136, 50, 13, 31, A224857, 43) #13,31
    drLD(x, 64, 10, 49, 67, A224857, 43) #49,67
    drLD(x, 116, 36, 23, 41, A224857, 43) #23,41
    drLD(x, 44, 4, 59, 77, A224857, 43) #59,77
```

```
# 41 = A202104
    drLD(x, 48, -1, 41, 91, A224857, 41) #41,91
    drLD(x, 42, 0, 49, 89, A224857, 41) #49,89
    drLD(x, 102, 24, 19, 59, A224857, 41) #19,59
    drLD(x, 120, 39, 23, 37, A224857, 41) #23,37
    drLD(x, 108, 25, 11, 61, A224857, 41) #11,61
    drLD(x, 72, 7, 29, 79, A224857, 41) #29,79
    drLD(x, 90, 22, 43, 47, A224857, 41) #43,47
    drLD(x, 150, 62, 13, 17, A224857, 41) #13,17
    drLD(x, 78, 12, 31, 71, A224857, 41) #31,71
    drLD(x, 30, 2, 73, 77, A224857, 41) #73, 77
    drLD(x, 60, 9, 53, 67, A224857, 41) #53,67
    drLD(x, 90, 6, 7, 83, A224857, 41) #7,83

# 61 = A202113
    drLD(x, 28, -1, 61, 91, A224860, 61) #61,91
    drLD(x, 82, 8, 19, 79, A224860, 61) #19,79
    drLD(x, 100, 27, 37, 43, A224860, 61) #37,43)
    drLD(x, 100, 15, 7, 73, A224860, 61) #7,73
    drLD(x, 98, 16, 11, 71, A224860, 61) #11,71
    drLD(x, 62, 0, 29, 89, A224860, 61) #29,89
    drLD(x, 80, 17, 47, 53, A224860, 61) #47,53
    drLD(x, 80, 5, 17, 83, A224860, 61) #17,83
    drLD(x, 100, 19, 13, 67, A224860, 61) #13,67
    drLD(x, 118, 38, 31, 31, A224860, 61) #31,31
    drLD(x, 82, 18, 49, 49, A224860, 61) #49,49
    drLD(x, 80, 9, 23, 77, A224860, 61) #23,77
    drLD(x, 98, 26, 41, 41, A224860, 61) #41,41
    drLD(x, 62, 10, 59, 59, A224860, 61) #59,59

# 59 = A202101
    drLD(x, 30, -1, 59, 91, A224860, 59) #59,91
    drLD(x, 120, 38, 19, 41, A224860, 59) #19,41
    drLD(x, 66, 7, 37, 77, A224860, 59) #37,77
    drLD(x, 84, 12, 23, 73, A224860, 59) #23,73
    drLD(x, 90, 9, 11, 79, A224860, 59) #11,79
    drLD(x, 90, 19, 29, 61, A224860, 59) #29,61
    drLD(x, 126, 39, 7, 47, A224860, 59) #7,47
    drLD(x, 54, 3, 43, 83, A224860, 59) #43,83
    drLD(x, 114, 31, 13, 53, A224860, 59) #13,53
    drLD(x, 60, 0, 31, 89, A224860, 59) #31,89
    drLD(x, 60, 8, 49, 71, A224860, 59) #49,71
    drLD(x, 96, 18, 17, 67, A224860, 59) #17,67

# 73 = A195993
    drLD(x, 16, -1, 73, 91, A224862, 73) #73,91
```

```
    drLD(x, 124, 41, 19, 37, A224862, 73) #19,37
    drLD(x, 146, 58, 11, 23, A224862, 73) #11,23
    drLD(x, 74, 8, 29, 77, A224862, 73) #29,77
    drLD(x, 74, 14, 47, 59, A224862, 73) #47,59
    drLD(x, 56, 3, 41, 83, A224862, 73) #41,83
    drLD(x, 106, 24, 13, 61, A224862, 73) #13,61
    drLD(x, 106, 30, 31, 43, A224862, 73) #31,43
    drLD(x, 124, 37, 7, 49, A224862, 73) #7,49
    drLD(x, 34, 2, 67, 79, A224862, 73) #67,79
    drLD(x, 74, 0, 17, 89, A224862, 73) #17,89
    drLD(x, 56, 7, 53, 71, A224862, 73) #53,71

# 71 = A202129
    drLD(x, 18, -1, 71, 91, A224862, 71) #71,91
    drLD(x, 72, 0, 19, 89, A224862, 71) #19,89
    drLD(x, 90, 21, 37, 53, A224862, 71) #37,53
    drLD(x, 90, 13, 17, 73, A224862, 71) #17,73
    drLD(x, 138, 51, 11, 31, A224862, 71) #11,31
    drLD(x, 102, 27, 29, 49, A224862, 71) #29,49
    drLD(x, 120, 36, 13, 47, A224862, 71) #13,47
    drLD(x, 30, 1, 67, 83, A224862, 71) #67,83
    drLD(x, 150, 61, 7, 23, A224862, 71) #7,23
    drLD(x, 78, 15, 41, 61, A224862, 71) #41,61
    drLD(x, 42, 3, 59, 79, A224862, 71) #59,79
    drLD(x, 60, 6, 43, 77, A224862, 71) #43,77

primelist_A224854 = [i for i,x in enumerate(A224854) if x == 0]
print("A224854", primelist_A224854)
print("A224854", len(primelist_A224854))

primelist_A224855 = [i for i,x in enumerate(A224855) if x == 0]
print("A224855", primelist_A224855)
print("A224855", len(primelist_A224855))

primelist_A224856 = [i for i,x in enumerate(A224856) if x == 0]
print("A224856", primelist_A224856)
print("A224856", len(primelist_A224856))

primelist_A224857 = [i for i,x in enumerate(A224857) if x == 0]
print("A224857", primelist_A224857)
print("A224857", len(primelist_A224857))

primelist_A224859 = [i for i,x in enumerate(A224859) if x == 0]
print("A224859", primelist_A224859)
print("A224859", len(primelist_A224859))
```

```
primelist_A224860 = [i for i,x in enumerate(A224860) if x == 0]
print("A224860", primelist_A224860)
print("A224860", len(primelist_A224860))

primelist_A224862 = [i for i,x in enumerate(A224862) if x == 0]
print("A224862", primelist_A224862)
print("A224862", len(primelist_A224862))

primelist_A224864 = [i for i,x in enumerate(A224864) if x == 0]
print("A224864", primelist_A224864)
print("A224864", len(primelist_A224864))

primelist_A224865 = [i for i,x in enumerate(A224865) if x == 0]
print("A224865", primelist_A224865)
print("A224865", len(primelist_A224865))
```

10 Results

Big Omega.

For A201804, using the Python implementation we start with a finite list of 0 then add +1 to every address that is generated by the solutions. When we read the list we discover values including 0,1,2,3, ..., etc. These values represent amplitudes associated with the number of factors for a given number. We see primes have zero amplitude, but what about 1,2,3,4,5 ..., etc. as values? The +1 events which accumulate can be understood in terms of *factor families.* The branches for factor families are the solutions to (2*n)+1, where n is a seed value. *What is meant by seed value?*

For the number 1 in a cell, the base-10 Conway Number we recover is a semi-prime of type (a*b). Both factors are prime. If we use 1 as an input in the equation (2*n)+1 we get 3. Every address containing 3 is of type a*b*c with a,b,c prime. (2*n)+1 where n=3 is 7. For 7 we have a*b*c*d. (2*n)+1 is a *branching operator* which we can use to establish the underlying *factor forms* or *factor-type families.*

Here is a results table for some of the seed values and equivalent amplitudes for (2*n)+1 for A201804:

1 = semiprime (a*b)
3 = a*b*c
7 = a*b*c*d
15= a*b*c*d*e
...break
2 = (a*a)*b
5 = (a*a)*b*c
11 = (a*a)*b*c*d
...break
4 = (a*a*a*a)*b

9 = (a*a*a*a)*b*c
19 = (a*a*a*a)*b*c*d
...break
6 = (a*a*a*a*a*a)*b
13 = (a*a*a*a*a*a)*b*c
27 = (a*a*a*a*a*a)*b*c*d
...break
8 = (a*a)*(b*b)*c
17 = (a*a)*(b*b)*c*d
35 = (a*a)*(b*b)*c*d*e
...break
10 = (a*a*a*a*a*a*a*a*a*a)*b = a(10)*b(1)
...break
12 = a(12)*b(1)
...break
14 = a(4)*b(2)*c(1)
29 = a(4)*b(2)*c(1)*d(1)
59 = a(4)*b(2)*c(1)*d(1)*e(1)
...break

Big Omega is derived from the amplitude that is generated by accumulation of outputs at the address. Big Omega for the Twin Prime sieve requires operating A201804 and A201816 in separate containers and then taking the union set of the two lists so as to preserve Big Omega for each subsequence. The molecular sieve can then be configured to produce Big Omega or the characteristic function.

11 Conclusion

I began this playthrough of the Game of Primes intent on returning with a prime printing function. This was a quest to locate an operator or finite number of operators which could faithfully directly reproduce arbitrarily long sequences of prime numbers and do so as easily as we determine the composite certificates for trivial numbers. How would one explore for such a function?

The complement to A201804 has been demonstrated. The implication is that a simple generator is capable of producing a complementary quasi-random sequence. If we say it is possible for a system such as A000027 to simultaneously contain randomness (A000040) and non-randomness (the complement to A000040), then by removing all non-randomness you are necessarily left with the random component. A simple order can be revealed within the composite distribution complementary to A201804 precisely because the prime distribution is functionally random.