Finding large Primes using Standing Wave Harmonics

By TakenItSeriously, June 22, 2017 in Mathematics

★★★★★

TakenItSeriously

Molecule

Senior Members

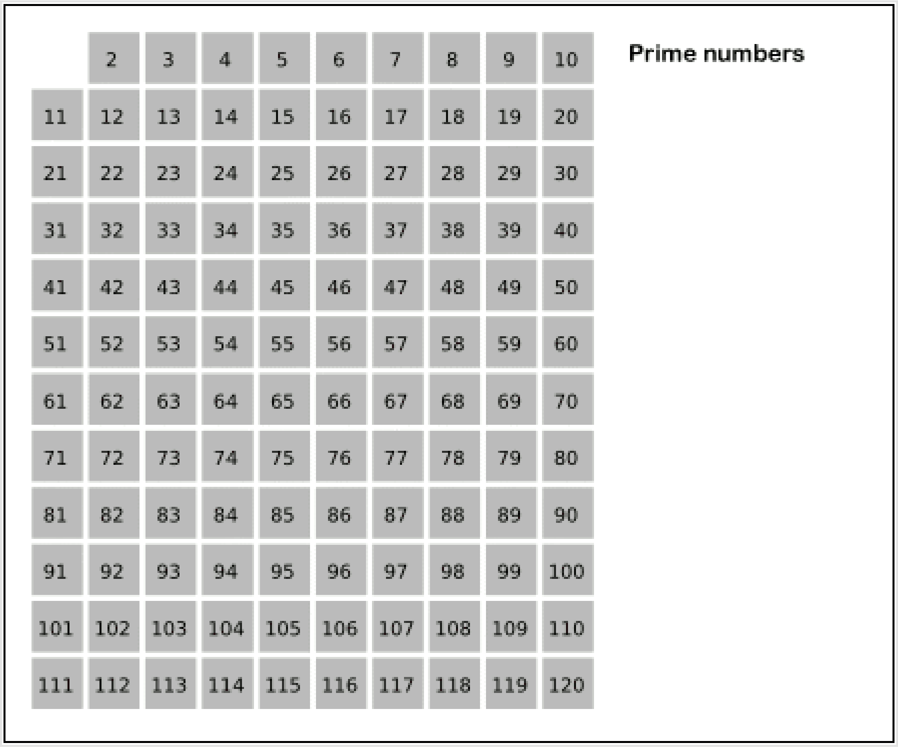
⊕ 9

468 posts

Location: Silicon Valley

Posted June 22, 2017 (edited)

I discovered an improved method for finding large prime numbers by ignoring the prime numbers alltogether and only focus on the wavelike patterns of prime factors which is loosely related to the Sieve of Erasthonenes.



	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
91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110
111	112	113	114	115	116	117	118	119	120

Prime numbers

Figure 1: The Sieve of Erasthonenes

By noting that prime factors occur at regular intervals. ie multiples of 2 reoccur every other number, multiples of 3 reoccur at every third number, etc. we can leverage this periodicity of prime factors to identify all non-prime positions within a predifined large range of natural numbers arranged in an array.

This periodicity of prime factors means that we can apply the concepts of Standing Wave Harmonics to find all composite numbers with a given range based on these wave patterns. Therefore we will also know the relative positions of all prime numbers within the same range.

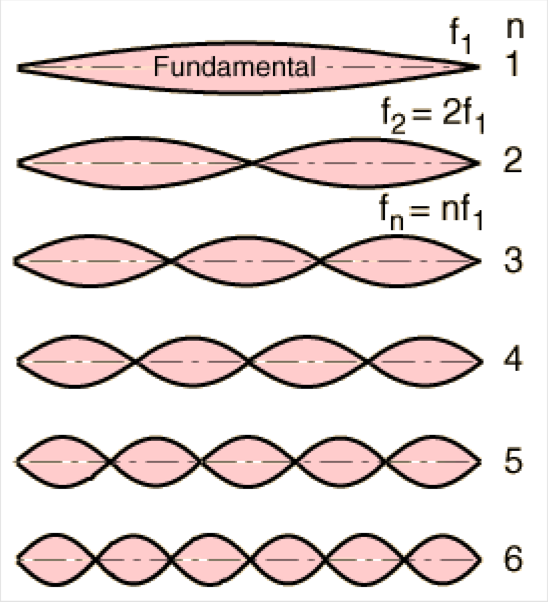


Figure 2: Patterns of Standing Wave Harmonics

These positions can be stored as a "0" for non-prime or "1" for prime rather than needing to store the entire number, thus alleviating the computational issues with large numbers.

The method

The key is to arrange numbers into rows of N numbers which is defined by the product of the first n primes.

$N = P_1 x\ P_2 ... x\ Pn$

The standing wave like effect of those first prime numbers will cause their prime factors allign into their periodic columns which, in turn causes the primes to allign themselves within the remaining collumns though they will be intermingled with other composite that are defined by primes greater than Pn.

for N = 2x3 = 6

multiples of 2:

xx, **02**, xx, 04, xx, 06,

xx, 08, xx, 10, xx, 12,

xx, 14, xx, 16, xx, 18,

xx, 20, xx, 22, xx, 24,

xx, 26, xx, 28, xx, 30,

xx, 32, xx, 34, xx, 36,

multiples of 3:

xx, xx, **03**, xx, xx, 06,

xx, xx, 09, xx, xx, 12,

xx, xx, 15, xx, xx, 18,

xx, xx, 21, xx, xx, 24,

xx, xx, 27, xx, xx, 30,

xx, xx, 33, xx, xx, 36,

after we combine those multiples we get:

xx, **02, 03**, 04, xx, 06,

xx, 08, 09, 10, xx, 12,

xx, 14, 15, 16, xx, 18,

xx, 20, 21, 22, xx, 24,

xx, 26, 27, 28, xx, 30,

xx, 32, 33, 34, xx, 36,

Therefore we can see that the prime numbers must be located within the remaining columns that are not already occupied by the composite numbers.

For prime numbers greater than Pn their prime factors form diagonal patterns which define the gaps between the prime numbers in those remaining collumns.

xx, xx, xx, xx, **05**, xx,

xx, xx, xx, 10, xx, xx,

xx, xx, 15, xx, xx, xx,

xx, 20, xx, xx, xx, xx,

25, xx, xx, xx, xx, 30,

xx, xx, xx, xx, 35, xx,

or

xx, xx, xx, xx, xx, xx,

07, xx, xx, xx, xx, xx,

xx, 14, xx, xx, xx, xx,

xx, xx, 21, xx, xx, xx,

xx, xx, xx, 28, xx, xx,

xx, xx, xx, xx, 35, xx,

combining all waves we get:

xx, **02, 03**, 04, **05**, 06,

07, 08, 09, 10, xx, 12,

xx, 14, 15, 16, xx, 18,

xx, 20, 21, 22, xx, 24,

25, 26, 27, 28, xx, 30,

xx, 32, 33, 34, 35, 36,

note the numbers in bold are recognized as primes as well as those numbers marked as xx i.e. no prime divisors. The exception is 01 which always shows up as a prime number but you can just ignore it.

With the numbers arranged in a 2D array, we can treat it like a matrix and therefore we can ignore the value of the numbers themselves and only define the relative positions of all composite numbers within the array which of course also defines the relative positions of all of the prime numbers within the array.

Since we are only treating positions of the array as prime (1) or non-prime (0), then we can alleviate the issues with computational complexity of long numbers by only dealing with their primality and position.

Example:

By arranging numbers into rows of N numbers then you will notice that all primes will become aligned into columns that number fewer than columns of composite numbers.

e.g.

For the first 2 primes determine the positions of the first 11 primes within the first 36 numbers.

N = 2x3 = 6

0,1,1,0,1,0,

1,0,0,0,1,0,

1,0,0,0,1,0,

1,0,0,0,1,0,

0,0,0,0,1,0,

1,0,0,0,0,0,

For the first 3 primes we can define the positions of the first 29 primes within the first 100 numbers.

N = 2x3x5 = 30

0,1,1,0,1,0,1,0,0,0,1,0,1,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0, 1,0,0,0,0,0,1,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,

1,0,0,0,0,0,1,0,0,0,1,0,1,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0,1,0,

1,0,0,0,0,0,1,0,0,0,1,0,1,0,0,0,1...

where

1 ≈ prime number position

0 ≈ composite number position

We can scale the Node ranges to the size of the prime numbers which we are focused on by adding more primes to the composite node.

$N_2 = 2x3 = 6$ $N^2 = 36$

$N_3 = N_2x5 = 30$ $N^2 = 900$

$N_4 = N_3x7 = 210$ $N^2 = 44,100$

$N_5 = N_4x11 = 2310$ $N^2 = 5,336,100$

$N_6 = N_5x13 = 30,030$ $N^2 = 901,800,900$

$N_7 = N_6x17 = 510,510$ $N^2 = 260,620,460,100$

$N_8 = N_7x19 = 9,699,690$ $N^2 = 94,083,986,096,100$

$N_9 = N_8x23 = 223,092,870$ $N^2 = 49,770,428,644,836,900$

$N_{10} = N_9x27 = 6,023,507,490$ $N^2 = 36,282,642,482,086,100,100$

$N_{11} = N_{10}x31 = 186,728,732,190$ $N^2 = 34,867,619,425,284,742,196,100$

$N_{12} = N_{11}x37 = 6,908,963,091,030$ $N^2 = 47,733,770,993,214,812,066,460,900$

...

Nn

Below is an excel spreadsheet that uses the first 4 prime numbers to define a node length:

N = 2x3x5x7 = 210

This node was then used to find the first 8555 prime number positions from within 88,200 natural numbers by using the wave patterns of all prime factors to eliminate all composite numbers leaving only primes behind.

I was able to validate the correctness of those prime numbers to be 100% correct compared to downloaded samples.

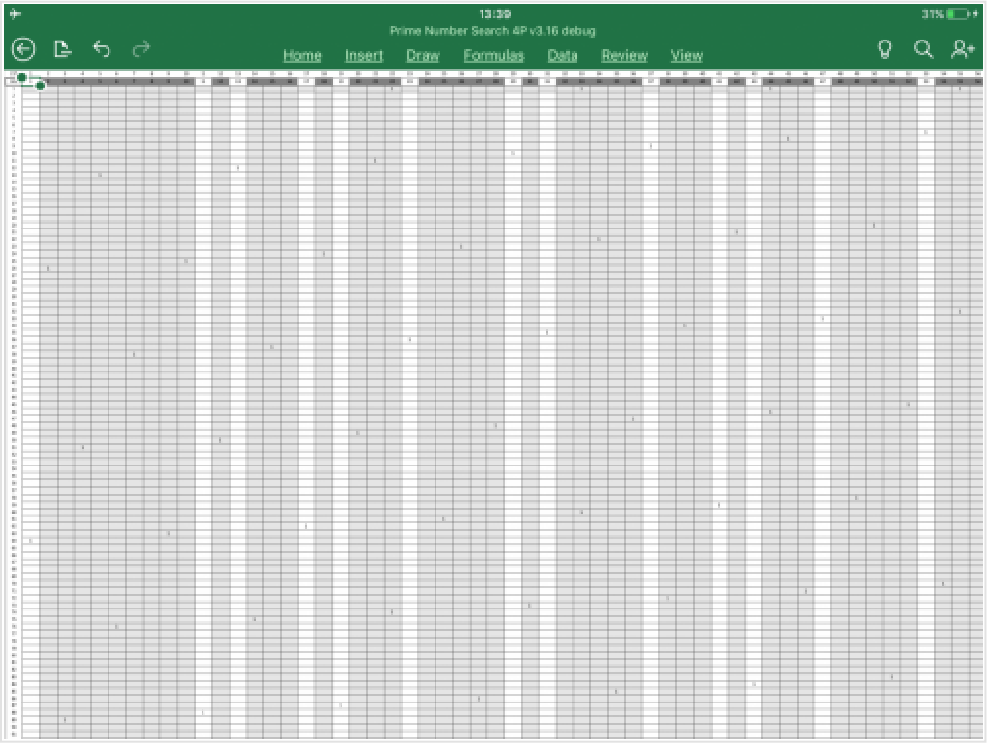
While this is not important for finding small primes, the method uses the same process for finding prime number positions without needing access or operate on the large numbers themselves.

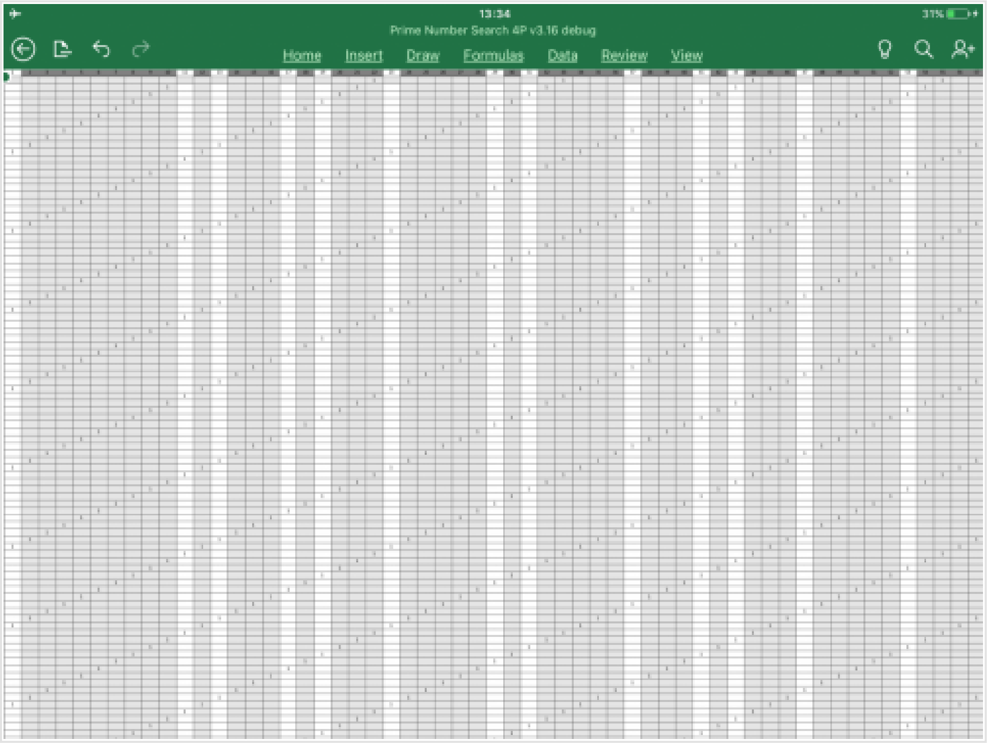
All prime number positions can be predifined using prime factor wave-like patterns so that accessing large prime numbers should be usable in real time.

While the spreadsheet had over 70 sheets involved, I show sample sheets for two prime factor patterns for 11 and 101.

The third sheet shows the prime number positions are shown in red within a field of natural numbers shown in grey, although the numbers themselves are just there to provide context for the result.

The actual numbers used in formulas are simply a 0 or 1 to show primality as shown in the fourth sheet.


13:39
Prime Number Search 4P v3.16 debug
31%
Home
Insert
Draw
Formulas
Data
Review
View


13:34
Prime Number Search 4P v3.16 debug
31%
Home
Insert
Draw
Formulas
Data
Review
View

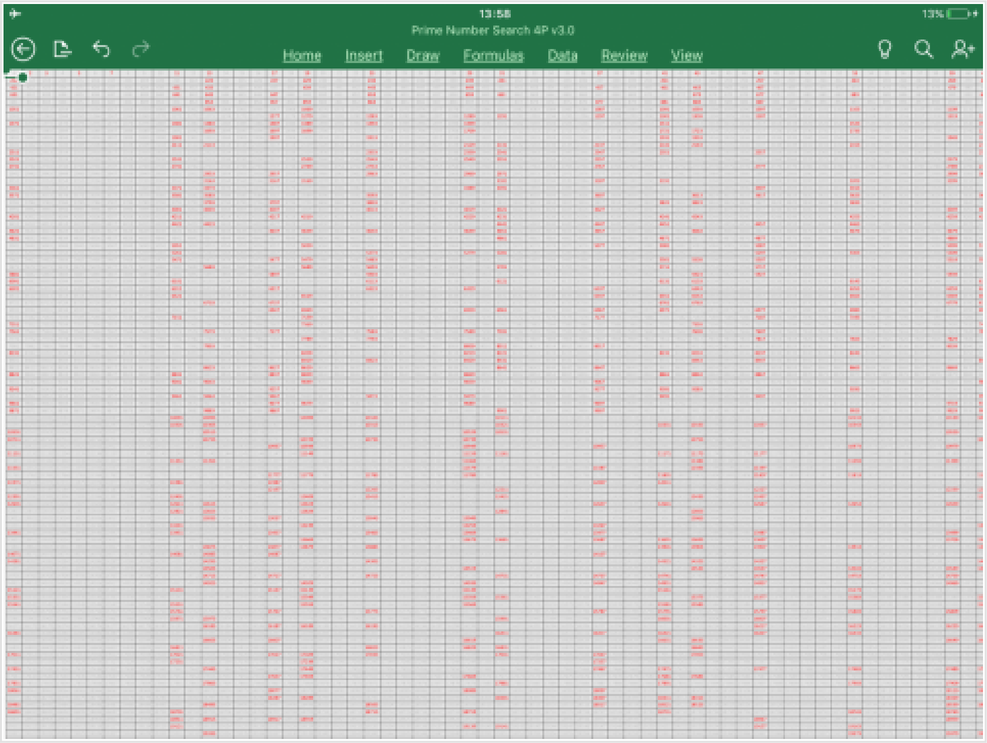


Figure 3: Proof of concept in using an Excel spreadsheet which uses wave patterns of prime factors shown as the patterns of "1" intersecting harmonic columns shown as the white cells in order to derive the positions for the first 8555 prime numbers within the first 88,200 natural numbers.

While creating this kind of matrix for the largest prime numbers in question would be a large undertaking the memory and speed cost can be greatly reduced relative to current methods by dealing with relative prime number positions only without needing to store or perform operations on large prime numbers directly.

It may even be possible to model prime number positions using electronic signal waves or perhaps light wave frequencies at wavelengths that correspond with the prime factor periodicity in order to determining where they intersect with prime harmonic standing waves in order to identify the prime number positions.

Ultimately, with access to large primes in near real time, it should be possible to use methods of infinite compression, by taking large strings of binary data, and converting it to small number keys using universally accessable large mersenne primes to compress and decompress the data at either end.

Edited June 23, 2017 by TakenItSeriously

imatfaal

lazy do-nothing mudslinger

Moderators

★★ Staff

⊕ 2481

7809 posts

Location: St James's Park

Posted June 23, 2017

It is a basic sieve - in fact I fail to see any substantive difference from the sieve of eratosthenes

Sieve of Eratosthenes runs with time / complexity of the Order $\mathcal{O}(n \log\log n)$ Whereas more streamlined algorithms can run at $\mathcal{O}(n)$ or even if memory is not an issue at $\mathcal{O}(\frac{n}{\log\log n})$

TakenItSeriously

Molecule

Senior Members

⊕ 9

468 posts

Location: Silicon Valley

Posted June 23, 2017 (edited)

False Alarm.

Reveal hidden contents

> **On 6/23/2017 at 5:08 AM, imatfaal said:**
>
> It is a basic sieve - in fact I fail to see any substantive difference from the sieve of eratosthenes
>
> Sieve of Eratosthenes runs with time / complexity of the Order $\mathcal{O}(n \log\log n)$ Whereas more streamlined algorithms can run at $\mathcal{O}(n)$ or even if memory is not an issue at $\mathcal{O}(\frac{n}{\log\log n})$

You're right in part. It uses something like the SoE for the first line segment.

However it's quite a bit different after that.

The SoE works on a single infinite line, has only absolute references, counts only the pitches for finding prime factors.

This means that it must starts from the first prime every time, must save actual number values to record primaliety information, eventually runs out of space on the hard drive.

However, when breaking the line into segments at lengths equal to a resonant nodes,

$N = P_1 \times P_2 ... xPn$

Then it can leverage 2D wave like patterns vs the 1D pitches the SoE depends on.

For primes < Pn it will create vertical columns of composite multiples that can be ignored.

That reduces the numbers that need to be acted on by 80-95%.

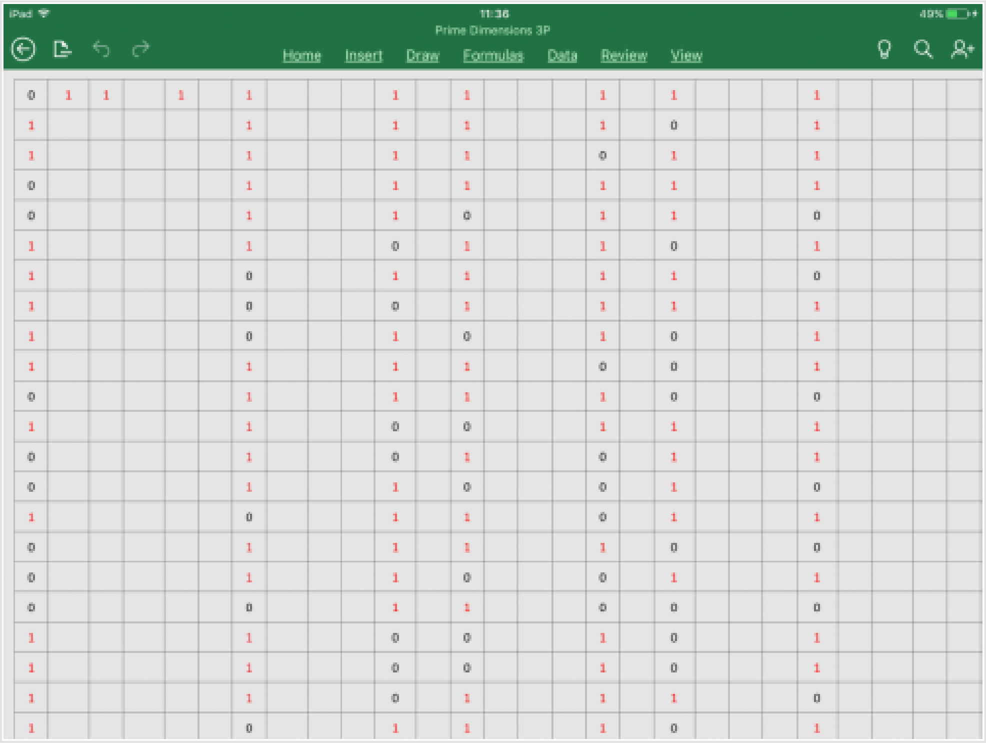
For primes > Pn then the composite multiples of those primes create consistant diagonal patterns that can be defined by a starting position along with a Δx and Δy slope. Finding the points of intersection in the remaining collumns defines the remaining composite numbers, meaning all prime locations are what are left behind.

Therefore rather than marking all primes and composite numbers locations by recording their large numbers that could be humdreds of millions of digits long on the hard drive, the new method finds the relative positions of primes and composite numbers, saving only the data of the collumn as a binary string.

Furthermore, the 2D region can be localized to a using a datum which can't be done in 1D. A datum makes the position references all based on local numbers instead of large numbers.

The format of the new method is optimized to need only a tiny fraction of the space to record all primality data. eg for any region of 44,100 numbers, It only needs 44 binary strings 110 bits each regardless,of the number sizes involved.

example:



0	1	1		1		1				1		1				1		1				1				
1						1				1		1				1		0				1				
1						1				1		1				0		1				1				
0						1				1		1				1		1				1				
0						1				1		0				1		1				0				
1						1				0		1				1		0				1				
1						0				1		1				1		1				0				
1						0				0		1				1		1				1				
1						0				1		0				1		0				1				
1						1				1		1				0		0				1				
0						1				1		1				1		0				0				
1						1				0		0				1		1				1				
0						1				0		1				0		1				1				
0						1				1		0				0		1				0				
1						0				1		1				0		1				1				
0						1				1		1				1		0				0				
0						1				1		0				0		1				1				
0						0				1		1				0		0				0				
1						1				0		0				1		0				1				
1						1				0		0				1		0				1				
1						1				0		1				1		1				0				
1						0				1		1				1		0				1				

Without the memory limitations, accessing large primes or even large Mersenne Primes should just be a matter of indexing them in real time.

Again, if that's true then infinite compression may just be a step away based on an idea that was told to me by a brilliant man whom I was introduced to through the Stanford professor who taught me EE.

Edited June 23, 2017 by TakenItSeriously

imatfaal

lazy do-nothing mudslinger

Moderators

★★ Staff

2481

7809 posts

Location: St James's Park

Posted June 23, 2017

It is still just an optimized sieve of eratosthenes. It is neat - but to convince me otherwise I would have to see a breakdown showing how with n as a very large that the time taken grew at a smaller rate than n. Because sieves based on Erathosthenes with wheel portions and clever bits from Atkins can get to O(n)

And this is not a potential solution to large number factorization. And infinite compression is silly

Trurl

Quark

Senior Members

13

139 posts

Posted June 24, 2017

I don’t claim to understand this. But if you are using waves that have a frequency of the factor (say 2…4...6…8 and 4…8…12) than the next bracket (numbers above 9; in ten digit brackets) you would only have to use the Prime numbers for the next wave. (11…13…17…19) and any new waves would have to be Prime numbers since those are the only new waves.

So, you use the factors of the non-prime waves and just add those numbers that do not fit the waves (which are Prime numbers) to the new waves.

This is just my understanding. But is this how the sieve works? And is this what the matrices were trying to predict, in this way?

I like the idea of waves. If the wave was harmonic, always having the same frequency, the starting point of the wave would always be at a Prime number. The problem is finding where those waves start because you must know the Prime numbers that came before.

I do like the graphical depiction using waves. But it is the same as using the modulus. There would have to be an anomaly in the wave that finds Prime numbers. I think it would be hard to find Prime numbers by eliminating those that were not Prime.

Of course, I am not claiming any of this is right. I am trying to learn. But a wave theory is interesting. It reminds me of a 3 stage alternator were adding waves equals a line. But knowing that the waves frequency is the modulus, the Prime number would be between all the non-Prime wave's frequencies. Like in your diagram of the waves, the Prime number will fall between the sum of wave frequencies. But I don't know if that makes anything in finding the Prime numbers easier, because of the computation of all the known, non-Prime waves.

There may not be a pattern in the placement of the numbers in the sieve, however do you think it is possible for a pattern in the waves?

Again, this is just how I understood it.

TakenItSeriously

Molecule

Senior Members

9

468 posts

Location: Silicon Valley

Posted June 26, 2017 (edited)

> **On 6/23/2017 at 2:46 PM, imatfaal said:**
>
> It is still just an optimized sieve of eratosthenes. It is neat - but to convince me otherwise I would have to see a breakdown showing how with n as a very large that the time taken grew at a smaller rate than n. Because sieves based on Erathosthenes with wheel portions and clever bits from Atkins can get to O(n)

Regarding the wheel modification, it essentially ignored even numbers and redundant prime factors since only one prime factor is required to define a composite number and I had already accounted for all of those mods in the Excel proof. It's difficult to display the algorithms contained in each cell directly and Excel syntax is not very reader friendly.

Also the last completed version was corrupted by hackers so it won't even open in the first place. Right now I'm only recreating portions to be able to display examples.

The Harmonic Wave modification does much more than that and is a real game changer.

> **Quote**
>
> And infinite compression is silly

One form of infinite compression is already embedded in the method when you consider that the SoE saves the primality data as absolute numbers where a single number could take up hundreds of gigabytes of data.

Using wave patterns the same data can be stored as local coordinate data within a local matrix reference frame with a single long index to the matrix datum.

> **On 6/24/2017 at 1:42 PM, Trurl said:**
>
> I dont claim to understand this. But if you are using waves that have a frequency of the factor (say 24...68 and 4812) than the next bracket (numbers above 9; in ten digit brackets) you would only have to use the Prime numbers for the next wave. (11131719) and any new waves would have to be Prime numbers since those are the only new waves.
>
> So, you use the factors of the non-prime waves and just add those numbers that do not fit the waves (which are Prime numbers) to the new waves.
>
> This is just my understanding. But is this how the sieve works? And is this what the matrices were trying to predict, in this way?
>
> I like the idea of waves. If the wave was harmonic, always having the same frequency, the starting point of the wave would always be at a Prime number. The problem is finding where those waves start because you must know the Prime numbers that came before.
>
> I do like the graphical depiction using waves. But it is the same as using the modulus. There would have to be an anomaly in the wave that finds

> Prime numbers. I think it would be hard to find Prime numbers by eliminating those that were not Prime.
>
> Of course, I am not claiming any of this is right. I am trying to learn. But a wave theory is interesting. It reminds me of a 3 stage alternator were adding waves equals a line. But knowing that the waves frequency is the modulus, the Prime number would be between all the non-Prime waves frequencies. Like in your diagram of the waves, the Prime number will fall between the sum of wave frequencies. But I dont know if that makes anything in finding the Prime numbers easier, because of the computation of all the known, non-Prime waves.
>
> There may not be a pattern in the placement of the numbers in the sieve, however do you think it is possible for a pattern in the waves?
>
> Again, this is just how I understood it.

Think of it like this:

2 is a prime factor for all numbers divisible by 2,

3 is a prime factor for all numbers divisible by 3,

5 is a prime factor for all numbers divisible by 5 and so on.

The thing that each of those series of prime factors have in common is that they each occur at a constant interval or they each have a constant period though each is unique from eachother.

Period is by definition the inverse of frequency.

So each prime number would be unique like the leading edge of a wave that has a different frequency.

What this means is thay by taking the product of the first n prime numbers you find a common node point for all n primes and then the pattern for those primes repeats itself.

So if we break the infinite number line into line segments that are each N numbers long and line them up side by side to form a matrix, then all composite numbers containing primes below the nth prime will allign and form columns.

The composite numbers containing prime factors that are larger than the nth prime will be arranged into diagonal patterns because their will always be a little extra piece of the wave left over after reaching the end of the line segment that will be be at the start of the next line segment.

To understand it better, you should just take the smallest node:

N = 2x3 = 6

You can then write the numbers down in rows of six and see what happens:

01 02 03 04 04 06

07 08 09 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

Note that this is a base 6 structure that I'm expressing as decimals because no one thinks of primes in base 6.

Because 3 goes into 6 twice there are two columns that contain all numbers divisible by 3, and because 2 goes into 6 3 times there are 3 columns that contain all numbers divisible by 2. Note that there is one collumn that is redundant where it contains numbers divisible by both two and three.

This leaves only two out of six collumns where all other numbers both prime and composits of prime factors larger than 3 must exist.

Since the multiples of the prime factors greater than three form diagonal lines they must intersect those vertical collumns periodically and thus they mark the position of another composite number.

Note that the columns containing the mix of primes/composites are vertical while the numbrs are incrementing horizontally within the matrix, thus we gain the advantage of havig a two way reference system to mark the relative positions of those numbers in a 2D plane.

Note that we ave taken a 1D number line and just converted it into a 2D region of space. We could actually do this again to form a 3D region of space by multiplying another series of primes together, but I don't have the ability to demonstrate this in excel or the bandwidth to create a new software application that could achieve this.

However if we keep it a 2D rectangle of infinite length then we can use the next series of prime products, e.g. Ny = 5x7 to find a vertical node pattern that is 35 squares long and those prime factors also repeat every 35 squares. more primes we add to the second product the longer the rectangle becomes up to infinity.

Now we can take any square and project it down the line to any point in space and be able to calculate the waves in that region using local coordinates instead of the large prime numbers themselves.

This is the infinite, or unbounded if you prefer compression that I was speaking about before. This can be further exploited in other forms to transmit data of any form which is another topic.

Edited June 26, 2017 by TakenItSeriously

Trurl

Quark

Senior Members

13

139 posts

Posted July 6, 2017

(((((x^2*PNP^4 + 2*PNP^2 * x^5) + x^8)/
PNP^4) - ((1 - x^2/(2*PNP)))) * ((PNP^2/x^2))) == PNP^2

TakeItSeriously,

I have a question. You are looking for a pattern in Prime numbers, where to predict larger and larger values. But have you ever tried it to work backwards to find factors? That is the products of large Prime numbers.

I have an equation which I can’t solve, but I argue that it gives a linear progression towards the smallest of the two products of the large Prime number. I know that it gives y, knowing x. That isn’t very useful. However, with several test values it gives a feeling of where x is.

I was thinking this “Standing Wave Harmonics” will be solved by using the wave harmonics and a form of my equation. I have no scientific proof. I am just stating an idea that may need testing. I do know a little bit about matrices. They are basically vectors. And vectors are why I designed my statement that a logarithmic spiral could show a pattern in Prime numbers and why I designed the ever-complicated triangles.

Let me know if such an idea is feasible. My equation is a one-to-one, increasing function. Knowing as little as I know, I think your pattern of waves applied to my equation may do something. I just don’t know what. It may do nothing for all I know.

Let me know what you think. See my Product Primes post.

Country Boy
Molecule

Senior Members
61
490 posts

Posted July 6, 2017

"Factors of large prime numbers"????

Trurl
Quark

Senior Members
13
139 posts

Posted July 7, 2017

> **Quote**
>
> "Factors of large prime numbers"????

I mean, the product whose factors are 2 Prime numbers.

Know the product find the factors, instead of finding the Prime placement along a modulus in a wave. Start with the product and use the wave to see if there are any patterns in the factorization.

Trurl
Quark

Senior Members

Posted July 11, 2017

(((((x^2*PNP^4 + 2*PNP^2 * x^5) + x^8)/
PNP^4) - ((1 - x^2/(2*PNP)))) * ((PNP^2/x^2))) == PNP^2

You don’t have to use this clunky equation.

13
139 posts

I have about 5 to 10 equations that could be plugged into the wave distribution.

I will compile a list.

But basically, in my old posts the left or right side will work when separated from the equation.

So

2*(85^2*x^2) + x^5 = 0, where 85 = N and x is the smaller factor.

If anyone is interested I will compile a list. I wrote the above shortened equation without testing it, so I hope it works. Over the next few days I will compile a list.

TakenItSeriously
Molecule

Senior Members
9
468 posts
Location: Silicon Valley

Posted July 12, 2017

> **On 7/11/2017 at 12:45 PM, Trurl said:**
>
> (((((x^2*PNP^4 + 2*PNP^2 * x^5) + x^8)/
>
> PNP^4) - ((1 - x^2/(2*PNP)))) * ((PNP^2/x^2))) == PNP^2
>
> You dont have to use this clunky equation.
>
> I have about 5 to 10 equations that could be plugged into the wave distribution.
>
> I will compile a list.

> But basically, in my old posts the left or right side will work when separated from the equation.
>
> So
>
> 2*(85^2*x^2) + x^5 = 0, where 85 = N and x is the smaller factor.
>
> If anyone is interested I will compile a list. I wrote the above shortened equation without testing it, so I hope it works. Over the next few days I will compile a list.

While I believe it is possible to factorize large semiprimes more efficiently than can currently be done, I have no interest in breaking security protocals at least until companies replace using prime factor security.

Trurl

Quark

Senior Members

⊕ 13

139 posts

Posted July 13, 2017

> **On 7/12/2017 at 4:26 AM, TakenItSeriously said:**
>
> While I believe it is possible to factorize large semiprimes more efficiently than can currently be done, I have no interest in breaking security protocals at least until companies replace using prime factor security.

I admire your ethics and integrity. I do not want to necessary break RSA to steal information. I just wonder if it could be done. I have really read-up on cryptography, including a couple of Bruce Schneier's books. It is more of a competition to see who can encipher and who can decipher. After all, if there is a hole in the Prime factorization problem we should know. But you don't think my work is close, do you? Of course not. After all, it is a one-way function and should be impossible to solve. If you can solve it, it is no longer a one-way. So, I try to figure it out, not to destroy public key cryptography, because it should be impossible for me to do so. I am after the mathematical answer and not the breaking of technology.

Ironically, finding Prime numbers and their patterns is also a threat to security. Not necessarily a bad one, but if your standing wave harmonics works then new cryptography must be built. I know that is not why you are looking for patterns in Prime numbers, but it must be considered. It goes back to the enciphers vs. deciphers.

I understand where how you are thinking. A pattern in Prime numbers is beautiful, where destroying RSA isn’t. Agreed. I don’t think I can crack RSA. I have only found patterns. But if a pattern did work, it may cause disaster, where math is meant to create.

But for a pattern in Prime numbers I have always imagined it as a logarithmic spiral. But all end this post there, because I might make you interested in factoring instead of the Prime sieve.

Daedalus

Molecule

Senior Members

312

709 posts

Location: Oklahoma, US

Posted July 19, 2017 (edited)

Instead of dealing with real valued functions such as wave harmonics, why not try an integer based approach. In my Seventh Challenge, I post a function that completely maps out the factors for any number. Perhaps you can use the equations.

> **On 8/20/2013 at 4:46 AM, Daedalus said:**
>
> **Edit: It's important to understand that the lists below are based on factors of 2 for $f(x)=2x$, factors of 3 for $f(x)=3x$, etc... The final equation at the bottom is the one that lists factors for any number x.**
>
> When I analyzed the sequence of exponents for the factors 2 through 8, I noticed that the ones repeat themselves in a regular pattern. This led me to look for a pattern in the twos, threes, and so forth. As expected, the exponents cover the entire set of natural number and repeat in regular patterns.
>
> {1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 6, ...}

{1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 4, 1, 1, 2, 1, 1, ...}

{1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 3, ...}

{1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 1, 1, 1, 1, 3, 1, 1, 1, 1, 2, 1, 1, ...}

{1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 2, 1, 1, ...}

{1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, ...}

{1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, ...}

Let's look at the sequence of exponents for factors of two.

{1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 6, ...}

We can easily see that the ones begin at the first position and repeat every other number. The twos begin at the second position and repeat every four numbers, and the threes begin at the fourth position and repeat every eight numbers. If we chart this out we'll get:

{1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 6, ...}

{1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 6, ...}

{1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 6, ...}

{1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 6, ...}

exp	start	spacing
1	1	2
2	2	4
3	4	8
4	8	16
...	...	...

If you look at the sequence of exponents for factors of three, you will notice that the ones are doubled up in the sequence:

{1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 4, 1, 1, 2, 1, 1,

...}

This means that there are two equations that predict the positions of the ones, twos, threes, etc... And once again, if we chart this out, we get:

{1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 4, 1, 1, 2, 1, 1, ...}
{1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 4, 1, 1, 2, 1, 1, ...}

{1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 4, 1, 1, 2, 1, 1, ...}
{1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 4, 1, 1, 2, 1, 1, ...}

{1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 4, 1, 1, 2, 1, 1, ...}
{1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 3, 1, 1, 2, 1, 1, 2, 1, 1, 4, 1, 1, 2, 1, 1, ...}

exp	start	spacing
1	1	3
1	2	3
2	3	9
2	6	9
3	9	27
3	18	27
...	...	...

We can see that exponential equations govern the pattern as well as the start indices and spacings. Furthermore, if we had an equation that would produce the red highlighted numbers in their exact position, but be zero everywhere else, then we could sum all the sequences together to have a generating function.

{1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, ...}
+
{0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, 0, 0, 2, 0, ...}
+
{0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, ...}
+
{0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0, ...}
=

{1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, ...}

This is where the modulo operation comes into play. If we use Mathematica, we can see that Mod[x,2] generates the sequence of ones as found in the exponents for the factors of two.

{1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, ...}

However, the sequence generated by Mod[x,3] has a bunch of twos in it, and we need a sequence of ones and zeros.

{1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2, 0, 1, 2, ...}

Here's the trick. We know that Mod[1,3] = 1 and that Mod[3x, 3]=0. So if we take the factor 3 and raise it to the power of Mod[x,3], $3^{\mathrm{Mod}[x,3]}$, we'll get a one when Mod[x,3]=0, and a multiple of 3 everywhere else:

{3, 9, 1, 3, 9, 1, 3, 9, 1, 3, 9, 1, 3, 9, 1, 3, 9, 1, 3, 9, ...}

If we apply the modulo operation again, we get $\mathrm{Mod}[\ 3^{\mathrm{Mod}[x,\ 3]},\ 3\ ]$:

{0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, ...}

This allows us to generate a one every three numbers and a zero everywhere else. This works for any factor, and by using the floor function instead of the modulo operator, we arrive at a function that generates any spacing of ones that we need with zeroes everywhere else:

$$\mathrm{FactorMap}\,(n,x) = n\left(\frac{n^{x-1}}{n^{n\left\lfloor\frac{x}{n}\right\rfloor}} - \left\lfloor\frac{n^{x-1}}{n^{n\left\lfloor\frac{x}{n}\right\rfloor}}\right\rfloor\right)$$

If you prefer, you can still use the following in Mathematica:

$$\mathrm{FactorMap}\,(n,x) = \mathrm{Mod}[\ n^{\mathrm{Mod}[x,\ \mathrm{n}]},\ \mathrm{n}\ ]$$

Edit: I later discovered a better function for mapping factors:

$$\mathrm{FactorMap}\,(n,x) = \frac{\mathrm{Mod}[x-1,\ \mathrm{n}]-\mathrm{Mod}[x,\ \mathrm{n}]+1}{n}$$

The reason I call this function the FactorMap is because it produces a 1 whenever a number contains the factor or a zero otherwise. Furthermore, if we sum the sequences produced by the function, we'll get a count of unique factors for each number.

numbers
1 2 3 4 5 6 7 8 9 ...
2 {0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1}
3 {0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0}
Factors: 4 {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1}
5 {0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1}
6 {0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0}

Now that we have this function, we can choose the correct spacing, multiply it by the value of our exponents, and set the correct start indices. By using the charts we made earlier, we can derive such a function (i and j are indices used to offset to the correct start position and choose the proper spacing):

$$F(n, x, i, j) = j\,\mathrm{FactorMap}\left(n^{j-1}, x - i\,n^{j}\right)$$

The only thing left is to determine how many times we need to sum F in order to create the generating function. The key is knowing where the ones, twos, threes, etc... start. Look at the sequence of exponents for factors of two:

{1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 6, ...}

Since every exponent is represented by its own equation, we would have to add a sum whenever a one, two, three, etc..., occurred. The following formula accomplishes this feat for any factor, n:

$$\lfloor \log_n x \rfloor + 1$$

Regarding the sequence of exponents for factors of two, we can see this formula in action:

{1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 6, ...}
{1, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, ...}

> Now we can put all of the pieces of the puzzle together, and arrive at the solution for the challenge:
>
> $$\text{Factors}(n,x) = \sum_{i=1}^{n-1} \left(\sum_{j=1}^{\lfloor \log_n x \rfloor + 1} F(n,x,i,j)\right) - 1$$
>
> numbers
> 1 2 3 4 5 6 7 8 9 10 ...
> 2 {0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0, 4, 0, 1, 0, 2, 0, 1, 0, 3, 0, 1, 0, 2, 0, 1, 0, 5}
> 3 {0, 0, 1, 0, 0, 1, 0, 0, 2, 0, 0, 1, 0, 0, 1, 0, 0, 2, 0, 0, 1, 0, 0, 1, 0, 0, 3, 0, 0, 1, 0, 0}
> 4 {0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 2, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 2}
> Factors: 5 {0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 0, 1, 0, 0}
> 6 {0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0}
> 7 {0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0}
> 8 {0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1}

Edited July 19, 2017 by Daedalus

Trurl
Quark

Senior Members
13
139 posts

Posted July 22, 2017

I have read some of your factor work. I don't admit to understand it yet. But here are my thoughts intuitively with no work yet.

I claim that my equation can tell if you are higher or lower than the actual factor. In your pattern of exponents, I would say the pattern is in the position of the factor. For example, more 1's would mean the distance from the factor is closer or further away from a product. It would tell you if you have one factor that is small and one that is larger. Or they could both be larger. This is hard for me to explain, but say you had a product of (7229 * 7757) = 56,075,353 and a difference of exponents from (7229 *7) = 50,603 your pattern would distinguish between the distance of 7757 and 7, because 7757 would be out of range of the exponent and 7 would require a smaller exponent.

I don’t know how to word this, but I think your pattern of exponents allows the distance between factors to be found. Thus, eliminating test values to a smaller range.

I did not think it would be related to my equation, but now I think they are both eliminating options. The equation relies on testing values, but I think they both speed-up the process.

I know the argument exists that it is more work than division. But I have 5 or more equations that show patterns in division. I am uncertain that combining them would improve the problem of complexity. However, I think anything that shows a pattern is significant. After all there is not an answer to the Prime factorization problem.

I thought that if the Prime numbers for an exponential range were known, you could eliminate test values. If you know that numbers raised to a specific exponent will be larger than the given N for your test factors, it may not seem useful at first, but if it were programmed into a computer it should be faster than division. It isn’t perfect because you would have to test values, but with those test values would have a theme; an organized search.

I hope this makes sense. And I hope I understood you. I will continue to read this and see if I can come up with any good brainstorms.

Oh, and let me know if my equation is useful to you eliminating factors and I will work towards understanding your work.

But if nothing else, this will be a different look at the same problem. I need a different look, I have put a lot of effort into this project, but results are difficult.

TakenItSeriously

Molecule

Posted August 9, 2017

> **On 7/19/2017 at 7:17 PM, Daedalus said:**

Senior Members
⊕ 9
468 posts
Location: Silicon Valley

> Instead of dealing with real valued functions such as wave harmonics, why not try an integer based approach. In my Seventh Challenge, I post a function that completely maps out the factors for any number. Perhaps you can use the equations.

Sorry, about taking so long before replying, but since your question involved an opinion on your own project, I was compelled to finish my own solution before looking at any other solution. It's nothing personal or ego driven, its just my way.

I just have a strong belief that for solving known problems with standard solutions, then fine, load up on the knowledge.

However, when solving unsolved problems that have been tested countless times before without success, then I feel the less I know about traditional methods or traditional thinking, the better. Thats because traditional thinking is obviously not the proper solution for that particular problem. However it could be a line of thinking that could bias my own line of thinking which I have great faith in. I'm sure far better mathematicians have tried and failed so why would I try the same math which I'd be a novice at. Especially when my gift is in logic based solutions.

And since using logic for problem solving outside of Computer science is practically unheard of in the past few centuries. My guess is my success isn't all due to my own gifts but half due to a serious lack of others trying logic based solutions. So I might be a big fish in a tiny puddle. Who knows?

Also I'm not a fan of focusing too hard on multiple lines of thinking at one time. I'm best when taking a single line of thinking them through one at a time in a methodical way. Focusing on multiple altenatives is just a terrible way to find results, at least for me. Humans don't think in parallel as far as I know. If they do, then I'm more handicapped than I realized.

So my method involves avoiding all external input like the plague until I either solved the problem or run out of ideas. At that point I'm happy to review other methods.

So now that I'm done with my excuses, I did have a chance to look at your challenge post. As far as solving primes, I was skeptical because it never addressed any of the problems common to Large Number Theory.

However, I did fint the idea that all points could take deterministic yet random like path to a single point intriguing. The only part I didn't like is the result being

1 which seems like a sooution looking for a problem that doesnt exist. At least nothing that sounds familiar to me.

If that could be modified somehow to all paths deterministicly leading to a string of 1's as in Mersenne primes, then you may really be onto something.

My ultimate goal was always to find an unbounded method for data compression.

Previously I was convinced it was connected to Mersenne Primes. More recently, however, the current solution has me looking harder at any prime, considering I have a built in hypper effecient indexing system for all primesality data. and tranforming to the closest prime number has to be a lot easier (smaller key) than transforming to the nearest MPN.

So I'm now split towards MPN or primes in general, for data transformation.

The idea is simple, Big information string plus small key transforme to either Big prime or Big MP. Either of which can be accessed from any point in spacetime using my Prime Factor Harmonic Matrix. All thats required is transmitting the small key. The practical applications are huge and future applications are revolutionary.

So if you did solve that universal deterministic path to Mersenne Primes, Then that would answer my question and we should definitely talk, as we could each have 1/2 of a really huge deal. Or I could just be wrong.

Risk/Reward seems pretty heavy on the Reward side and light on the Risk side if you ask me.

zerodrama

Lepton

New Members

● 0

1 post

Posted August 25

This is not a sieve.

It is a kind of map.

You're exposing patterns by changing the base.

Check this out

https://m.phys.org/news/2018-04-quantum-simulator-faster-route-prime.html

TakenItSeriously

Posted August 28 (edited)

Molecule

Senior Members
⊕ 9
468 posts
Location: Silicon Valley

> **On 8/25/2018 at 2:55 PM, zerodrama said:**
>
> This is not a sieve.
>
> It is a kind of map.
>
> You're exposing patterns by changing the base.
>
> Check this out
>
> https://m.phys.org/news/2018-04-quantum-simulator-faster-route-prime.html

Yes, that’s exactly right.

A seive is one dimensional which is why they always need to be started from 0 and very innefficient in terms of storage space while the matrix can be two or three dimensional and the base can be a primorial of any size. The larger the base, the more composite number collumns are segregated out and the greater the probability that the remaining numbers are either prime or semi-prime.

There are a number of huge advantages over a number sieve.

Thanks for the link. It does indeed appear to be based on the same principles though I never intended it to be used as a method for faster prime factorization.

My aim, at least in part, was to find a method for a faster primality test for discovering large primes, especially Mersenne primes.

I can see how they might succeed using quantum probability analysis to speed up the factorization process though.

Prime factors are harmonic which was the basis of the PFHM after all.

Edited August 29 by TakenItSeriously

Home > Sciences > Mathematics > Finding large Primes using Standing Wave Harmonics · All Activity

Privacy Policy · Contact Us