

Primality test. My second contribution.

Dante Servi

Abstract

This article describes a better result than that described in the article "Primality test. My contribution".
With this review I want to show how I have performed an additional 10 million checks; of which 8 with negative results.

The reference is still Fermat's little theorem.
However, I changed the base from 2 to 3; so in this article I start from $(3^a-3)/a$.

I consider $(3^a-3)/a$ as the comparison of two sequences which I call (a) and (b); it is evident that the numbers present in the sequence (b) are a function $(b=3^a-3)$ of the corresponding numbers present in the sequence (a).

As described in the previous article I have again worked on the reason for the growth of the numbers belonging to the sequence (b) starting by removing the dependence on the corresponding numbers of the sequence (a).

According to the previous verifications I made, I found a beginning and a reason for growth for the numbers belonging to the sequence (b) such that the result of b/a is always an integer if the number of the sequence (a) is a prime number and on the contrary b/a is always a fractional number if the number of the sequence (a) is a composite number.

This article is written in English and Italian, the original language is Italian which is my language, the translation into English was done using the Google translator.

The two sequence (a) and (b) from which I started; there is no doubt that $b/a=(3^a-3)/a$.

a=a+1	b=3^a-3	b/a	factor(b)
1	0	0	---
2	6	3	2x3
3	24	8	2x2x2x3
4	78	4,...	2x3x13
5	240	48	2x2x2x2x3x5
6	726	121	2x3x11x11
7	2184	312	2x2x2x3x7x13
8	6558	819,...	2x3x1093
9	19680	2186,...	2x2x2x2x2x3x5x41
10	59046	5904,...	2x3x13x757
11	177144	16104	2x2x2x3x11x11x61

In this second article, after having shown the two sequence (a) and (b) from which I started, I pass directly to the sequences (a) and (b) that I propose as the final solution.

I reserve the right to publish a new article dedicated exclusively to intermediate steps.
Now I just want to say that the intermediate steps follow the same logic described in the previous article.

In summary, the logic is to reduce the growth reason of the numbers belonging to sequence (b) as much as possible, trying to maintain as much as possible a connection with the original growth reason.

In practice, in the various passages I have always checked how the factors of the numbers of the sequence (b) changed.

Having said that, on the next page I present the first numbers of the two final sequence (a) and (b); as for the two starting sequences, I also report the results of b/a and the factors of the numbers belonging to sequence (b).

I specify that with $b=b+b+b+1$ I mean that (starting from 10) every number belonging to sequence (b) derives from the sum of the previous three numbers belonging to the same sequence (b); 1 is added to this sum.

a=a+1	b=b+b+b+1	b/a	factor(b)
2	1	-	-
3	3	-	-
4	5	-	-
5	10	2	2x5
6	19	3,...	19
7	35	5	5x7
8	65	8,...	5x13
9	120	13,...	2x2x2x3x5
10	221	22,...	13x17
11	407	37	11x37

In the sequence (b), the first three numbers are fixed and I have identified them in 1, 3 and 5; to these three numbers correspond in the sequence (a) the numbers 2, 3 and 4.

At this point I used PARI/GP to see if b/a continues to distinguish prime numbers from composite numbers, as happens in these first cases.

The following command lines for PARI/GP are used to calculate and write the numbers of the two sequence (a) and (b) and also the results (b/a) of dividing the numbers into (b), with the corresponding numbers in (a); starting from a=5 up to a=50000.

`default(log,1)` → To write the results on `pari.log`.

`a=4`

`b00=1`

`b0=3`

`b=5`

`for(i=1, 49996, a=a+1; m=b; b=b+b0+b00+1; b00=b0; b0=m; print("a=",a," ; b=",b," ; b/a=",b/a))`

It can be seen that I have improved the use of the for () function; now he only writes the results that I found useful; it also seems to me that the readability of both the commands and the results obtained has improved.

Note: The limit of 50000 can be exceeded (it should be possible to reach over 60000), in any case the `pari.log` file must not exceed the size of about 1 Gb; beyond this size Notepad fails to open it.

These are the first 20 results obtained.

`a=5 ; b=10 ; b/a=2`

`a=6 ; b=19 ; b/a=19/6`

`a=7 ; b=35 ; b/a=5`

`a=8 ; b=65 ; b/a=65/8`

`a=9 ; b=120 ; b/a=40/3`

`a=10 ; b=221 ; b/a=221/10`

`a=11 ; b=407 ; b/a=37`

`a=12 ; b=749 ; b/a=749/12`

`a=13 ; b=1378 ; b/a=106`

`a=14 ; b=2535 ; b/a=2535/14`

`a=15 ; b=4663 ; b/a=4663/15`

`a=16 ; b=8577 ; b/a=8577/16`

`a=17 ; b=15776 ; b/a=928`

`a=18 ; b=29017 ; b/a=29017/18`

`a=19 ; b=53371 ; b/a=2809`

`a=20 ; b=98165 ; b/a=19633/4`

`a=21 ; b=180554 ; b/a=180554/21`

`a=22 ; b=332091 ; b/a=332091/22`

`a=23 ; b=610811 ; b/a=26557`

`a=24 ; b=1123457 ; b/a=1123457/24`

Now I want to describe how I verified the result of (b/a) from a=5 to a=10 million; the thesis is that if b/a=integer then (a) is a prime number.

I changed the use of the print() function of the previous command lines; now the value of (a) is written to `pari.log` only if (b/a) is an integer.

The values of (a) obtained in this way must however be verified with the function `isprime(a)`; at least until the validity of b/a= integer is recognized as a primality test.

So in modifying the print() function, I expected the strings "isprime("a,")" to be written in `pari.log`.

default(log,1) → To write the results on pari.log.

```
a=4
b00=1
b0=3
b=5
for(i=1,9999996, a=a+1; m=b; b=b+b0+b00+1; b00=b0; b0=m; type(b/a) == "t_INT" && print("isprime(",a,""))
```

These are the first 10 and last 10 strings obtained.

isprime(5)	isprime(17)	isprime(31)	isprime(9999907)	isprime(9999943)
isprime(7)	isprime(19)	isprime(37)	isprime(9999929)	isprime(9999971)
isprime(11)	isprime(23)	isprime(9999889)	isprime(9999931)	isprime(9999973)
isprime(13)	isprime(29)	isprime(9999901)	isprime(9999937)	isprime(9999991)

The verification with the function isprime() of PARI/GP has shown that 8 results of $b/a = \text{integer}$ do not correspond to a prime number in the sequence (a); here are the 8 cases I found.

isprime(25201) = 0	11x29x79	isprime(2487941) = 0	911x2731
isprime(54289) = 0	233x233	isprime(3542533) = 0	1087x3259
isprime(804055) = 0	5x7x22973	isprime(3761251) = 0	13x53x103
isprime(1889041) = 0	7x11x24533	isprime(6829689) = 0	3x23x98981

Finding 8 errors out of 10 million numbers tested is enough to say that the two sequences do not work; too bad I almost believed it.

I therefore decided to delete the remainder of the article which still remains (in revision v1) available to anyone interested.

This is the link that corresponds to this article

<http://doi.org/10.5281/zenodo.6397327>

The following links correspond to my other publications on prime numbers.

Primality test. My contribution.

<http://doi.org/10.5281/zenodo.6380548>

Twin primes. But even more.

<http://doi.org/10.5281/zenodo.6227979>

Twin primes. Where they can be found.

<http://doi.org/10.5281/zenodo.5902559>

News on the mechanism of prime numbers.

<http://doi.org/10.5281/zenodo.5844231>

The mechanism of prime numbers.

<http://doi.org/10.5281/zenodo.4769674>

Finding prime numbers.

<http://doi.org/10.5281/zenodo.4786547>

Graphic representation of the mechanism of prime numbers.

<http://doi.org/10.5281/zenodo.5655072>

Goldbach's conjecture. Because I think it's true.

<http://doi.org/10.5281/zenodo.5707188>

Dante Servi

Bressana Bottarone (PV)) Italy

dante.servi@gmail.com

Test di primalità. Il mio secondo contributo.

Dante Servi

Abstract

Questo articolo descrive un risultato migliore di quello descritto nell'articolo "Primality test. My contribution".
Con questa revisione voglio mostrare come ho eseguito ulteriori 10 milioni di verifiche; delle quali 8 con esito negativo.

Il riferimento è ancora il piccolo teorema di Fermat.
Ho però cambiato la base da 2 a 3; quindi in questo articolo parto da $(3^a-3)/a$.

Considero $(3^a-3)/a$ come il confronto di due successioni che chiamo (a) e (b); è evidente che i numeri presenti nella successione (b) sono in funzione $(b=3^a-3)$ dei corrispondenti numeri presenti nella successione (a).

Come descritto nel precedente articolo ho di nuovo lavorato sulla ragione di crescita dei numeri appartenenti alla successione (b) iniziando con il togliere la dipendenza dai corrispondenti numeri della successione (a).

Stando alle precedenti verifiche da me effettuate, ho trovato un inizio ed una ragione di crescita per i numeri appartenenti alla successione (b) tali che il risultato di b/a è sempre un numero intero se il numero della successione (a) è un numero primo ed al contrario b/a è sempre un numero frazionario se il numero della successione (a) è un numero composto.

Questo articolo è scritto in Inglese ed Italiano, la lingua originale è l'Italiano che è la mia lingua, la traduzione in Inglese è stata fatta utilizzando il traduttore di Google.

Le due successioni (a) e (b) dalle quali sono partito; non ci sono dubbi che $b/a=(3^a-3)/a$.

a=a+1	b=3 ^a -3	b/a	factor(b)
1	0	0	---
2	6	3	2x3
3	24	8	2x2x2x3
4	78	4,...	2x3x13
5	240	48	2x2x2x2x3x5
6	726	121	2x3x11x11
7	2184	312	2x2x2x3x7x13
8	6558	819,...	2x3x1093
9	19680	2186,...	2x2x2x2x2x3x5x41
10	59046	5904,...	2x3x13x757
11	177144	16104	2x2x2x3x11x11x61

In questo secondo articolo, dopo aver mostrato le due successioni (a) e (b) dalle quali sono partito, passo direttamente alle successioni (a) e (b) che propongo come soluzione finale.

Mi riservo di eventualmente pubblicare un nuovo articolo dedicato esclusivamente ai passaggi intermedi.
Ora voglio solo dire che i passaggi intermedi seguono la stessa logica descritta nel precedente articolo.

In sintesi la logica è ridurre il più possibile la ragione di crescita dei numeri appartenenti alla successione (b) cercando di mantenere il più possibile un collegamento con la ragione di crescita originale.

In pratica nei vari passaggi ho sempre controllato come cambiavano i fattori dei numeri della successione (b).

Detto questo, nella pagina seguente presento i primi numeri delle due successioni (a) e (b) finali; come per le due successioni di partenza riporto anche i risultati di b/a ed i fattori dei numeri appartenenti alla successione (b).

Preciso che con $b=b+b+b+1$ intendo che (iniziando dal 10) ogni numero appartenente alla successione (b) deriva dalla somma dei precedenti tre numeri appartenenti alla stessa successione (b); a questa somma viene aggiunto 1.

a=a+1	b=b+b+b+1	b/a	factor(b)
2	1	-	-
3	3	-	-
4	5	-	-
5	10	2	2x5
6	19	3,...	19
7	35	5	5x7
8	65	8,...	5x13
9	120	13,...	2x2x2x3x5
10	221	22,...	13x17
11	407	37	11x37

Nella successione (b), i primi tre numeri sono fissi e li ho individuati in 1, 3 e 5; a questi tre numeri corrispondono nella successione (a) i numeri 2, 3 e 4.

A questo punto ho utilizzato PARI/GP per vedere se b/a continua a distinguere i numeri primi dai numeri composti, come succede in questi primi casi.

Le seguenti righe di comando per PARI/GP servono a far calcolare ed a far scrivere in pari.log i numeri delle due successioni (a) e (b) ed anche i risultati (b/a) della divisione dei numeri in (b) con i corrispondenti numeri in (a); partendo da a=5 fino ad a=50000.

default(log,1) → Per scrivere su pari.log i risultati.

a=4

b00=1

b0=3

b=5

for(i=1, 49996, a=a+1; m=b; b=b+b0+b00+1; b00=b0; b0=m; print("a=",a," ; b=",b," ; b/a=",b/a))

Si può notare che ho migliorato l'utilizzo della funzione for(); ora scrive solo i risultati che ho ritenuto utili; mi sembra anche migliorata la leggibilità sia dei comandi che dei risultati ottenuti.

Nota: Il limite di 50000 può essere superato (si dovrebbe poter arrivare ad oltre 60000), in ogni caso il file pari.log non deve superare la dimensione di circa 1 Gb; oltre questa dimensione Notepad non riesce ad aprirlo.

Questi sono i primi 20 risultati ottenuti.

a=5 ; b=10 ; b/a=2

a=6 ; b=19 ; b/a=19/6

a=7 ; b=35 ; b/a=5

a=8 ; b=65 ; b/a=65/8

a=9 ; b=120 ; b/a=40/3

a=10 ; b=221 ; b/a=221/10

a=11 ; b=407 ; b/a=37

a=12 ; b=749 ; b/a=749/12

a=13 ; b=1378 ; b/a=106

a=14 ; b=2535 ; b/a=2535/14

a=15 ; b=4663 ; b/a=4663/15

a=16 ; b=8577 ; b/a=8577/16

a=17 ; b=15776 ; b/a=928

a=18 ; b=29017 ; b/a=29017/18

a=19 ; b=53371 ; b/a=2809

a=20 ; b=98165 ; b/a=19633/4

a=21 ; b=180554 ; b/a=180554/21

a=22 ; b=332091 ; b/a=332091/22

a=23 ; b=610811 ; b/a=26557

a=24 ; b=1123457 ; b/a=1123457/24

Ora voglio descrivere come ho verificato il risultato di (b/a) da a=5 fino ad a=10 milioni; la tesi è che se b/a=intero allora (a) è un numero primo.

Ho modificato l'utilizzo della funzione print() delle precedenti righe di comando; ora viene scritto in pari.log il valore di (a) solo se (b/a) è un numero intero.

I valori di (a) ottenuti in questo modo devono però essere verificati con la funzione isprime(a); almeno fino a quando non viene riconosciuta la validità di b/a=intero come test di primalità.

Quindi nella modifica della funzione print() ho previsto che venissero scritte in pari.log le stringhe "isprime(",a,")".

default(log,1) → Per scrivere su pari.log i risultati.

```
a=4
b00=1
b0=3
b=5
for(i=1,9999996, a=a+1; m=b; b=b+b0+b00+1; b00=b0; b0=m; type(b/a) == "t_INT" && print("isprime(",a,""))
```

Queste sono le prime 10 e le ultime 10 stringhe ottenute.

isprime(5)	isprime(17)	isprime(31)	isprime(9999907)	isprime(9999943)
isprime(7)	isprime(19)	isprime(37)	isprime(9999929)	isprime(9999971)
isprime(11)	isprime(23)	isprime(9999889)	isprime(9999931)	isprime(9999973)
isprime(13)	isprime(29)	isprime(9999901)	isprime(9999937)	isprime(9999991)

La verifica con la funzione isprime() di PARI/GP ha evidenziato che 8 risultati di b/a=intero non corrispondono ad un numero primo nella successione (a); ecco gli 8 casi che ho trovato.

isprime(25201) = 0	11x29x79	isprime(2487941) = 0	911x2731
isprime(54289) = 0	233x233	isprime(3542533) = 0	1087x3259
isprime(804055) = 0	5x7x22973	isprime(3761251) = 0	13x53x103
isprime(1889041) = 0	7x11x24533	isprime(6829689) = 0	3x23x98981

Trovare 8 errori su 10 milioni di numeri testati è sufficiente a dire che le due sequenze non funzionano; peccato ci avevo quasi creduto.

Ho quindi deciso di cancellare la parte restante dell'articolo che comunque rimane (nella revisione v1) a disposizione di chi fosse interessato.

Questo è il link che corrisponde a questo articolo

<http://doi.org/10.5281/zenodo.6397327>

I seguenti link corrispondono ad altre mie pubblicazioni sui numeri primi.

Primality test. My contribution.

<http://doi.org/10.5281/zenodo.6380548>

Twin primes. But even more.

<http://doi.org/10.5281/zenodo.6227979>

Twin primes. Where they can be found.

<http://doi.org/10.5281/zenodo.5902559>

News on the mechanism of prime numbers.

<http://doi.org/10.5281/zenodo.5844231>

The mechanism of prime numbers.

<http://doi.org/10.5281/zenodo.4769674>

Finding prime numbers.

<http://doi.org/10.5281/zenodo.4786547>

Graphic representation of the mechanism of prime numbers.

<http://doi.org/10.5281/zenodo.5655072>

Goldbach's conjecture. Because I think it's true.

<http://doi.org/10.5281/zenodo.5707188>

Dante Servi

Bressana Bottarone (PV)

dante.servi@gmail.com