WRIGHT'S FOURTH PRIME

ROBERT BAILLIE

Abstract. Wright proved that there exists a number c such that if $g_0 = c$ and $g_{n+1} = 2^{g_n}$, then $\lfloor g_n \rfloor$ is prime for all $n > 0$.

Wright gave $c = 1.9287800$ as an example. This value of c produces three primes, $\lfloor g_1 \rfloor = 3$, $\lfloor g_2 \rfloor = 13$, and $\lfloor g_3 \rfloor = 16381$. But with this c, $\lfloor g_4 \rfloor$ is a 4932-digit composite number. However, this slightly larger value of c,

$$c = 1.9287800 + 8.2843 \cdot 10^{-4933},$$

reproduces Wright's first three primes and generates a fourth:

$$\lfloor g_4 \rfloor = 1913966420463110498403837302 58\ \ldots\ 3032775178002738220154174184 99$$

is a 4932-digit prime.

1. Introduction

In 1947, Mills [3] proved that there exists a number A such that

$$\lfloor A^{3^n} \rfloor$$

is prime for all $n > 0$. (Here, $\lfloor x \rfloor$ is the largest integer $\leq x$.) Mills did not give an example of such an A.

Caldwell and Cheng [1] calculate such an $A \approx 1.3063778838630806904 6$ which generates a sequence of primes that begins 2, 11, 1361, 2521008887, and 1602223620400981813183132018 3. The next prime has 85 digits. Their digits of A are in [4]. Their sequence of primes is in [6].

In 1951, Wright [8], [10] proved that there exists a number c such that, if $g_0 = c$ and, for $n \geq 0$, we define the sequence

$$g_{n+1} = 2^{g_n},$$

then

$$\lfloor g_n \rfloor$$

is prime for all $n > 0$.

This sequence grows much more rapidly than Mills' sequence.

The key ingredient in Wright's proof is the relatively elementary fact that, for every $N \geq 2$, there is a prime between N and $2N$.

Wright gave an example of such a constant: $c = 1.9287800$. This value of c produces three primes, $\lfloor g_1 \rfloor = \lfloor 3.8073\ldots \rfloor = 3$, $\lfloor g_2 \rfloor = \lfloor 13.9997\ldots \rfloor = 13$, and $\lfloor g_3 \rfloor = \lfloor 16381.3640 \rfloor = 16381$.

But with this value of c, $\lfloor g_4 \rfloor$ is a 4932-digit composite number,

$$\lfloor g_4 \rfloor = \lfloor 2^{2^{2^{2^c}}} \rfloor = 19139664204631104\ \ldots\ 822015417386540.$$

In 1954, Wright [11] proved that the sets of values of such A and c have the cardinality of the continuum, are nowhere dense, and have measure 0.

Date: June 6, 2017.

2010 *Mathematics Subject Classification.* Primary 11A41; Secondary 11A51.

Key words and phrases. prime numbers, prime-representing function.

Neither Mills' nor Wright's formula is useful in computing primes that are not already known.

In the next section, we show how to compute a value of c slightly larger than Wright's constant, which causes $\lfloor g_4 \rfloor$ to be a prime. Our modified value preserves all of Wright's original seven decimal places.

In a leter section, we discuss another modification of c that gives a different fourth prime.

2. Modifying Wright's Constant to Produce Four Primes

The calculations here were done in *Mathematica*. Snippets of *Mathematica* code are scattered throughout this paper, in a font that looks like this:

```
2 + 3
```

In *Mathematica*, if we specify a floating-point value such as 1.9287800, *Mathematica* assumes this number has only machine precision. Therefore, it is better for calculations like ours to use the *exact* value of Wright's constant:

```
c = 1 + 92878/10^5
```

With this c, we can verify that the integer parts of g_1, g_2, and g_3,

```
Floor[2^c]
Floor[2^(2^c)]
Floor[2^(2^(2^c))]
```

are Wright's three primes 3, 13, and 16381.

However, the integer part of g_4 is composite:

```
n4 = Floor[2^(2^(2^(2^c)))] ;
PrimeQ[n4]
```

(The trailing semicolon suppresses (lengthy) output that we don't need.) `PrimeQ[n4]` returns `False`, so $n4$ is not prime. `PrimeQ[ ]` is a probable prime test; see [9] and [12].

The *Mathematica* command `N[n4]` shows that $n4$ is about $1.913966420463110 \cdot 10^{4931}$. The last 10 digits of $n4$ can be found with `Mod[n4, 10^10]`. These digits are 5417386540, so we can see that $n4$ is not prime.

We can use *Mathematica*'s `NextPrime` function to locate the first (probable) prime larger than $n4$:

```
prp4 = NextPrime[n4] ;
diff = prp4 - n4
```

$prp4$ is in Appendix A. The first and last 35 digits of $prp4$ are

$$19139664204631104984038373025808682 \ \ldots \ 26398303277517800273822015417418499 \, .$$

This difference `prp4 - n4` is 31959, which is small compared to $n4$. This suggests that we can compute a new starting value for the sequence, say, $g_0 = w$, where w is only slightly larger than c, which makes $\lfloor g_4 \rfloor = prp4$, and which reproduces Wright's first three primes.

Because $prp4$ is computed with the `Floor` function, this value of w must satisfy the inequalities

$$prp4 \le 2^{2^{2^{2^w}}} < prp4 + 1 \, .$$

Solving the inequalities for the minimum and maximum possible values of w,

```
wMin = Log[2, Log[2, Log[2, Log[2, prp4 + 0]]]] ;
wMax = Log[2, Log[2, Log[2, Log[2, prp4 + 1]]]] ;
```

If we attempt to see how much larger `wMin` and `wMin` are compared to c, we find that `N[wMin - c, 20]` gives $0. \cdot 10^{-70}$ and the warning,
"`Internal precision limit $MaxExtraPrecision = 50. reached` ..." .

To remedy this, we use the `Block` structure to do the calculation with plenty of added precision inside the `Block`:

```
Block[ {$MaxExtraPrecision = 6000}, N[wMin - c, 20] ]
Block[ {$MaxExtraPrecision = 6000}, N[wMax - c, 20] ]
```

These differences are

$$wMin - c \approx 8.28423705953245085 41 \cdot 10^{-4933}$$
$$wMax - c \approx 8.28449628180367196 50 \cdot 10^{-4933} .$$

Any number between these two values, for example, $8.2843 \cdot 10^{-4933}$, when added to c, should produce the value $prp4$. Note that $8.2842 \cdot 10^{-4933}$ is too small and that $8.2845 \cdot 10^{-4933}$ is too large. Also, values with four or fewer significant digits, like $8.284 \cdot 10^{-4933}$, $8.28 \cdot 10^{-4933}$, or $8.29 \cdot 10^{-4933}$, are either too small or too large.

As above, we should use the *exact* value of $8.2842 \cdot 10^{-4933}$ in our calculations. The value of w that should produce $prp4$, is

$$w = c + (8 + 2843/10^4) \cdot 10^{-4933} .$$

A quick check in *Mathematica* verifies this:

```
w =  (1 + 92878/10^5) + (8 + 2843/10^4) * 10^-4933 ;
prp4a = Floor[2^(2^(2^(2^w)))] ;
prp4a - prp4
```

This difference is 0. We can also check that this w gives Wright's first three primes:

```
{ Floor[2^w] , Floor[2^(2^w)] , Floor[2^(2^(2^w))] }
```

These three values are 3, 13, and 16381.

Note: $prp4$ is not the *closest* probable prime to $n4$. Let's search for probable primes just less than $n4$:

```
previousPrp = NextPrime[n4, -1]
n4 - previousPrp
```

returns 129, so this is closer to $n4$ than was $prp4$. (The negative second argument to `NextPrime` causes *Mathematica* to search for the largest probable prime less than $n4$.)

Like we did above, we can compute the value of g_0 which starts a sequence such that $g_4 = n4 - 129$. The value we get is

$$c - (3 + 35/10^2) \cdot 10^{-4935} \approx 1.9287799999 \cdots .$$

Unfortunately, this changes the last two of Wright's original decimal places.

If you don't have *Mathematica*.

The *Wolfram Alpha* website http://www.wolframalpha.com can do some of the calculations shown here. First, put the entire calculation into one expression, such as

```
PrimeQ[ Floor[ 2^(2^(2^( 2^(1 + 92878/10^5) ))) ] ]
```

and paste it into that webpage. On the main Wolfram Alpha page, this returns no (that is, the number is not prime). This works because we are computing `PirmeQ` of an even number, so this takes very little time to evaluate.

However, *prp*4 is probably prime, so `PrimeQ[prp4]` takes a while to compute. So, this expression for *prp*4

```
PrimeQ[ Floor[ 2^(2^(2^(2^(1 + 92878/10^5 + (8 + 2843/10^4) * 10^-4933)))) ] ]
```

remains unevaluated. But if you select "Open code", then press the little button with the arrow to evaluate it, *Wolfram Alpha* will return `True`.

3. The Fifth Term in Wright's Sequence

What can we say about the *fifth* term in Wright's sequence?

The fourth term in Wright's sequence is

$$g_4 = 2^{2^{2^{2^w}}} .$$

The fifth term,

$$g_5 = 2^{g_4}$$

is too large for *Mathematica* to compute directly. It would be even harder to adjust g_5, like we did above with g_4, to produce a 5^{th} prime.

However, we can use base 10 logarithms to calculate how many digits g_5 has, and even to calculate the first few of those digits.

Suppose L is the base 10 logarithm of g_5, that is,

$$L = \log_{10} g_5 = \log_{10} 2^{g_4} = g_4 \cdot \log_{10} 2 .$$

Then $g_5 = 10^L$. Let k be the integer part of L, that is, $k = \lfloor L \rfloor$, and let f be the fractional part of L, that is, $f = L - k$. Then f is between 0 and 1, and

$$g_5 = 10^L = 10^{f+k} = 10^f \cdot 10^k .$$

The factor 10^k determines only how many digits there are, not *what* those digits are. f determines the digits of g_5.

Here's some *Mathematica* code. We'll define c and w again here to make this code be self-contained.

```
c = 1 + 92878/10^5 ;
w = c + (8 + 2843/10000) * 10^-4933 ;
g4 = 2^(2^(2^(2^w))) ;  (* about 10^4931 *)
capL = g4 * Log[10, 2] ;  (* log base 10 of g5 *)
(* convert L from exact expression to a numerical approximation *)
capL = N[capL, 5100] ; (* compute this to 5100 digits *)
N[capL]
```

The result is $L \approx 5.761613032530158 \cdot 10^{4930}$. Next, extract the integer and fractional parts of L and display rough approximations to the much more accurate values that are stored internally.

```
k = Floor[capL] ;
f = capL - k ;  (* f is between 0 and 1 *)
{ N[k] , N[f] }
```

The results are $k \approx 5.761613032530158 \cdot 10^{4930}$, and $f \approx 0.776988577922$. k is a very large integer, having 4931 digits. The first few digits of k are 5761613032. The last few digits of k can be obtained from `Mod[k, 10^10]`; they are 8933273637. (Or, we could just display k itself to see *all* of its 4931 digits.)

So, $k = 5761613032\ldots8933273637$. The *number of digits* in g_5 is the 4931-digit number

$$k + 1 = 5761613032\ldots8933273638\,.$$

We can also obtain the first few digits of g_5 itself.

$$g_5 = 10^f \cdot 10^k = 10^f \text{ times (a large power of 10)}\,. \tag{3.1}$$

The digits of g_5 come from 10^f. Here are two approximations to 10^f:

```
{ N[10^f] , N[10^f, 20] }
```

These approximations are 5.98395856859 and 5.9839585685895397357. The "large power of 10" in Equation (3.1) just moves the decimal point over. Therefore, g_5 begins with the digits 5983958568.

These are the first few digits of g_5, the fifth *term* in Wright's sequence. What about the first few digits of the fifth *prime* in Wright's sequence? The reader may wonder if the leading digits we just computed would have to be changed if g_5 were not prime, but like g_4, its value had to be adjusted to obtain a prime or a probable prime.

The answer is "no." Given the large size of g_5, we only need to make a proportionately small increment in g_5 in order to reach the next prime greater than g_5. Dusart [2, Proposition 6.8] shows that, for any $x > 396738$, there is a prime p in the interval

$$x < p < x\left(1 + \frac{1}{25(\ln x)^2}\right)\,.$$

Therefore, for values of x near g_5, we would need to increase x by only a tiny fraction of x to reach the next prime larger than x. Unless x was very slightly below a power of 10 (which g_5 is not), we can do this without changing the first few leading digits of x.

For $x = g_5$, we have $\ln x = \ln g_5 = \log_{10} g_5 \cdot \ln 10$. This is `capL * Log[E, 10]`, which is about $1.32666 \cdot 10^{4931}$, so the fraction

$$\frac{1}{25(\ln x)^2} \approx 2.3 \cdot 10^{-9864}$$

in Dusart's estimate means that g_5 and the next prime larger than g_5 have ≈ 9800 leading digits that are the same.

4. Another Version of Fourth Term in Wright's Sequence

As mentioned above, Wright later proved that his original value, 1.9287800, is not the only one that works.

In OEIS [5], Charles Greathouse defines the sequence:

$$a_0 = 1\,,$$
$$a_n = \text{greatest prime } < 2^{a_{n-1}+1}\,.$$

Wright does not say anything about a "greatest prime ...", so Greathouse's formulation is slightly different from Wright's.

The first three terms in Greathouse's sequence match Wright's three primes $a_1 = 3$, $a_2 = 13$, and $a_3 = 16381$. In Greathouse's sequence, a_4 is the 4932-digit probable prime,

$$q = 2^{16382} - 35411 = 29743287383930794127\ldots11756822667490981293\,.$$

q is roughly $prp4 \cdot 1.554$, so it is much larger than $prp4$.

We can transform Greathouse's a_1, a_2, a_3 and a_4 into a sequence of the form proposed by Wright. That is, we can find a z such that

$$q = \lfloor 2^{2^{2^{2^z}}} \rfloor .$$

We work backwards from q to estimate z, just as we did above:

```
zMin = Log[2, Log[2, Log[2, Log[2, q + 0]]]] ;
zMax = Log[2, Log[2, Log[2, Log[2, q + 1]]]] ;
{ N[zMin, 20] , N[zMax, 20] }
Block[{$MaxExtraPrecision = 6000}, N[zMax - zMin, 20] ]
```

`zMin` and `zMax` are both about 1.928782187150216, which is about $c + 2.187150216\ldots \cdot 10^{-6}$.

The difference `zMax - zMin` is about $1.6680090447391719120 \cdot 10^{-4937}$. The fact that `zMax` and `zMin` are so close together means that, in order to get q as the fourth term in the sequence, we must specify z to at least 4937 decimal places.

So, a value of z that produces q is

$$z = 1.9287800 + 2.187150216\ldots \cdot 10^{-6} \approx 1.928782187150216\ldots .$$

We can verify that this z reproduces Wright's first three primes.

The probable prime q has a form that is easy to write down, which is a very nice feature. However, z is *not* easy to write. In addition, this q leads to a z whose 6^{th} and 7^{th} decimal places are different from Wright's.

5. Proof That the 4932-digit Prp is Prime

The proof of primality of *prp*4 was kindly carried out by Marcel Martin, the author of the software *Primo*.

Mr. Martin has supplied the primality certificate as a 1.5 megabyte text file. It has been uploaded as an ancillary file to Math arXiv along with the LaTeX for this pdf, and so is publicly available. The file (converted to use PC-style end of line characters) is `P4932Proof.txt`. The link to the file may be found at https://arxiv.org/abs/1705.09741.

The *Primo* website [7] allows one to download a Linux version of *Primo*. Inside this compressed file is the file `verifier-f4.txt`, which explains the format of the certificate file. Excerpts of the primality certificate are in Appendix B.

References

[1] Chris K. Caldwell and Yuanyou Furui Cheng, *Determining Mills' Constant and a Note on Honaker's Problem*, Journal of Integer Sequences, vol. 8 (2005), Article 05.4.1 https://arxiv.org/abs/1010.4883v2 .

[2] Pierre Dusart, Estimates of Some Functions Over Primes Without R.H. https://arxiv.org/abs/1002.0442

[3] W. H. Mills, *A prime-representing function*, Bulletin of the American Mathematical Society, vol. 53, no. 6 (1947) p. 604. Available at https://dx.doi.org/10.1090/S0002-9904-1947-08849-2 .

[4] OEIS Foundation, The On-Line Encyclopedia of Integer Sequences, Sequence A051021, https://oeis.org/A051021 .

[5] Charles Greathouse's version of Wright's Constant, OEIS Foundation, The On-Line Encyclopedia of Integer Sequences, Sequence A016104, https://oeis.org/A016104 .

[6] OEIS Foundation, The On-Line Encyclopedia of Integer Sequences, Sequence A051254, https://oeis.org/A051254 .
[7] Primo, http://www.ellipsa.eu/public/primo/primo.html
[8] *Formula for primes - Wright's formula*, Wikipedia, https://en.wikipedia.org/wiki/Formula_for_primes#Wright.27s_formula .
[9] *Probable prime*, Wikipedia, https://en.wikipedia.org/wiki/Probable_prime
[10] E. M. Wright, *A prime-representing function*, American Mathematical Monthly, vol. 58, no. 9 (November, 1951) pp. 616–618.
Available through JSTOR at http://www.jstor.org/stable/2306356
or from sci-hub at http://www.jstor.org.sci-hub.io/stable/2306356 .
[11] E. M. Wright, *A class of representing functions*, J. London Math. Soc. vol. 29 (1954), pp. 63–71, DOI 10.1112/jlms/s1-29.1.63 .
Available from sci-hub at http://sci-hub.io/10.1112/jlms/s1-29.1.63
[12] Wolfram Research, *Some Notes on Internal Implementation*, http://reference.wolfram.com/language/tutorial/SomeNotesOnInternalImplementation.html#6849

Email: rjbaillie@frii.com
State College, PA 16803

Appendix A. Wright's 4932-digit prime

The first 49 lines have 100 digits each; the 50^{th} line has 32 digits.

1913966420463110498403837302580868256924068401302629071247047560451589953807435264854392127830031342
3720949605721845025408541416289929256457498154990879565082392381927933483828466923960616991247583802
9883619110692151423464455379009608955465329715762183525752181460161562758974828177320048995147265873
2612842019664152281348948186398732693179381636809020721953435180587581344583081883196010622609758369
7767679075913848908389442791706899766927582774282426822260152187401770387233733171089048849946028924
5157524523119653473649546027804630790081490847531422093148584816528706829028167565311356355769106236
3436320845234403098854218594788259973815323356158099262195691733448838182334266034192071091676439686
2265510565742437869510668830397269397683367735888705803587084949626067429633365796074180406455970912
6668629783145246115021331298258625391024527249386102804251972165845252223261489931721363858670767105
4248335979372860336346520880575277778195467928472906091523630043216007514518228150815939447637411344
7712964754879402278403788872581553350618216740637465014766709739600860598424027452722397314233461324
3554366511854496799080845751134901098644519782916179343298293778573716056985606298935133474382630689
78845392196306713516645882632302622074827056197334511601317245554940671642976964524514177738197270571
6260612525087797573837427738480872744295214490072136635685672009176364146942578052383727557842247752
0489235482698668059234538429338152975120991845467698753576138403308689515770500766600367307264278129
6385199828644069340857669387939933468049070293624684340733033979908337085150906551694480809932331078
4299823276664647874098243326085316925154595386244016672887268180889840689362868533728799717395246308
0696821574733303165983255851113959313600965591108237415113344253763706010736391497450693583518788498
9316240962578111378784767345090298474993211780832777310463314442498353207880211862181892505036969192
6358036293250638428178824929430101370102857291008926965268198566945315354949835108825630767446425352
8255517056088310108344188100033286300440272304204667045844782263635047937424899000008973802858903851
9748379376099891646854634960901972506408778897357053934891789555018010603618963530602596099183804781
0360978482444901802367043983445091002616257070484353735336147322263371079906065560935578380773451803
1173070669642004942453790987585636362653350652338065854739543791298263679254019831557724763818029721
3219709231135045411837760886872911184087847453476473287761421955339776082963457215653389473532068042
2896599568484948987980124474303061131986362734512646340720392969482112940602526373872206989443058193
1620141152155506003013424587420882071027258150198366577849094713484227826337925082326126175462465106
6868554119469144757936097242081831537989150352203809462200094995640266444226124488319524815681306603
0484157312049475657455891806365572112483013296726703660139554346800216828207331697036722386869392311
7292728546281362276533617800058376140638880594510851920336646467207029946648186516349117729671239742
5307352469839177272906801170012224650004900465089338396056769817737441550570475001251265423682386679
7945373320921735535070727011460716567753315821711151709577072011256921210645071144106769656232777967
9712143026787946775552002385781696535492852451410632829991646997299282058508493824870685216851349891
8837841957503924236096057526988025206369863841766041242781123259746568355235428110056821962067490859
7630720929036339268766145451467553499079094345628826630593691489913445697457355771973676182806709095
5822476263514167129358028644201899398196479955267365540315495270204454613648164354927715315704906160
7842283189320789528849092970997530950569199259664184338490237020981512354685212233644530571861380675
9476152236368752591110037492477330873880509706164240194120108979685832705949794804010107552166822371
8800333019592503026994765628237259041892766899605886737660022733611133263300330029814813332221700392
9138019039259725692903265121015384141844579067125197202876330274110854956458327458736535750234264686
2512351569657042171007843392151231622234642402206399830780627559005632810029705079330184748085168278
7891711142260164705330094716896433094972273104545975536507425228377448041335380637723281310588466416
6076451106098414524772889596613043904939811847731404498273115078750425281713654707217389503796320339
6772643838198916813637341158650996525326962288014806047148707699365436502651859563251338295880787872
3977512952742511645979138811315043112958677288227319515070476476280954007473543924297078456157036816
2775502659204072125435057943997343367555279443817064367795616277338252366227141309865186313329020191
1338004574944784907370433680531046924845370160482308921524150821680599425722426069993404921595292153
7941880077357717639937345624965111126437199559302572359749060529914946957249528750108643997547579130
9794685593260532036334783066721091921856176171076768710355454274734242608506330432587215831882651263
98303277517800273822015417418499

Appendix B. Excerpts of the Primality Certificate

Below is an abbreviated version of the primality certificate file produced by Marcel Martin's *Primo* program.

```
[PRIMO - Primality Certificate]
Version=4.2.1 - LX64
WebSite=http://www.ellipsa.eu/
Format=4
ID=B3CC803D5740A
Created=May-7-2017 05:52:04 PM
TestCount=540
Status=Candidate certified prime

[Comments]
Put here any comment...

[Running Times (Wall-Clock)]
1stPhase=20253s
2ndPhase=6336s
Total=26588s

[Running Times (Processes)]
1stPhase=156839s
2ndPhase=50655s
Total=207494s

[Candidate]
File=/home/primo64/work/Baillie4932.in
N=$292F...0303
HexadecimalSize=4096
DecimalSize=4932
BinarySize=16382

[1]
S=$12
W=$B979...FD56
J=$4FE3...51F2
T=$2

[2]
S=$39330122B
W=$12FA...F0F5
J=$A137...006F
T=$1

...

[537]
S=$5CB304
W=-$AD87C590F88A80304CA
A=$2
B=0
T=$3
```

[538]
S=$5A
Q=$3

[539]
S=$379
W=-$19D0C7AD9EA43AB5
A=0
B=$3
T=$1

[540]
S=$5E0A0257FA10
B=$2

[Signature]
1=$06DE9AC57B1F53C2FD64648659604AEF1531E97C9871A932
2=$C5858EFD6BB8AADF1C00EA4A566005740310857178FBCE25