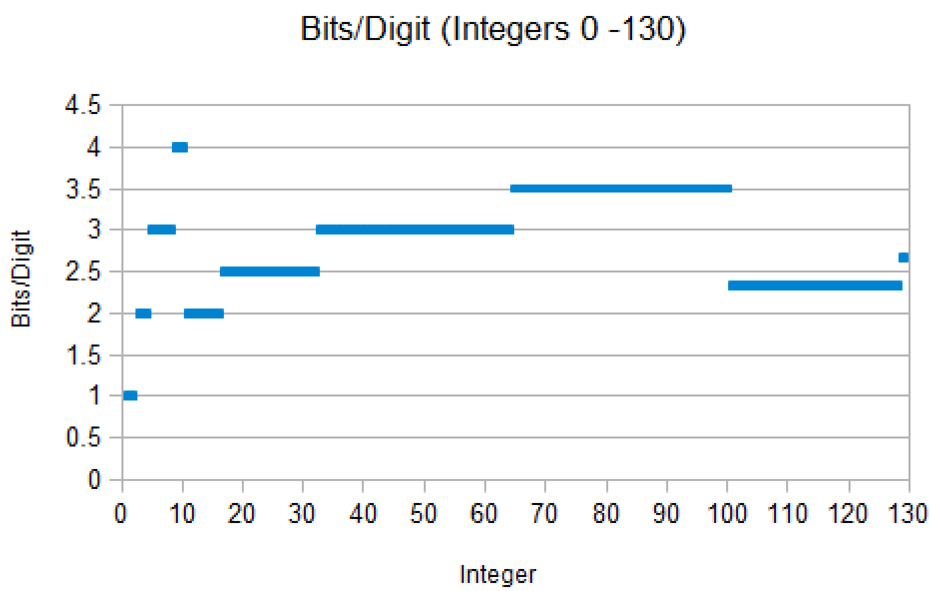
Exploring Binary

Ratio of Bits to Decimal Digits

By Rick Regan January 31st, 2013

Excluding 0 and 1, it takes more digits to express an integer in binary than in decimal. How many more? The commonly given answer is $\log_2(10) \approx 3.32$ times as many. But this is misleading; the ratio actually depends on the specific integer. So where does '$\log_2(10)$ bits per digit' come from? It's a theoretical limit, a value that's approached only as integers grow large. I'll show you how to derive it.



Bits Per Digit Varies with the Integer's Value

Bits/Digit

To represent an integer *n* in binary, $\lfloor log_2(n) \rfloor + 1$ bits are required; to represent an integer *n* in decimal, $\lfloor log_{10}(n) \rfloor + 1$ decimal digits are required. The ratio of bits to digits is thus

$$\frac{\lfloor log_2(n) \rfloor + 1}{\lfloor log_{10}(n) \rfloor + 1}$$

The graph above shows this ratio for *n* equal 0 through 130. As expected, the ratio varies; it zig-zags, changing at power of two and power of ten boundaries. When you cross a power of two, the number of bits goes up, so the ratio goes up; when you cross a power of ten, the number of decimal digits goes up, so the ratio goes down. This variation continues forever — but does the ratio ever converge? Let's take a look.

Evidence That the Ratio Converges to $\log_2(10)$

First let's compute the ratio for a few small integers (I used PARI/GP to compute them):

Bits Per Digit (Examples 1)

Decimal Number	Ratio	Bits Per Digit
1	1/1	1
3	2/1	2
8	4/1	4
30	5/2	2.5
780	10/3	approx 3.333
2012	11/4	2.75

That didn't tell us much; let's try bigger numbers:

Bits Per Digit (Examples 2)

Decimal Number	Ratio	Bits Per Digit
19,888,377	25/8	3.125
14,578,757,201	34/11	approx 3.091
98,456,378,461,883	47/14	approx 3.357

Interesting. Those are all over 3.0. Let's try even bigger numbers (for simplicity I'll switch to just using powers of ten):

Bits Per Digit (Examples 3)

Decimal Number	Ratio	Bits Per Digit
10^{50}	167/51	approx 3.275
10^{60}	200/61	approx 3.279

10^{120}	399/121	approx 3.298
10^{520}	1729/521	approx 3.317
10^{20000}	66439/20001	approx 3.322
$10^{1000000}$	3321929/1000001	approx 3.322
$10^{10000000}$	33219281/10000001	approx 3.322

Now that looks like it's converging (at least to 3 places). It turns out it does *converge* — I'll prove it.

Proof That the Ratio Converges to $\log_2(10)$

(My proof is adapted from these two proofs: Appendix O: "The Ratio of Decimal To Binary Digits" [sic] from Gerald R. Rising's book "Inside Your Calculator", and "Fun With Math Volume 2" by Peter M. Maurer.)

We want to show what happens to the ratio as the integers *n* get bigger. In calculus terms, we are looking for its *limit*, which is shown with the limit notation, $\lim\limits_{n\to\infty}$. This is what we want to find:

$$\lim_{n\to\infty} \frac{\lfloor log_2(n) \rfloor + 1}{\lfloor log_{10}(n) \rfloor + 1}$$

The floor notation makes this hard to manipulate. Fortunately, we can just get rid of it — the fractional part of the logarithms become insignificant as *n* grows large:

$$\lim_{n\to\infty} \frac{log_2(n) + 1}{log_{10}(n) + 1}$$

Similarly, we can remove the '+ 1' from the numerator and denominator, since that becomes insignificant as well:

$$\lim_{n\to\infty} \frac{log_2(n)}{log_{10}(n)}$$

Now we have something we can work with. Let's change the base of one of the logarithms so all will be in the same base; let's chose the denominator:

$$\lim_{n \to \infty} \frac{log_2(n)}{\left(\frac{log_2(n)}{log_2(10)}\right)}$$

This simplifies to

$$\lim_{n \to \infty} log_2(10)$$

But notice that *n* is gone, so there's no need for the limit notation. This means the limit reduces to this constant:

$$log_2(10) \approx 3.32$$

This says, as *n* grows large, there are approximately 3.32 times as many bits as there are decimal digits. (This doesn't mean that each decimal digit requires approximately 3.32 bits; it's just an average across all digits.)

The Minimum and Maximum Ratios

If you look at the graph you'll see a ratio of one for integers 0 and 1. This is the minimum ratio, since for all other integers, more bits than decimal digits are required. How about the maximum ratio? The graph shows a ratio of four, for integers 8 and 9. Is four the maximum? Does it occur anywhere else? The answers are "yes it is the maximum" and "no it does not occur anywhere else", but I don't have a proof. Any takers? (*Update*: I had one — see the comments below.)

Binary Vs. Octal and Hexadecimal

You can apply the above analysis to representations in any pair of bases. How about the ratio of bits to octal digits? This works out to be $log_2(8) = 3$. Similarly, the ratio of bits to hexadecimal digits is $log_2(16) = 4$. Of course, this is not surprising, since everyone "knows" there are 3 bits per octal digit and 4 bits per hexadecimal digit. But remember that I'm talking about pure mathematical representations, not fixed-sized computer representations with explicit leading zeros. For example, 17 octal = 1111 binary, a ratio of 4/2 = 2; 1f hexadecimal = 11111 binary, a ratio of 5/2 = 2.5.

Number of Bits in a Decimal...

What a Binary Counter Looks and Sounds...

Binary Subtraction

Floating-Point Will Still Be Broken In...

An Hour of Code... A Lifelong...

Binary Addition

My Fascination with Binary Numbers

Get articles by RSS (What Is RSS?)

Get articles by e-mail

Binary numbers / Binary integers

5 comments

1. **sachse**

   January 31, 2013 at 7:53 pm

   > The answers are "yes it is the maximum"
   > and "no it does not occur anywhere else",
   > but I don't have a proof. Any takers?

   Somehow this way, maybe:

With N >= 2 the following is true:
1.6^N > 2
16^N > 2 * 10^N
2^(4N – 1) > 10^N
It means any 4N-digit binary number (which is always greater-or-equal than 2^(4N – 1)) is always greater than any N-digit decimal number. So, binary for the same number can't be 4 times longer anymore.

---

2. **Rick Regan**

February 1, 2013 at 12:54 pm

@sachse,

That looks pretty good.

Not that it changes the result but I think you meant to say "any 4N-digit binary number… is always greater than any N**+1** digit decimal number" (10^N has N+1 digits.)

(I was already using a similar '1.6 * 10^n' idea in the next article I'm writing — about properties of powers of two and powers of ten crossings — but I failed to see the connection to a proof here.)

Thanks for your response.

---

3. **rtoal**

March 8, 2015 at 4:26 pm

The proof is easier if you use ceilings. Let B(n) be the bit length and D(n) be the decimal length. B(n) = ceil(log2(n+1)) and D(n) = ceil(log10(n+1)).

Now all you need is:

$$\begin{eqnarray}
B(n) &=& \left\lceil{\log_2(n+1)}\right\rceil \nonumber \\
&=& \left\lceil{\log_2(10) \times \log_{10}(n+1)}\right\rceil \nonumber \\
&\leq& \left\lceil{\log_2(10)}\right\rceil \times \left\lceil{\log_{10}(n+1)}\right\rceil

\nonumber \\
&=& 4 \times \left\lceil{\log_{10}(n+1)}\right\rceil \nonumber \\
&=& 4 D(n) \nonumber
\end{eqnarray}$$

---

4. **Rick Regan**

March 8, 2015 at 8:37 pm

@rtoal,

Here is your proof, translated from LaTeX (which is not formatted in comments):

```
B(n) = ceil(log2(n+1))
     = ceil(log2(10) * log10(n+1))
     = ceil(log2(10)) * ceil(log10(n+1))
     = 4 * ceil(log10(n+1))
     = 4 * D(n)
```

Please explain what you are trying to show.

---

5. **GOLAK**

November 3, 2015 at 2:53 am

Nice explanation…!!!

Comments are closed.