Math 189H Joy of Numbers Activity Log

Tuesday, September 20, 2011

Richard Feynman: "There are 10^{11} stars in the galaxy. That used to be a huge number. But it's only a hundred billion. It's less than the national deficit! We used to call them astronomical numbers. Now we should call them economical numbers."

Eric Temple Bell: "Any impatient student of mathematics or science or engineering who is irked by having algebraic symbolism thrust upon him should try to get along without it for a week."

$438,579,088 = 4^4 + 3^3 + 8^8 + 5^5 + 7^7 + 9^9 + 0^0 + 8^8 + 8^8$, except that this is false! If we use $0^0 = 1$, then the right side of the equation is odd... But $3435 = 3^3 + 4^4 + 3^3 + 5^5$. [The book *The Penguin Dictionary of Curious and Interesting Numbers* claims that both are true, and that they are the only such numbers.]

Our question from last time, *How fast is the Euclidean algorithm?*, was the basis for most of our discussion. From the point of view of finding the gcd of two numbers, the EA (to use an not terribly common abbreviation, for ease of writing) repeatedly computes the remainder upon division of the previous remainder into the remainder just prior to it. Since the remainder must be smaller than the divisor, we can guarantee that it is at least one smaller than the previous, one, so we have to get to the gcd (which is bigger than 0), in no more than b long divisions, where b is the smaller of the two numbers that we started with, in computing $\gcd(a, b)$. But can we do better than that?

If we imagine the the range of sizes of the remainder when we divide b into a, we know it will lie in the range 0 to $b - 1$, so the 'average' size will be $b/2$, approximately. As we continue down the EA, successive remainders will, we expect, be on average half the size of the previous remainder. Some will be larger and some smaller; the larger ones will, we would think, slow down the process of reaching the gcd, while the smaller ones will speed this up. But we sort of expect remainders to drop by about a half each time. Since $2^{10} = 1024 \approx 10^3$, this would mean that 10 "turns of the crank" (i.e., 10 long division and remainder calculations) would result, typically, in remainders have 3 fewer digits than at the start. So a gcd computation of two 50-digit numbers (about 17 3's) would take around 170 long divisions to finish, typically.

This is a fairly rough estimate, based on the idea that successive remainders are half the size of their predecessor. But we know from our own experience that this is not always the case! We could cook up example in class where the next remainder was as bad as it could be:

$$21 = 11 \cdot 1 + 10 \qquad 114 = 37 \cdot 2 + 36, \qquad \text{etc.}$$

But what we noticed was that in these 'bad' cases, the next remainder was good! (In the two cases above, it was 1...). And in general, we noticed that if the n-th remainder r_n was 'bad', i.e., $r_n > r_{n-1}/2$, then in the next long division, we could figure out exactly what would happen. The number, r_n, which we are dividing into r_{n-1} is more than half of r_{n-1}, so it will go in exactly once, with remainder $r_{n-1} - r_n$:

$$r_{n-1} = r_n \cdot 1 + (r_{n-1} - r_n)$$

so $r_{n+1} = r_{n-1} - r_n$. But! If r_n is more than half of r_{n-1}, then $r_{n-1} - r_n$ is <u>less</u> than half of r_{n-1}. So even with 'bad' (i.e., too large) remainders in our EA computation, we know that <u>two</u> turns of the crank must drop the remainder by at least half. [With more work, we could probably do even better.] But this tells us that <u>at worst</u>, we could need no more than twice the number of turns of the crank that our 'typical' case analysis suggested. that is, 20 crank turns will guarantee a remainder that is at least a factor of $2^{10} \approx 10^3$ smaller, i.e., with 3 fewer digits. So the gcd of two, say, 12-digit numbers can be computed using EA and will involve no more than $20 \cdot 4 = 80$ long division calculations.

This might still sound like a lot, but in the world of number-theoretic calculations, it is blazingly fast. Compare this, for example, to our current best method for testing a 12-digit number n for primality; we have to test divide n by all of the primes up to 10^6, of which there are, it turns out, $78,498$.

It happens to be a fact that the worst performance, relative to their size, of the EA takes place when our numbers a and b are consecutive terms of the *Fibonacci sequence*

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, \ldots$$

The pattern (the fancy term is 'recursion') to the sequence is that each term is the sum of the previous two (starting with 1 and 1), so $r_{n+1} = r_n + r_{n-1}$. The point is that r_n divided into r_{n+1} goes in once, with remainder r_{n-1} (i.e., $r_{n-1} = r_{n+1} - r_n$), and so an EA calculation will write down all of the sequence from r_n back to 1 as our remainders. Which, it turns out, uses more long divisions than any other numbers of a size comparable to r_n.

The Fibonacci sequence, or rather, the numbers in it, occur in a myriad of different contexts throughout mathematics, the sciences, and, most strikingly perhaps, in nature. This passage is lifted directly from Wikipedia:

Fibonacci sequences appear in biological settings, in two consecutive Fibonacci numbers, such as branching in trees, arrangement of leaves on a stem, the fruitlets of a pineapple, the flowering of artichoke, an uncurling fern and the arrangement of a pine cone. In addition, numerous poorly substantiated claims of Fibonacci numbers or golden sections in nature are found in popular sources, e.g., relating to the breeding of rabbits, the spirals of shells, and the curve of waves. The Fibonacci numbers are also found in the family tree of honeybees.

The Wikipedia entry for 'Fibonacci number', as we discovered, has a wonderful photo of patterns of 13 and 21 appearing in the head of a Yellow Chamomile. There is even an entire journal, the *Fibonacci Quarterly* devoted to the scholarly study of Fibonacci numbers. One of those 'poorly substantiated' claims relates to the story of Fibonacci's discovery of the sequence: if you suppose that a rabbit can give birth to another rabbit after a year and every year after that, then the number of rabbits you will have after n years is r_{n+1} (of the r_n rabbits you have, r_{n-1} are old enough to give birth to another rabbit).

We could also use the EA as a kind of faster primality tester. If we choose some number of digits, say 12, for numbers to test for primality, as we've said, we'd need the primes up to

$10^6 = 1,000,000$ to test against. But if we take those $78,498$ and multiply them together into a single giant number N (which will, it happens, have about $430,000$ digits: Maple 15 quite happily built this number), then we can test a number n with up to 12 digits for primality by computing $\gcd(N,n)$. If n is not prime, it will have a prime factor smaller than 10^6, so it will have a factor in common with N and we will have $\gcd(N,n) > 1$. But if n is prime, it will have no factors in common with N, and $\gcd(N,n) = 1$. So with at most 80 long divisions, we can determine if n is prime! But that first long division, for n into N, will be a 'long' one! It's not entirely clear if this will save any time?!

Leaving this topic behind for awhile, we will move on to (seemingly) new ground. To lead up to it, suppose we did <u>not</u> know that $111 \cdot 123 = 13653$, and somebody told us that $111 * 123 = 13657$. We could immediately see that they were wrong, but why were they wrong? We 'knew' that a number ending in 1, times a number ending in 3, could never end in 7. Working through why that was, it came down to

if $n = 10a + 1$ and $m = 10b + 3$, then $nm = 10(10ab + 3a + b) + 3$, so nm ends in 3

The point, in essense, is that 'ending in 3' means 'has remainder 3 when you divide by 10'. And it would appear that we have all learned that the remainder of a product, when you divide by 10, has something to do with the reaminders of the two factors; it is their 'product'. This same idea can be use to detect other errors in computation: there is a technique called 'casting out nines' that people used to use very frequently before calculators became popular. In essence, it looks at the remainders of n and m on division by 9, and compares it to the remainder of nm. We will explore this more fully next time, mostly in order to find a more general pattern to study.

For next time, our question to think about was:

If $17|a-a_0$ and $17|b-b_0$ figure out how to show that $17|ab-a_0b_0$. And your answer should probably have nothing to do with 17 (!).