’Rithmetic Revisited: What we still don’t know about $+$ and $\times$

Carl Pomerance, **Dartmouth College**
Hanover, New Hampshire, USA

Museum of Mathematics, June 1, 2016

You would think that all of the issues surrounding addition and multiplication were sewed up in third grade!

Well in this talk we'll learn about some things they didn't tell you ...

Here's one thing they *did* tell you:

Find 483×784.

$$
\begin{array}{r}
483 \\
\times\ 784 \\
\hline
1932 \\
3864 \\
3381 \\
\hline
378672
\end{array}
$$

If instead you had a problem with two 23-digit numbers, well you always knew deep down that math teachers are cruel and sadistic. Just kidding!

In principle if you really have to, you could work out 23-digits times 23-digits on paper, provided the paper is big enough, but it's a lot of work.

So here's the real question: How much work?

Of course the amount of work depends not only on the length of the numbers. For example, multiplying 10^{22} by 10^{22}, that's 23-digits times 23-digits, but you can do it in your head.

In general, you'll take each digit of the lower number, and multiply it painstakingly into the top number. It's less work if some digit in the lower number is repeated, and there are definitely repeats, since there are only 10 possible digits. But even if it's no work at all, you still have to write it down, and that's 23 or 24 digits. At the minimum (assuming no zeroes), you have to write down $23^2 = 529$ digits for the "parallelogram" part of the product. And then comes the final addition, where all of those 529 digits need to be processed.

```
                          a a a a a a a a a a a a a a a a a a a a a a a
                        × b b b b b b b b b b b b b b b b b b b b b b b
                          ---------------------------------------------
                          c c c c c c c c c c c c c c c c c c c c c c c
                        c c c c c c c c c c c c c c c c c c c c c c c
                      c c c c c c c c c c c c c c c c c c c c c c c
                    c c c c c c c c c c c c c c c c c c c c c c c
                  c c c c c c c c c c c c c c c c c c c c c c c
                c c c c c c c c c c c c c c c c c c c c c c c
              c c c c c c c c c c c c c c c c c c c c c c c
            c c c c c c c c c c c c c c c c c c c c c c c
          c c c c c c c c c c c c c c c c c c c c c c c
        c c c c c c c c c c c c c c c c c c c c c c c
      c c c c c c c c c c c c c c c c c c c c c c c
    c c c c c c c c c c c c c c c c c c c c c c c
    . . .
+ c c c c c c c c c c c c c c c c c c c c c c c
  ---------------------------------------------------------------------
```

23 rows of at least 23 digits each

So in general if you multiply two n-digit numbers, it would seem that you'd be taking n^2 steps, unless there were a lot of zeroes. This ignores extra steps, like carrying and so on, but that at worst changes n^2 to maybe $2n^2$ or $3n^2$. We say that the "complexity" of "school multiplication" for two n-digit numbers is of order n^2.

A. A. Karatsuba (1937–2008): Devised a faster way to multiply two n-digit numbers taking about $n^{1.6}$ elementary steps.

Here is Karatsuba's idea: use high school algebra!

Say the numbers A and B each have n digits. Let $m = n/2$ (okay, we assume that n is even). Write

$$A = A_1 10^m + A_0, \quad B = B_1 10^m + B_0,$$

where A_1, A_0, B_1, B_0 are all smaller than 10^m, so have at most m digits. Then our product AB is

$$AB = (A_1 B_1) 10^{2m} + (A_1 B_0 + A_0 B_1) 10^m + A_0 B_0,$$

so our problem is broken down to 4 smaller multiplication problems, each of size $m \times m$, namely

$$A_1 B_1, \quad A_1 B_0, \quad A_0 B_1, \quad A_0 B_0,$$

and each of these would seem to take $1/4$ as much work as the original problem.

So, unfortunately 4 problems each taking 1/4 as much work, is no savings!

However, we also have

$$(A_1 + A_0)(B_1 + B_0) = A_1B_1 + (A_1B_0 + A_0B_1) + A_0B_0,$$

so we can really do it in 3 multiplications, not 4 (!). Namely,

$$A_1B_1, \quad A_0B_0, \quad (A_1 + A_0)(B_1 + B_0).$$

After we do these, we have our three coefficients, where the middle one, $A_1B_0 + A_0B_1$, is the third product minus the first two:

$$A_1B_0 + A_0B_1 = (A_1 + A_0)(B_1 + B_0) - A_1B_1 - A_0B_0.$$

This idea can then be used on each of the three smaller multiplication problems, and so on down the fractal road, ending in about $n^{1.6}$ elementary steps.

Karatsuba's method was later improved by Toom, Cook, Schönhage, & Strassen. After their efforts we have the *Fast Fourier Transform* that allows you to multiply in about $n \cdot L(n)$ steps, where $L(n)$ is short-hand for the number of digits of n. (So $L(n)$ is the number of digits of the number of digits of the numbers being multiplied!)

Small improvements were made by Fürer in 2007 and by De, Kurur, Saha, & Saptharishi in 2008.

We don't know if we have reached the limit! In particular:

What is the *fastest* way to multiply?

Let's play **Jeopardy Multiplication!**

Here are the rules: I give you the answer to the multiplication problem, and you give me the problem phrased as a question. You must use whole numbers larger than 1.

So, if I say "15", you say "What is 3×5?"

OK, let's play.

So, here's what we don't know:

How many steps does it take to figure out the factors if you are given an n-digit number which *can be* factored?

(A trick problem would be: 17. The only way to write it as $a \times b$ is to use 1, and that was ruled out. So, prime numbers cannot be factored, and the thing we don't know is how long it takes to factor the non-primes.)

The best answer we have so far is about $10^{n^{1/3}}$ steps, and even this is not a theorem, but our algorithm (known as the **number field sieve**) seems to work in practice.

This is all crucially important for the security of Internet commerce. Or I should say that Internet commerce relies on the premise that we **cannot** factor much more quickly than that.

A couple of words about factoring, that is, on how to win at **Multiplication Jeopardy**.

The trick with 8051 (due to **Fermat**), namely that $8051 = 8100 - 49$, is sort of generalizable as might be illustrated by 1649.

We look for a square just above 1649. The first is $41^2 = 1681$. Well

$$41^2 - 1649 = 32 \quad \text{and 32 is } \textbf{not} \text{ a square.}$$

Try again. The next square is $42^2 = 1764$ and

$$42^2 - 1649 = 115 \quad \text{and 115 is } \textbf{not} \text{ a square.}$$

Trying again, the next square is $43^2 = 1849$ and

$$43^2 - 1649 = 200 \quad \text{and 200 is } \textbf{not} \text{ a square.}$$

But wait, look at our 3 non-squares: 32, 115, 200.

Note that we can **make** a square out of them:

$$32 \times 200 = 6400 = 80^2.$$

In general, if N is a positive integer, we'll write $x \equiv y \pmod{N}$ if x, y leave the same remainder when divided by N. For example, $17 \equiv 37 \pmod{10}$ and $43 \equiv 98 \pmod{11}$. It's really very handy notation!

Let $N = 1649$, the number we're trying to factor. Then we have

$$41^2 \equiv 32 \pmod{N}, \quad 43^2 \equiv 200 \pmod{N},$$

and so

$$(41 \times 43)^2 = 41^2 \times 43^2 \equiv 32 \times 200 = 80^2 \pmod{N}.$$

Now $41 \times 43 \equiv 114 \pmod{N}$, so $114^2 \equiv 80^2 \pmod{N}$.

It is not true that $N = (114 - 80)(114 + 80)$, but it is true that the **greatest common divisor** of $114 - 80 = 34$ with N is 17.

Hey! That proves that $N = 1649$ is divisible by 17. Dividing, the other factor is 97. So, we have it: **What is 17×97?**

The various elements here can actually be made into a speedy algorithm, the **quadratic sieve**. The **number field sieve** is a fancier version but has the same underlying flavor of assembling squares whose difference is divisible by N.

Despite our success with factoring, it still is very difficult. Hard numbers with 300 decimal digits are beyond our reach at present. The really amazing thing is we can apply our ignorance to make a secure cryptographic system!

Let's play **Hack the bank!**

We'll discuss a simple cryptographic system and how we might hack it.

Here's the set up. We have public numbers N, E and the bank has a secret number D.

Say we have a message M to send the bank. First, M is digitized and broken into pieces, so that each piece is a number smaller than N. Let's just assume that M is already one of these pieces.

Here's how to send a message to the bank: Multiply your message M by E, then divide by N and get the remainder R. So

$$EM \equiv R \pmod{N}.$$

The bank then decrypts and finds the message M by multiplying R by the secret number D, divides by N, and the remainder will be M. That is,

$$DR \equiv M \pmod{N}.$$

Before we play, here's a simple example to get us started.

Say the public numbers are $E = 96$, $N = 1001$. And say the message is $M = 561$.

So first up, encrypting: $96 \times 561 = 53856$, and when we divide by 1001, the remainder (just subtract 53053) is 803.

The bank has a secret number D, which in this case happens to be 73. Now $73 \times 803 = 58619$, and dividing that by 1001, the remainder is, sure enough, 561.

It's not important that the message is 561, this would work for any number up to 1000, with the same public numbers E, N and the same secret number D.

OK, let's try it from the bank's perspective.

An encrypted message has come in, and it is the number 591.

We're using the same public numbers $E = 96$, $N = 1001$, and the bank's secret number $D = 73$. Their computer is down at the moment, let's help them decrypt and find the message.

Now this part is more fun. We're going to hack the bank. We're going to try and decrypt an encrypted message **not knowing** the secret number D.

So, to keep things simple, we'll use the same public number $N = 1001$, but we'll change the public encryption number to $E = 106$. The encrypted message is $R = 789$.

What could the message be?

The numbers here are not so large, and we could use brute force. If the original message were the number 1, then the encrypted message would be E, and then D would satisfy

$$DE \equiv 1 \pmod{N}, \quad \text{that is} \quad 106D \equiv 1 \pmod{1001}.$$

We could start at $D = 1$, then $D = 2$, and so on until the correct choice is found.

Now replace 1001 with a googol plus 1, namely $10^{100}+1$. Now it's not so easy to sequentially guess secret numbers D.

How can we hack the bank without guessing?

Now replace 1001 with a googol plus 1, namely $10^{100}+1$. Now it's not so easy to sequentially guess secret numbers D.

How can we hack the bank without guessing?

Actually, there's a simple and speedy algorithm due to **Euclid**, discovered 2300 years ago that will let us hack the bank!

In our case with public numbers $E = 106$ and $N = 1001$, let's work it out.

We look at the division problems

$1001 \div 106 = 9\,r47,\ \ 106 \div 47 = 2\,r12,\ \ 47 \div 12 = 3\,r11,\ \ 12 \div 11 = 1\,r1.$

Each of these can be written as an equation for the remainder:

$47 = 1001 - 9 \cdot 106,\ \ 12 = 106 - 2 \cdot 47,\ \ 11 = 47 - 3 \cdot 12,\ \ 1 = 12 - 1 \cdot 11.$

We then carefully back substitute, starting by replacing "11" in the last equation with the expression in the next-to-last one:

$$1 = 12 - 1 \cdot (47 - 3 \cdot 12) = 4 \cdot 12 - 1 \cdot 47.$$

We then substitute for 12, getting

$$1 = 4 \cdot (106 - 2 \cdot 47) - 1 \cdot 47 = 4 \cdot 106 - 9 \cdot 47,$$

and once more:

$$1 = 4 \cdot 106 - 9 \cdot (1001 - 9 \cdot 106) = 85 \cdot 106 - 9 \cdot 1001.$$

Our last line from before:

$$1 = 85 \cdot 106 - 9 \cdot 1001.$$

And there you have it!

$$85 \cdot 106 \equiv 1 \pmod{1001}.$$

The secret number is $D = 85$.

Recall the encrypted message was $R = 789$. Then, $85 \times 789 = 67065$, and dividing by 1001, the remainder is 999. That's it! The message was 999, we've hacked the bank.

Actually, a slight enhancement of this cryptosystem is in widespread use in e-commerce! The enhancement: Instead of encrypting by multiplying the message by E, we encrypt by raising the message to the power E:

$$R \equiv M^E \pmod{N}.$$

And decrypting is via raising to the power D:

$$R^D \equiv M \pmod{N}.$$

To hack this cryptosystem, it's necessary to factor N. If you can do so, you find another number N' in a straightforward way, and D, E are related by $DE \equiv 1 \pmod{N'}$. So, you can hack this system with Euclid, but first you need to find N', and finding N' requires factoring N.

And there it is, an application of our ignorance!

Here are two famous unsolved problems involving both addition and multiplication:

Goldbach's conjecture: **Every even number starting with 4 is the sum of two primes.**

The twin prime conjecture: **There are infinitely many pairs of primes that differ by 2.**

Here's a famous unsolved problem involving only simple 'rithmetic:

For even n, let $f(n) = n/2$ and for odd n, let $f(n) = (3n+1)/2$.

Consider the sequence $n, f(n), f(f(n)), \ldots$.
For example: $3 \mapsto 5 \mapsto 8 \mapsto 4 \mapsto 2 \mapsto 1 \mapsto 2 \mapsto 1 \ldots$
Or: $7 \mapsto 11 \mapsto 17 \mapsto 26 \mapsto 13 \mapsto 20 \mapsto 10 \mapsto 5 \mapsto \ldots \mapsto 1$

Is it true that starting with any positive integer n, the sequence $n, f(n), f(f(n)), \ldots$ eventually hits the number 1?

And here's another famous problem (in disguised form):

Let $A(N)$ be the least common multiple of $1, 2, \ldots, N$

Let $B(N) = \dfrac{1}{1} + \dfrac{1}{2} + \cdots + \dfrac{1}{A(N)}$.

For example: $A(10) = 2520$ and
$B(10) = \frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{2520} \approx 8.4$.
And: $B(100{,}000{,}000) \approx 99{,}998{,}243.4$.

Do we always have $|B(N) - N| < \sqrt{N}L(N)^2$?
(Recall: $L(N)$ is the number of digits of N.)

The Clay Mathematics Institute offers \$1,000,000 for a proof!

Here's an unsolved problem concerning just addition.

We all recall the addition table:

+	1	2	3	4	5	6	7	8	9	10
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
3	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
4	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
5	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
6	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
7	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
8	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
9	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**

The 10×10 array of sums has all the numbers from $\mathbf{2}$ to $\mathbf{20}$ for a total of 19 different sums.

If you were to try this for the $N \times N$ addition table we'd see all of the numbers from $\mathbf{2}$ to $\mathbf{2}N$ for a total of $2N - 1$ different sums.

Now, what if we were to be perverse and instead of having the numbers from 1 to N, we had some arbitrary list of N different numbers.

Can you arrange it so there are *fewer* than $2N - 1$ different sums?

The 10×10 array of sums has all the numbers from $\mathbf{2}$ to $\mathbf{20}$ for a total of 19 different sums.

If you were to try this for the $N \times N$ addition table we'd see all of the numbers from $\mathbf{2}$ to $\mathbf{2}N$ for a total of $2N - 1$ different sums.

Now, what if we were to be perverse and instead of having the numbers from 1 to N, we had some arbitrary list of N different numbers.

Can you arrange it so there are *fewer* than $2N - 1$ different sums?

If you answered "No, there are always at least $2N - 1$ different sums," you'd be right.

Here's an example where there are many different sums:

+	1	2	4	8	16	32	64	128	256	512
1	**2**	**3**	**5**	**9**	**17**	**33**	**65**	**129**	**257**	**513**
2	**3**	**4**	**6**	**10**	**18**	**34**	**66**	**130**	**258**	**514**
4	**5**	**6**	**8**	**12**	**20**	**36**	**68**	**132**	**260**	**516**
8	**9**	**10**	**12**	**16**	**24**	**40**	**72**	**136**	**264**	**520**
16	**17**	**18**	**20**	**24**	**32**	**48**	**80**	**144**	**272**	**528**
32	**33**	**34**	**36**	**40**	**48**	**64**	**96**	**160**	**288**	**544**
64	**65**	**66**	**68**	**72**	**80**	**96**	**128**	**192**	**320**	**576**
128	**129**	**130**	**132**	**136**	**144**	**160**	**192**	**256**	**384**	**640**
256	**257**	**258**	**260**	**264**	**272**	**288**	**320**	**384**	**512**	**768**
512	**513**	**514**	**516**	**520**	**528**	**544**	**576**	**640**	**768**	**1024**

So, sometimes there are few distinct sums and sometimes many.

What structure is forced on the set if there are few distinct sums?

We know the answer when there are very few distinct sums:

Gregory Freiman

Here's something with multiplication tables.

Let's look at the $N \times N$ multiplication table using the numbers from 1 to N. With addition, we were able to count exactly how many distinct numbers appear in the table.

How many different numbers appear in the $N \times N$ multiplication table?

Let $M(N)$ be the number of distinct entries in the $N \times N$ multiplication table.

$\times$	1	2	3	4	5	6	7	8	9	10
1	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
2	2	4	6	8	10	**12**	**14**	**16**	**18**	**20**
3	3	6	9	12	**15**	18	**21**	**24**	**27**	**30**
4	4	8	12	16	20	24	**28**	**32**	**36**	**40**
5	5	10	15	20	**25**	30	**35**	40	**45**	**50**
6	6	12	18	24	30	36	**42**	**48**	**54**	**60**
7	7	14	21	28	35	42	**49**	**56**	**63**	**70**
8	8	16	24	32	40	48	56	**64**	**72**	**80**
9	9	18	27	36	45	54	63	72	**81**	**90**
10	10	20	30	40	50	60	70	80	90	**100**

So $M(10) = 42$.

It is really amazing that though $M(N)$ is not far below N^2 looking "from a distance", if we look "close up" we see that $M(N)/N^2$ tends to 0 as N grows larger and larger.

It may be too difficult to expect a neat exact formula for $M(N)$.

After **Erdős**, **Tenenbaum**, and **Ford**, we now know the (complicated) order of magnitude for $M(N)$ as N grows. (It's something like $N^2/L(N)^E L(L(N))^{1.5}$, where $E = 0.086\ldots$ is an explicitly known constant.)


Paul Erdős, 1913–1996

Find an asymptotic formula for $M(N)$ as N grows?

Let me close with one unified problem about addition and multiplication tables. It's due to **Erdős & Szemerédi**.

Look at **both** the addition and multiplication tables for N carefully chosen numbers.

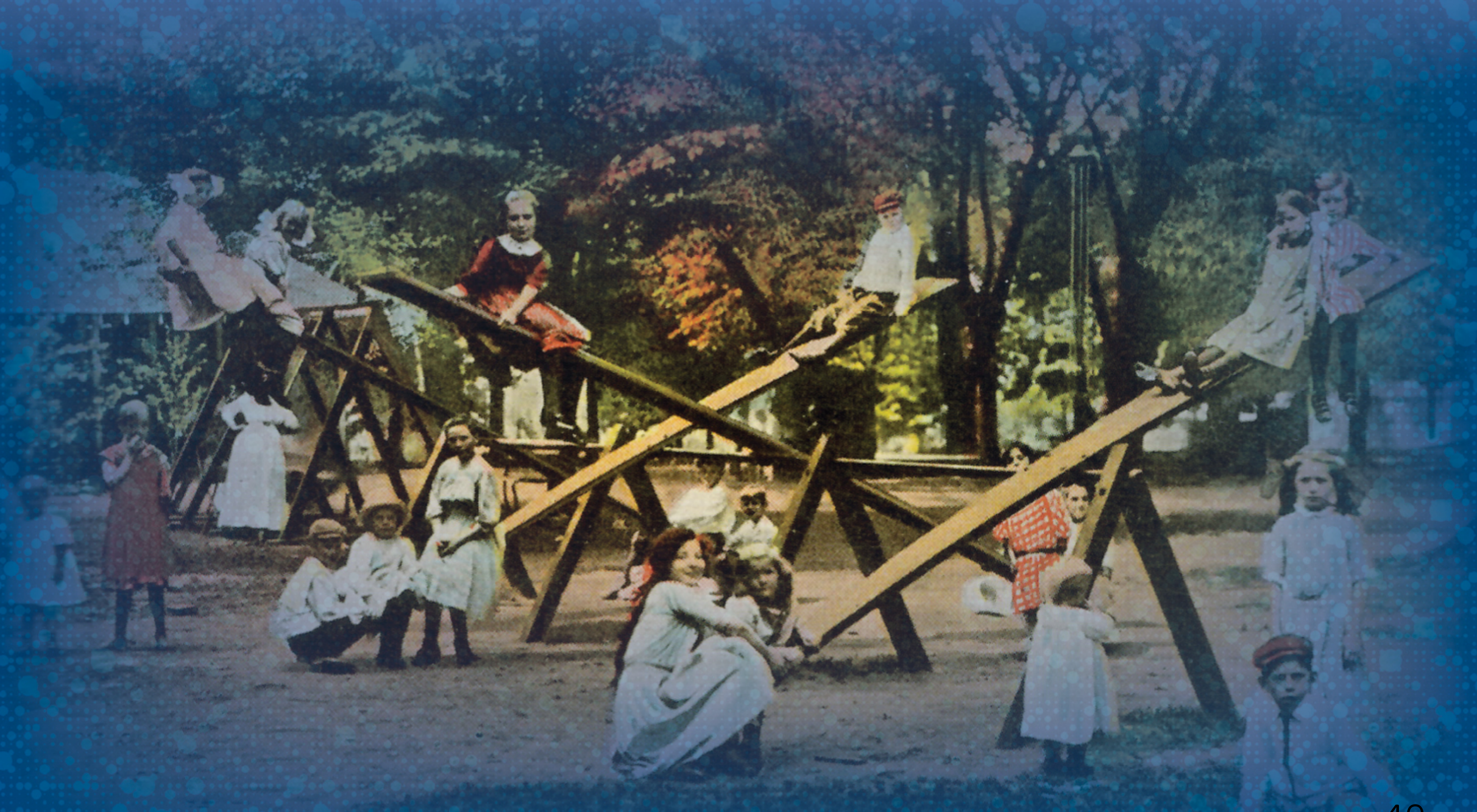
We've seen that if we take the first N numbers we get close to N^2 distinct entries in the multiplication table, but few in the addition table.

At the other extreme, if we take for our N numbers the powers of 2, namely $1, 2, 4, \ldots, 2^{N-1}$, then there are at least $\frac{1}{2}N^2$ distinct entries in the addition table and only $2N-1$ entries in the multiplication table.

If we take N *random* numbers, then it's likely both tables have close to N^2 distinct entries.

The question is: **If we choose our numbers so that the number of distinct entries in one table is small, must the other always be large?**

MATH ENCOUNTERS



40

'RITHMETIC *revisited*

×	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

+	1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10	11
2	3	4	5	6	7	8	9	10	11	12
3	4	5	6	7	8	9	10	11	12	13
4	5	6	7	8	9	10	11	12	13	14
5	6	7	8	9	10	11	12	13	14	15
6	7	8	9	10	11	12	13	14	15	16
7	8	9	10	11	12	13	14	15	16	17
8	9	10	11	12	13	14	15	16	17	18
9	10	11	12	13	14	15	16	17	18	19
10	11	12	13	14	15	16	17	18	19	20

Many products Few sums

$$\{1, 2, \ldots, N\}$$

×	**1**	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
1	1	2	4	8	16	32	64	128	256	512
2	2	4	8	16	32	64	128	256	512	1024
4	4	8	16	32	64	128	256	512	1024	2048
8	8	16	32	64	128	256	512	1024	2048	4096
16	16	32	64	128	256	512	1024	2048	4096	8192
32	32	64	128	256	512	1024	2048	4096	8192	16384
64	64	128	256	512	1024	2048	4096	8192	16384	32768
128	128	256	512	1024	2048	4096	8192	16384	32768	65536
256	256	512	1024	2048	4096	8192	16384	32768	65536	131072
512	512	1024	2048	4096	8192	16384	32768	65536	131072	262144

+	**1**	**2**	**4**	**8**	**16**	**32**	**64**	**128**	**256**	**512**
1	2	3	5	9	17	33	65	129	257	513
2	3	4	6	10	18	34	66	130	258	514
4	5	6	8	12	20	36	68	132	260	516
8	9	10	12	16	24	40	72	136	264	520
16	17	18	20	24	32	48	80	144	272	528
32	33	34	36	40	48	64	96	160	288	544
64	65	66	68	72	80	96	128	192	320	576
128	129	130	132	136	144	160	192	256	384	640
256	257	258	260	264	272	288	320	384	512	768
512	513	514	516	520	528	544	576	640	768	1024

Few products Many sums

$$\{1, 2, 4, \ldots, 2^{N-1}\}$$

Must one always be large?

Put more precisely: **If we have N distinct numbers, must one of**

- **the number of distinct pairwise sums,**

- **the number of distinct pairwise products,**

be greater than $N^{1.999}$ for all large values of N?

We don't know.

And it's not for lack of trying.

The game players with the sum/product problem include: **Erdős**, **Szemerédi**, **Nathanson**, **Chen**, **Elekes**, **Bourgain**, **Chang**, **Konyagin**, **Green**, **Tao**, **Solymosi**, ...

The best that's been proved (**Solymosi**) is that there are at least $N^{4/3}$ different entries.

This list of mathematicians contains two Fields Medalists, a Wolf Prize winner, an Abel Prize winner, four Salem Prize Winners, two Crafoord Prize winners, and an Aisenstadt Prize winner.

And still the problem is not solved!

My message: We could use a little help with these problems!!

My message: We could use a little help with these problems!!

THANK YOU