

Tune-Up: Times Tables

Problem Statement

If you consider N randomly chosen integers from 1 to x - - - N random elements of the set $\{1, 2, \dots, x\}$ - - - how large should N be so that with high probability we can take a bunch of the numbers we have just chosen, multiply them and get a perfect square?

Goal Factor large integers.

This is a difficult problem to resolve. Let's choose $x = 100$ and $N = 5$ there are $\binom{x}{N} = \binom{100}{5} < 100^5 = 10^{10}$. The exact number of 5 element subsets of $\{1, \dots, 100\}$ is

$$\#\{subsets\} = \binom{100}{5} = \frac{100 \times 99 \times 98 \times 97 \times 96}{5 \times 4 \times 3 \times 2 \times 1} = 75287052.0 \approx 7.5 \times 10^8$$

The symbol we are using " \approx " deserves a lot of qualification given the level of journal we are reading there. There are many possible notions of "closeness" we could define here, example: [agree in the first 3 digits, so that $300 \approx 30.0$.] This is a silly notion of closeness yet if we do enough consistency checks, this too could also be viable.

The set of subsets of $\{1, \dots, x\}$ is there a more natural element of saying this? Here's some examples:

$$\left\{ \{5, 10, 15, 20, 25\}, \{73, 74, 75, 76, 77\}, \{30, 40, 50, 60, 70\} \right\} \subseteq \left\{ \mathbf{5} \rightarrow \{1, \dots, 10^2\} \right\}$$

The we are defining a \times -like function $\times : \{\text{subsets}\} \rightarrow \mathbb{Z}$. Using a quick computer search here's nice example:

$$\left| \sqrt{21 \times 31 \times 71 \times 74 \times 79} - 16438 \right| < 10^{-2}$$

This requires us to have a **metric** on \mathbb{Q} that completes to \mathbb{R} (or to some other interesting topological space!) Let's roll the dice again:

$$\sqrt{44 \times 119 \times 132 \times 153 \times 189} = 141372 + \mathbf{0}$$

This is accurate to all decimal places. Our most basic example of a profinite group is \mathbb{R} itself. Why could the decimal system with carries matter so much?

$$\sqrt{28 \times 30 \times 42 \times 49 \times 180} = 17640 + \mathbf{0}$$

Exact results again. Are three decimal places sufficient for the problem we just described? No.

$$\left| \sqrt{14 \times 59 \times 109 \times 197 \times 198} - 59261 \right| < 10^{-3}$$

so with a computer we can somehow get a nice approximate result. All of our work is hidden in the computer program itself.

```
import numpy as np

N = 200
n = np.arange(N) + 1

for t in range(10000):
    # this method doesn't ensure DISTINCT random integers
    m = n[ (N*(np.random.random(5))).astype(int)]
    y = np.product(m)
    if (y**0.5 % 1) < 0.001:
        if (y**0.5 % 1) > 0:
            print(m, y, y**0.5)
```

So we have yet to look under the hood. In some cases, merely checking for prime numbers – I have yet to specify how we decided a number was a perfect square. E.g.

$$3511866204 \in \square$$

Really the test for squares in \mathbb{Z} is to check that $n/p^2 \in \mathbb{Z}$ and yet $n/p \notin \mathbb{Z}$. How do we tell our computer to do the times tables? How does our computer even know what decimal digits are?

Even if we do all that, how much heat are we generating getting these results? So we could let $N = 10^{100}$ and we have to decide how much integers there is to get a perfect square.

Bonus We need all these fabulous examples of numbers:

$$\left| \sqrt{2 \times 87 \times 159 \times 188 \times 212 \times 221} - 7929139 \right| < 10^{-4}$$

Professional hints: one the authors is specialist in **random graph theory** and **functional analysis**. So why are things so bad? It's just that we are consuming more and more paper trying to get exact multiplication results out of large numbers. Any time we did an approximation – settled for a “close” result – any time we made a choice of algorithms we could have done ten other ways. It's OK.

05/18 Let's find an example with 4-digit numbers:

$$\sqrt{1071 \times 2626 \times 3491 \times 3699 \times 4555 \times 8181 \times 8298} \in \mathbb{Z}$$

It looks like the answer is $\sqrt{\dots} = 1224323517$. This result is **false**. What does that mean? What's so bad about this result? Depending on how we measure it, this result could be acceptable.

$$\left| \sqrt{1071 \times 2626 \times 3491 \times 3699 \times 4555 \times 8181 \times 8298} - 1224323517 \right| < 1 \times 10^{-4}$$

We also have error in the other direction. The mistakes are consistently at least this big.

$$\left| \sqrt{1071 \times 2626 \times 3491 \times 3699 \times 4555 \times 8181 \times 8298} - 1224323517 \right| > 7 \times 10^{-5}$$

With a computing machine of some kind (in this case, just a laptop – a pocket calculator could also suffice), meaningful *highly non-trivial* example of numbers *approximately* in \square can be generated.

Numbers are mostly shorthands or representations of physical ideas or concepts or observations. Even symbols like “<” and “>” need to be reviewed.

- How much work is the computer doing to generate these examples?
- How do we know the computer is not lying to us?
- The typical program (e.g. solve $a \times b$) asks for resources like this:
 - we have a bunch of 5-digit numbers
 - we need 1000 additions
 - we need 3000 multiplications

Then we get a result. What did the steps in the middle look like?

- What's the difference between a result that's “close enough” and an outright “lie”. Solving a problem can be used to lie in other settings.

```
RuntimeWarning: invalid value encountered in power
if (y**0.5 % 1) < 0.0001:
untimeWarning: invalid value encountered in double_scalars
if (y**0.5 % 1) < 0.0001:
```

So we are approaching needle-in-haystack territory and yet the computer finds the examples just fine. The computing machine or collection of machines that produce a result together.

$$\left| \sqrt{1940 \times 2001 \times 2287 \times 4189 \times 4441 \times 6699 \times 8162} - 2671434353 \right| < 10^{-6}$$

This is a \$1000000 result (not really). The error message says we're running out of space doing our computation.

How much space are we using with these numbers $7136561502388545640 < 10^{20}$ so we needed this number to be exact to 20 decimal place. We are sure this problem has some slack somewhere, yet the usual places don't work.

- Can we benefit from our failed attempts to solve the problem?
- If we complain the result is not exact, describe the [opposite of exact]
- What if we run out of space resources or time resources in the middle of the problem?

Our search for solutions is example of **martingale** problem. We criterion for solution which is difficult to solve exactly, so we randomly search for solutions and stop when our solution is close enough.

Let's try this problem with complex numbers which are now called Gaussian integers.

$$\left| \sqrt{(90 + 13i) \times (30 + 17i) \times (15 + 92i) \times (85 + 46i) \times (32 + 37i)} - () \right| < 10^{-2}$$

When all we do take the answer, we throw away the wrapper, the shipping information and ton of valuable information. The answer for now is

$$\begin{aligned} \sqrt{\dots} &\approx 5194 + 36808i \\ \sqrt{\dots} &\in 10^{-2}\{|z| < 1\} + \mathbb{Z}[i] \end{aligned}$$

This shape is just a lattice with circles everywhere. Number Theorists are so confident their examples work and they're dime-a-dozen that they might prefer to work abstractly or in coded fashion.

References

[1] ...