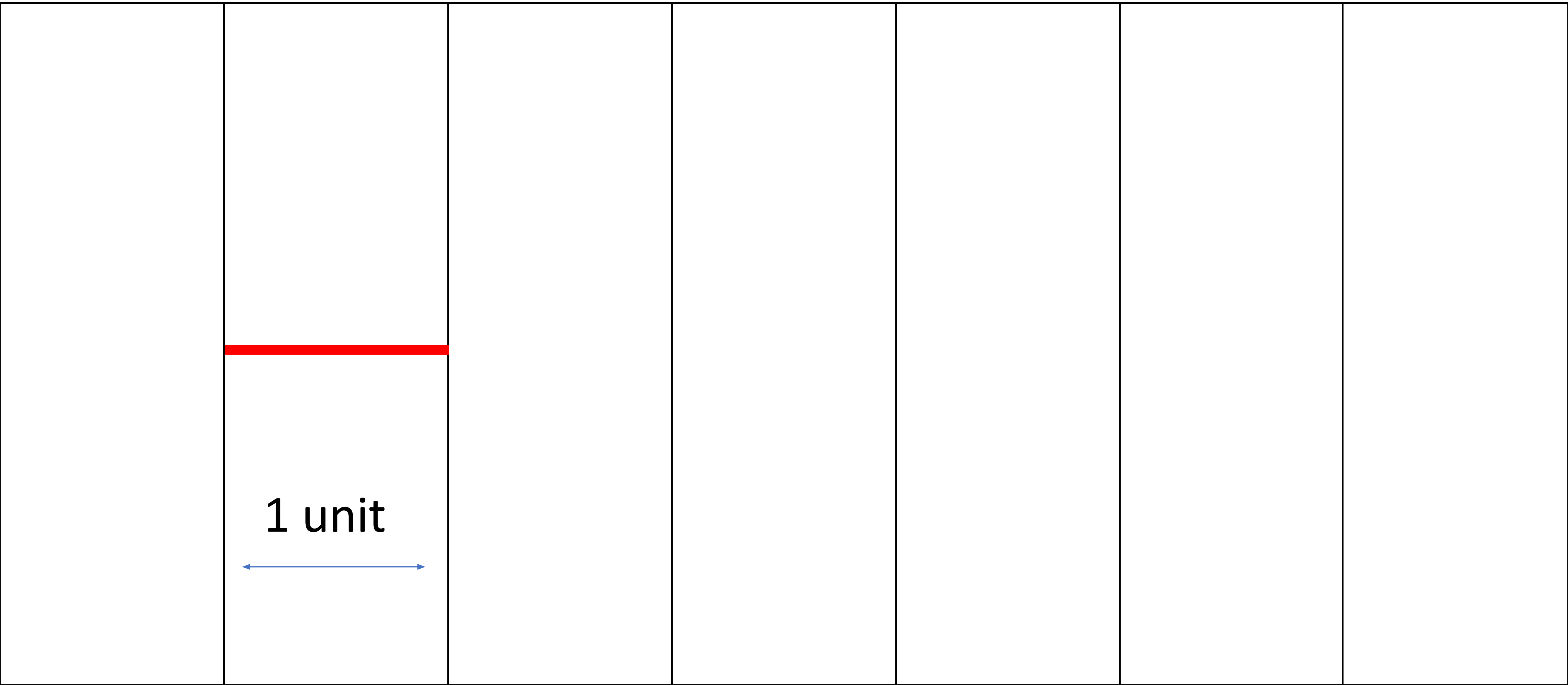
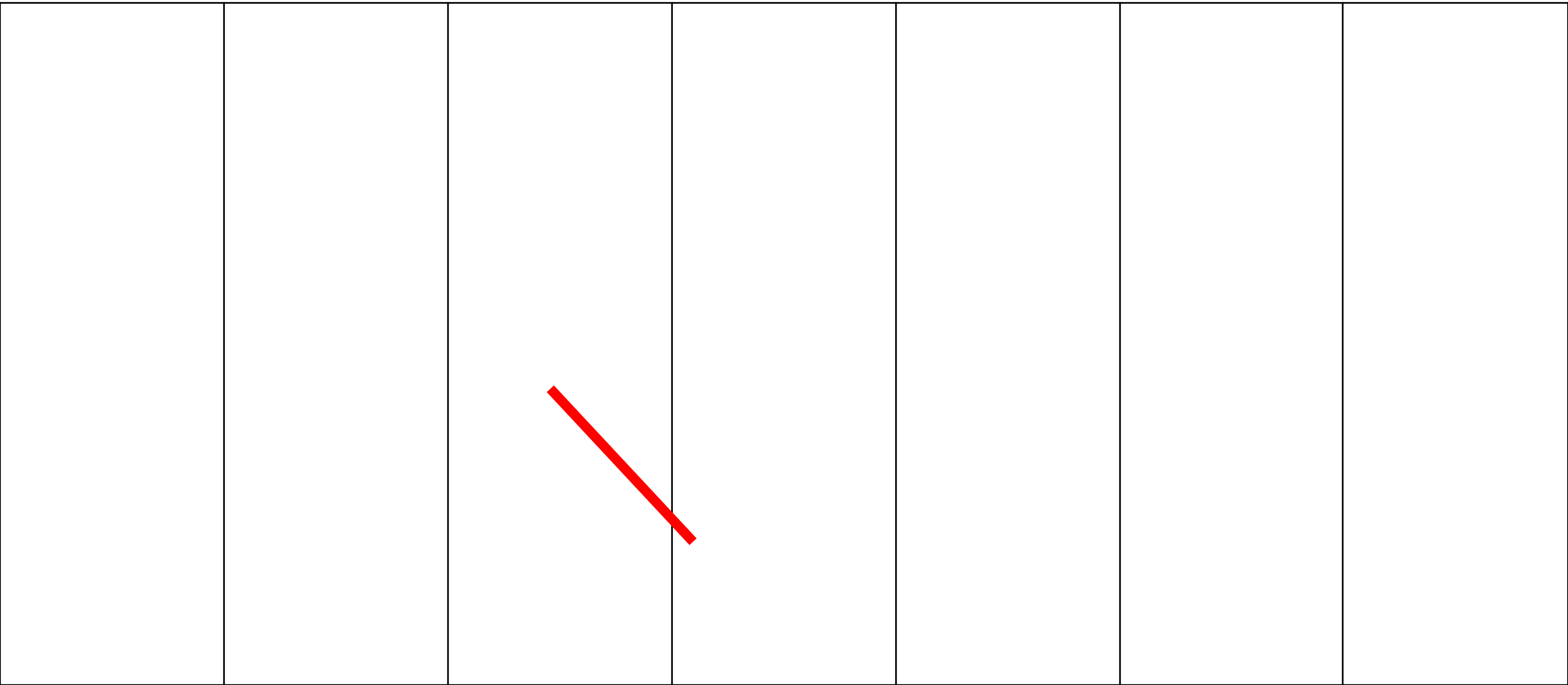
Next Generation Number Theory !

-Shivam Patel

There is a need!

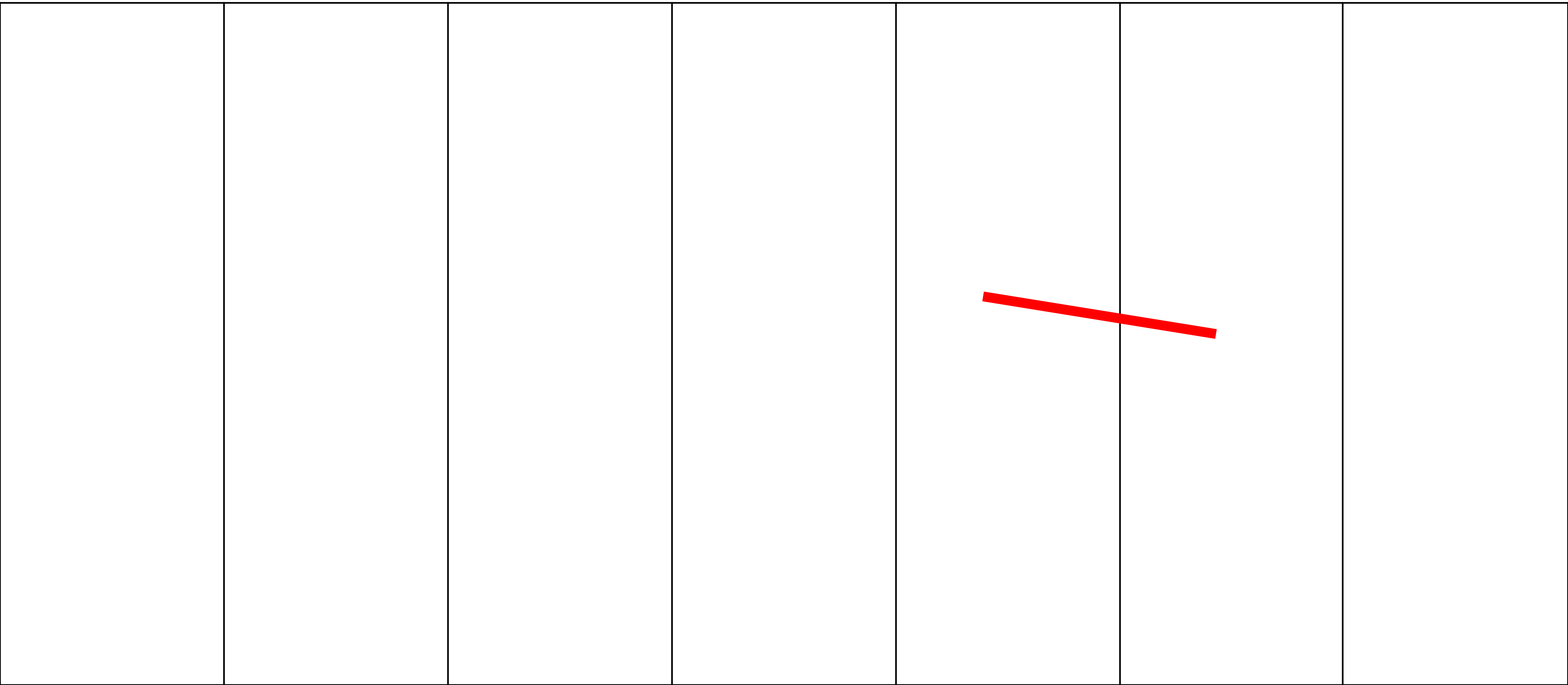


There is a need!



There is a need!

There is a need!



Experimental Mathematics

- **Experimental mathematics** is an approach to mathematics in which computation is used to investigate mathematical objects and identify properties and patterns.

- It is based upon the algorithms and computational power developed in the modern times . It is carefully generating data , or use available data exhaustively .

- It has numerous applications.

What has did to do with number theory?

In Number theory computation algorithms are becoming
Increasingly common.

Especially in tools like finding roots of polynomials , finding integer
Relations , the computational tools became indispensable.

These methods help algorithms improve other algorithms
at a fundamental level and hence are useful.

Consider the problem of integer factorisation!

Given a number N one desires to find p and q such that

$$N=pq$$

A naive way of doing

```
def FindAllDivisors(x):
    divList = []
    y = 1
    while y <= math.sqrt(x):
        if x % y == 0:
            divList.append(y)
            divList.append(int(x / y))
        y += 1
    return divList
```

A slight better way of doing

```
def factorize(n, primes):
    factors = []
    for p in primes:
        if p*p > n: break
        i = 0
        while n % p == 0:
            n //= p
            i+=1
        if i > 0:
            factors.append((p, i));
    if n > 1: factors.append((n, 1))

    return factors
```

```
def divisors(factors):
    div = [1]
    for (p, r) in factors:
        div = [d * p**e for d in div for e in range(r + 1)]
    return div
```

O(n)

An even better way !

The first non trivial algorithm for factorisation was due to John Poland -:

Consider some prime factor p of N. If we could magically find an exponent e such that e divides (p-1) then for almost any base b we would have:

$b^e = 1 \pmod{p}$ [Fermat's Little Thm]

$b^e - 1 = 0 \pmod{p}$

so $b^e - 1 = kp \pmod{N}$

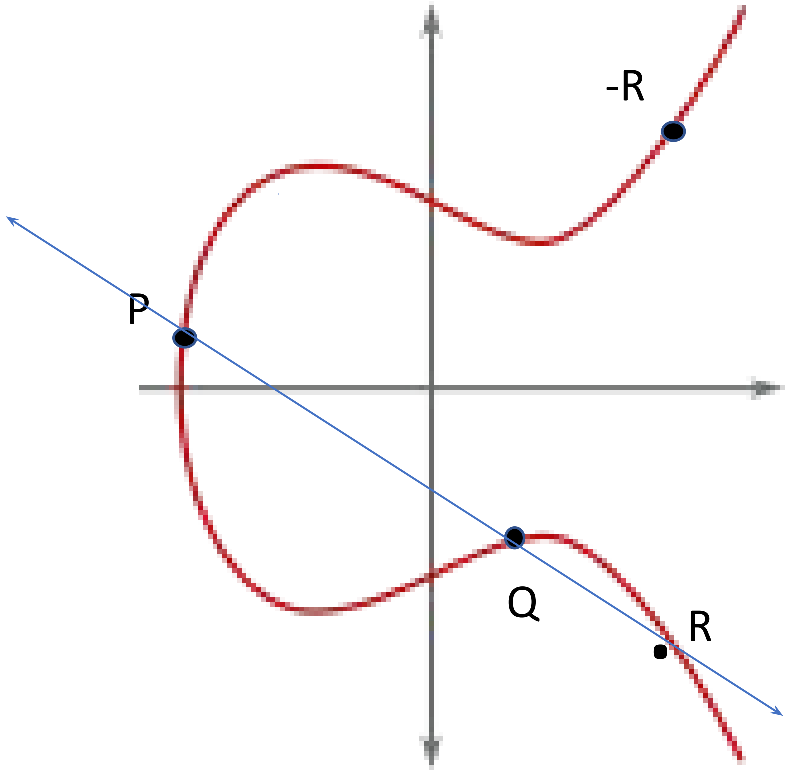
and $\gcd(b^e, N) = p$ (probably, unless k also divides N)

This does not show how to find the exponent e , it can be done as:

This, of course, avoids the question of how we found the magic exponent e. We will outline one popular approach. Guess a limit L such that L is greater than all prime factors of (p-1) (remember, you don't know the value of p, so this is just a guess). Now let e = (L!). This gives us the following algorithm:

The best method!

Elliptic Curve



$$y^2 = x^3 - x + 1$$

1) Choose two points A and B.
2) Draw a line from A to B.
3) Where the points intersects mark it to be R.
4) Then reflect the point to the curve and this newly obtained -R is your answer.

We do similar thing modulo n.

Lenstra's Elliptic Curve Algorithm. *Let $n \geq 2$ be a composite integer for which we are to find a factor.*

Step 1 *Check that* $\gcd(n, 6) = 1$ *and that n does not have the form m^r for some $r \geq 2$.*

Step 2 *Choose random integers b, x_1, y_1 between 1 and n.*

Step 3 *Let $c = y_1^2 - x_1^3 - bx_1 \pmod{n}$, let C be the cubic curve*

$$C : y^2 = x^3 + bx + c, \qquad \text{and let } P = (x_1, y_1) \in C.$$

Step 4 *Check that* $\gcd(4b^3 + 27c^2, n) = 1$. *(If it equals n, go back and choose a new b. If it is strictly between 1 and n, then it is a non-trivial factor of n, so we are done.)*

Step 5 *Choose a number k which is a product of small primes to small powers. For example, take*

$$k = \mathrm{LCM}[1, 2, 3, \ldots, K]$$

for some integer K.

Step 6 *Compute*

$$kP = \left(\frac{a_k}{d_k^2}, \frac{b_k}{d_k^3} \right).$$

Step 7 *Calculate $D = \gcd(d_k, n)$. If $1 < D < n$, then D is a non-trivial factor of n and we are done. If $D = 1$, either go back to Step 5 and increase k or go back to Step 2 and choose a new curve. If $D = n$, then go back to Step 5 and decrease k.*

Machine Learning ,Mathematics and Number Theory !

Is π a normal number?

A normal number in base b , a number in which all digits from 0 to b-1 occur equally likely.

- We can use a RNN for this thing , it would understands the sequence , and let it predict the sequence and we can see the accuracy and infer upon it.
- In the modern times a lot of other problems can generate a lot of data , the correct interpretation using machine learning tools could lead to great proofs and insights .

Thank you
For everything*