**The Relation Between Digit Lengths of Exponentiated Numbers and their base:**

**Introduction:**

The representation of numbers in base-10 notation carries more than aesthetic or structural value—it hints at deeper mathematical behaviour hidden within the count of digits themselves. Particularly in the case of exponentiated numbers, patterns emerge in how digit lengths grow, fluctuate, and behave under certain transformations.

This paper introduces a new lens through which to examine such patterns: a formal notation and framework built around digit-length analysis. Drawing inspiration from ancient linguistic roots and modern functional rigor, we present the foundation of a new exploratory field.

The study aims not only to understand how many digits a number possesses, but also to investigate how these counts behave under exponential growth and how they might reveal subtle structures beneath numeric expressions.

**Foundational Concepts:**

To meaningfully engage with the ideas explored in this study, the reader is expected to be familiar with a few foundational mathematical concepts. These provide the structural base upon which the exploration of digit behaviour—particularly in exponentiated expressions—is constructed.

**1. Number Theory (Integer Properties & Place Value)**

At the heart of digit-length analysis lies **number theory**, the study of integers and their inherent properties. Specifically, this work relies on a basic but precise understanding of:

* **The decimal (base-10) system**, where each digit’s position determines its contribution to the overall value.
* The **multiplicative structure of integers**, especially how powers of 10 create thresholds for digit growth.
* Familiarity with **natural numbers**, **prime decomposition**, and the behaviour of integers under operations like squaring and cubing.

Understanding that 10n marks the threshold for an n+1- digit number is a key intuition behind measuring digit length. While advanced number theory is not required, appreciating how integers scale with powers and how positional notation works is fundamental to the logic of this study.

**2. Modular Arithmetic (Remainders and Cycles)**

Modular arithmetic, often described as **"clock arithmetic,"** deals with the remainder when one number is divided by another. Though this study does not focus heavily on modulus operations, the concept becomes tangentially relevant in understanding digit cycles, digit-ending behaviours, and congruencies in exponential expressions.

For example:

* The **last digit** of powers like 2n follows a known cycle, which can be determined using mod 10.
* When assessing whether a certain power has crossed into a new digit length, **modular reasoning** can provide simplified pathways without computing the full number.

In particular, being comfortable with expressions like:

an ≡ r (mod m)

and understanding how they behave when n increases rapidly, helps build a mental model for digit behaviour without brute-force calculation.

**3. Binomial Expansions (Structure of Powers)**

The **binomial theorem** allows expansion of expressions like (a + b)n into a sum of terms involving coefficients and powers:

(a + b)n = ∑ ( ​) an−kbk

n

k

k=0

n

This is especially useful when studying expressions such as (10x+1)n, where:

* Each term contributes differently to the total digit length of the result.
* The **leading term**, often 10nxn, heavily influences the overall magnitude and thus the number of digits.

Understanding how binomial coefficients grow and how different terms dominate based on the value of n gives insight into why digit lengths in such expansions behave the way they do. It also supports the development of bounds and estimations when exact values are not computationally feasible.

**4. Logarithms (Base-10 and Magnitude Estimation)**

Logarithms—particularly base -10 logs—play a **central role** in this paper. The logarithm of a number gives a measure of its **order of magnitude**, making it the natural tool for estimating how many digits a number has.

The crucial identity:

Number of digits of x = ⌊log10​∣x∣⌋+1

is the mathematical heart of the digit-counting function introduced later. Familiarity with logarithmic properties such as:

* Log (ab) = log a + log b
* Log (ab) = b log a
* Log 10(10n) = n

helps analyse how digit lengths grow in exponential expressions and how the growth can be estimated without computing the number directly.

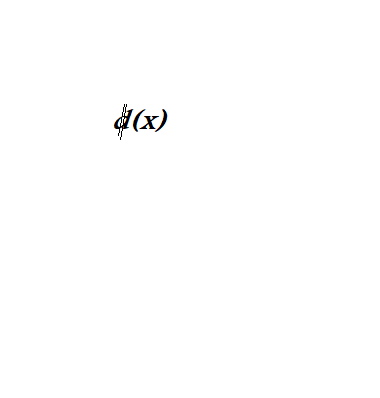
Logarithms serve as a bridge between raw exponential growth and human-readable number length — making them indispensable to the study of the number of digits of a number.

Together, these concepts form the groundwork necessary to dive into the paper’s main objective: defining a formal notation for digit length, analysing its behaviour under exponential operations, and laying the foundations of a new theoretical lens on numerical magnitude.

**Ank function** **: A Cleaner Approach to Digit Length**

In this paper, we introduce the **Ank function**, a compact and mathematically elegant representation of the **number of digits** in a positive integer. Though its internal logic is based on logarithmic principles, the Ank function abstracts away the floor-logarithm expression traditionally used for digit estimation — namely, ⌊log 10x⌋ + 1 — and replaces it with a standalone, symbolic notation. This serves two purposes: to **enhance readability** in digit-length-heavy expressions, and to **promote a consistent theoretical tool** for digit-based analysis in number theory.

Unlike the raw logarithmic form, which clutters expressions and interrupts the visual flow of equations, the Ank function behaves as a clean wrapper that transparently communicates "digit magnitude" without overburdening the notation. It allows digit counts to be composed into larger formulas (such as differences, limits, or transformations) without losing clarity. In this way, the Ank function is not just a mathematical shortcut — it is a notational framework designed for precision, elegance, and future generalization.

One would write the Ank function in the following way:

(The gap is due to some problem in image rendering)

This would be read as ‘Ank of x’ which equates to the number of digits of the number x

(As the following symbol is not present in a keyboard, this document would refer to it as Ank(x), however they way of writing it is the above)

**Properties of Ank:**

Using the idea of Ank, I want to show some properties of the digits of a number, which would be hard to represent using methods like logarithms. The proofs of these properties should not be required if one understands the basic concepts of these properties in other fields of number theory and has some problem solving skills to apply some logic here:

1. **Basic Definition (Integer Case)**

Ank(x) = ⌊log10 ∣x∣⌋ + 1 for x ∈ Z, x ≠0

This defines the number of base-10 digits in a non-zero integer. For negative integers, the minus sign is not counted.

1. **Power of Ten Identity**

Ank(10n) = n +1 for n ∈ N0

This property establishes a direct link between the Ank function and exponential growth of digit length.

1. **Inclusion of Decimal Digits**

Ank(x) =Total count of digits in decimal expansion of x

For numbers with decimal parts, Ank includes **all visible digits**, including those after the decimal point.

1. **Irrational Numbers and Infinite Digits**

Ank(x) = ∞ for all x ∈ Q

This follows from the non-repeating, infinite decimal nature of irrationals (e.g., π, square root of 2, etc.​) and the repeating, non-terminating rational numbers (e.g., 10/9, 250/7 etc.)

**Other Properties of Ank(x)**

1. **Additivity Under Concatenation (but not Arithmetic Addition)**  
   If two integers are concatenated to form a new number (e.g., 45 and 67 → 4567), then:

Ank(concat(a, b)) = Ank(a)+Ank(b)

But note:

Ank(a + b) ≤ max (Ank(a), Ank(b)) + 1

1. **Multiplicative Bound**

Ank(a × b) ≤ Ank(a) + Ank(b)

The product of two numbers will have at most the sum of their digit lengths, but often fewer due to carrying and overlaps.

1. **Monotonicity**  
   For all x, y > 0:

x < y ⇒ Ank(x) ≤ Ank(y)

This holds true assuming no leading zeros, and especially within the same integer digit count.

**The Main Theorem:**

The thing which has been teased since the start. Without further ado, let’s jump right into it.

The main theorem that has been building up till now goes like this:

Ank( (10x + 1)n ) ≡ Ank( xn ) (mod n)

This can be easily understood after review it’s binomial expansion:

(10x + 1)n = (10x)n + nC1 (10x)n-1 … nCn-1 10x + 1

Over hear, it can clearly be observed that the term (10x)n will have the highest number of digits as it has the highest power. From this we can conclude that:

Ank( (10x + 1)n ) = Ank( (10x)n )

This means if we can find the modulus of Ank( (10x)n ) (mod n), we can show the above formula.

We know that:

Ank(x) = ⌊log10 ∣x∣⌋ + 1 for x ∈ Z, x ≠0

And,

Log 10(10n) = n

Log (ab) = log a + log b

Hence:

(10x)n = 10nxn

Log (10nxn) = Log 10n + Log xn

Log 10n = n

Log (10nxn) = n + Log xn

Ank( (10x)n) – 1 = n + Ank( xn ) -1

Ank( (10x)n) = n + Ank( xn )

Taking modulo n on both sides:

Ank( (10x + 1)n ) ≡ Ank( xn ) (mod n)

Hence Proved

**Conclusion:**

The **Ank function**, introduced as a refined way of determining the number of digits in any real number, offers a robust and mathematically elegant alternative to traditional logarithmic methods. By encapsulating both whole and fractional components—including terminating and non-terminating decimals—the Ank function provides a comprehensive measure of a number’s digit structure, while maintaining consistency across positive and negative values.

Through its fundamental properties and behaviours—such as symmetry under negation, scalability with powers of ten, and its handling of irrational or infinitely repeating numbers—the Ank function lays the groundwork for a more nuanced understanding of numeric representation. Unlike the conventional use of ⌊log 10x⌋+1, which breaks down for non-integers or negative values, Ank extends seamlessly into broader domains.

The derived properties, algebraic behaviours under addition and multiplication, and its connections to binomial expansions, provide fertile ground for further exploration, particularly in number theory and computational mathematics. As a symbol of structural depth rather than mere magnitude, the Ank function opens up new paths in digit-based analysis and serves as a fresh conceptual tool in the mathematical toolkit.

The above theorem is one of the most basic uses of the Ank function and can be broadened to a much more complex use. This research however, only marks the beginning of what could be a deeper theoretical framework—one that views digits not merely as numerical artefacts, but as measurable, analysable units of information.

Thank you for your time

-Tanmay Gupta