Graham's Number : How close can we get to infinity?

Arnav Kachhia

March 2025

1 Introduction

In November 1977, the American mathematician Ronald Graham was working on a proof related to a problem in Ramsey theory. During this research, he encountered an upper bound so astronomically large that there is simply no intuitive way to comprehend its magnitude.

But why does this matter? It's not just about size—it's about what Graham's number represents. It challenges our very understanding of what numbers can be, showing us that mathematics can lead us to places we never imagined. Graham's number matters because it pushes the limits of human knowledge and understanding. It's a symbol of how maths can be both simple and mind-blowingly complex, a reminder that even the most seemingly abstract concepts can lead to profound discoveries.

For two decades, Graham's number held the unofficial title of the largest number ever used in a mathematical proof. Although it has since been surpassed, its relatively simple and easy-to-follow derivation has cemented its place in mathematical history, making it one of the most famous large numbers even today.

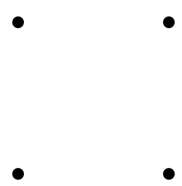
2 Edge Colouring in Graphs

Before delving into the titan of a number, it is crucial to understand the problem's foundation: edge colouring in multidimensional graphs.

Note : a **complete graph** is where there is all vertices are connected to each other by an edge.

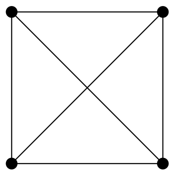
2.1 Complete Graph in 2 Dimensions

Consider the four vertices below, which could be joined to create a square. However, we wish to connect all the vertices as a complete graph, a network where all vertices are connected to each other by a single edge.

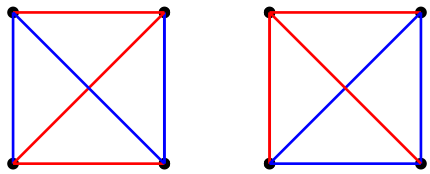


The number of edges required to form a complete graph with 4 vertices is given by the binomial coefficient $\binom{n}{k} = \frac{n!}{k!(n-k)!}$.

$$E = \binom{4}{2} = \frac{4!}{2!(4-2)!} = 6$$



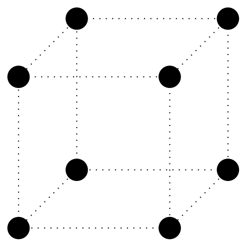
Now we can colour all 6 of the edges either red or blue. There are a total of $2^6 = 64$ ways to color the edges of the 2-D graph, as there are 2 options at each of the 6 edges.



The diagrams above shows 2 of the 64 possible configurations.

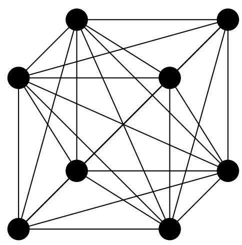
A graph like this (where there are 4 vertices, and 6 edges connecting all 4 together) is more commonly known as K_4 in graph theory.

2.2 Complete Graph in 3 Dimensions



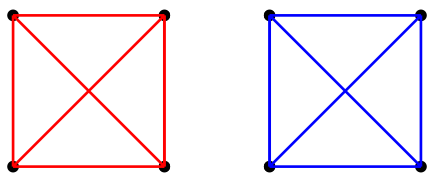
Above there are the 8 vertices of a cube in 3 dimensions. Using the binomial coefficient, we see that the number of edges required to connect all 8 vertices is:

$$E = \binom{8}{2} = \frac{8!}{2!(8-2)!} = 28$$

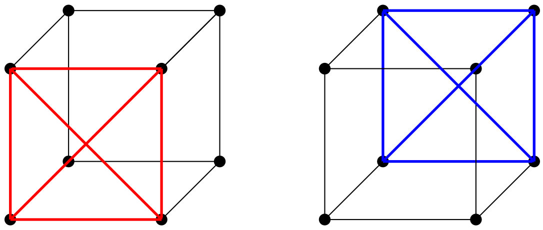


Similar to the 2D square, we can calculate the number of possible configuartions of edge colouring with red and blue as $2^{28} \approx 270$ million.

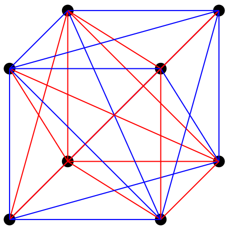
We wish to colour the edges of the cube such that the configuration K_4 never appears on a flat plane in all red or all blue.



The following configurations would therefore not be allowed as K_4 appears on a flat surface in one colour :



The configuration below however is valid :



3 The Original Problem

We define a configuration of coloured edges in any dimension as *bad* if it contains a monochromatic K_4, i.e., a complete subgraph on four vertices where all edges are either entirely red or entirely blue. In the previous section, we observed that in two dimensions, it is possible to construct a complete graph that is not *bad*. The same holds for three dimensions.

The problem asks to determine the smallest number of dimensions for which every possible edge coloring is necessarily *bad*, meaning that no matter how the edges are colored, a monochromatic K_4 is unavoidable.

$$\text{Number of verticies in an } n\text{-dimensional hypercube} = 2^n$$

$$\implies \text{Number of edges in the complete graph} = \binom{2^n}{2}$$

We let $E(\text{n}) = \binom{2^n}{2}$. The number of configurations with each edge being red or blue comes out to be $2^{E(n)}$

We can define $C(\text{n})$ to be the number of coloured configurations possible on an n-dimensional hypercube, where $C(n) = 2^{E(n)}$

n	$C(n)$	$\log(C(n))$
2	64	4.159
3	268435456	19.408
4	1.329×10^{36}	83.178
5	2.046×10^{149}	343.801

Note that in this section of the document, $\log$ **refers to the *natural* log with base** e**. Logs with base 10 will be represented via** log_{10}

Before we delve in $n > 5$, let's take a moment to appreciate how quickly the function $C(n)$ grows.

Already, $C(4) \approx 10^{36}$ is an incredibly accurate estimate of the total number of atoms in all 8 billion humans combined. But $C(5)$ is so large that it exceeds the estimated number of atoms in the entire universe by over 70 orders of magnitude! That is, we would need approximately 10^{70} universes like ours in order to accumulate $C(5)$ atoms.

Astonishingly, modern computers can still perform floating-point arithmetic using such extreme numbers, even though their size far surpasses any quantities we encounter in our day-to-day lives.

$C(n)$ for any $n > 5$ is so large that average computers fail to perform calculations on these numbers. Rather, the value is simply treated as infinity by their processers. In order to estimate values of $C(n)$ as n grows, we must use the indentity:

$$C(n) \equiv 2^{E(n)}$$

$$\implies \log(C(n)) \equiv E(n) \times \log 2, \text{ (Where } E(n) = \binom{2^n}{2})$$

$$\implies \log(C(n)) = \log 2 \times \binom{2^n}{2} \quad (1)$$

This allows us to find a value of the natural log of $C(n)$ without actually finding $C(n)$. In order to find an estimate for the actual value of $C(n)$ as an integer exponent of 10, we can convert all the natural logs to logs with base 10.

$$\implies \log_{10}(C(n)) = \log_{10} 2 \times \binom{2^n}{2}$$

Let $\alpha = \log_{10} 2 \times \binom{2^n}{2}$.

$$\implies \log_{10}(C(n)) = \alpha$$

$$\implies C(n) \approx 10^{\lfloor \alpha \rfloor} \quad (2)$$

Using equation (1) and (2), and also the help of Python code,:

n	$C(n)$	$\log(C(n))$
6	$\approx 10^{606}$	1397.4
7	$\approx 10^{2446}$	5633.9
8	$\approx 10^{9825}$	22624.3

The main thing to note here is that it is impossible to brute force the coloured configurations of hypercubes, even with the computational power of all global processors combined. Mathematicians must instead think extremely critically and determine if it is possible to avoid K_4 in a single colour somewhere in the hypercube's edges.

For all dimensions up to 12, mathematicians are certain that it is possible to colour the hypercube in such a way.. However, in 13 dimensions, it is *potentially* impossible to avoid such a *bad* configuration. Unfortunately, we are not yet completely certain whether $n = 13$ is indeed the smallest dimension where a *bad* configuration becomes unavoidable.

While no formal proof exists for 13 dimensions, there was an upper bound for the smallest value of n in which every possible colouring of the n-dimensional hypercube results in a *bad* configuration. This bound is famously known as Graham's Number.

The answer to the problem lied in the interval [13, Graham's Number].
(the upper bound has since been improved to a lower number.)

4 Knuth's Up Arrow Notation

4.1 Starting Small

Now that we have discussed the origin and use of Graham's Number, we can appreciate its unimaginable size. The bedrock for Graham's Number utilises the up-arrow notation, first thought of by computer-scientist Donald Knuth

Consider the expression $a \uparrow b$. This simply means a multiplied by itself repeatedly b times. Repeated multiplication is simply exponentiation. $\implies a \uparrow b = a^b$.

E.g : $3 \uparrow 3 = 3 \times 3 \times 3 = 3^3 = 27$

Now consider the expression $a \uparrow\uparrow b$. Just like how a single arrow is repeated multiplication, a double arrow is repeated single arrows. This is repeated exponentiation known as tetration.

$$a \uparrow\uparrow b = \underbrace{a \uparrow (a \uparrow (a \uparrow \ldots))}_{b \text{ times}}$$

Using $a \uparrow a = a^a$:

$$a \uparrow\uparrow b = \underbrace{a^{a^{a^{\cdots}}}}_{b \text{ times}}$$

Keep this definition in mind : $a \uparrow\uparrow b$ is simply a power tower of $a's$ with a height of b

Eg : $3 \uparrow\uparrow 3 = 3^{3^3} = 3^{27} = 7{,}625{,}597{,}484{,}987$

Before adding any more arrows, let's analyse our findings. A single arrow between two 3's gives a fairly small result of 27. To visualise it, $3 \uparrow 3$ seconds is about the duration of a short TV commercial. However, adding a second arrow yields a result of over 7.6 trillion. $3 \uparrow\uparrow 3$ seconds ago, the first Homo-sapiens had only just emerged, and were hundreds of millenia away from evolving into modern humans. We see that adding even a single arrow can explode numbers out of proportion.

4.2 A third arrow

With bravery, we can add an extra arrow, and evaluate $3 \uparrow\uparrow\uparrow 3$
Triple arrows can be represented as a repeated string of double arrows. In this case:

$$3 \uparrow\uparrow\uparrow 3 = 3 \uparrow\uparrow (3 \uparrow\uparrow 3)$$

We already know that $3 \uparrow\uparrow 3 = 7,625,597,484,987$,

$$\implies 3 \uparrow\uparrow\uparrow 3 = 3 \uparrow\uparrow (7,625,597,484,987)$$

Using the definition of double arrows being a tower of powers:

$$\implies 3 \uparrow\uparrow\uparrow 3 = \underbrace{3^{3^{3^{\cdots}}}}_{7.6 \text{ trillion times}}$$

It may be difficult to appreciate the magnitude of $3 \uparrow\uparrow\uparrow 3$ in this form, so here is a way to understand it.

Consider the number $3^{3^{3^3}}$ ($3 \uparrow\uparrow 4$). This number is so large that it contains over 3.6 trillion digits! This means that, even if we were to write out each digit at a rate of one per second, by the time we finished writing the final digit, the tectonic activity of the Earth would have already reshaped the geography beyond recognition. To even WRITE this number is impossible for any human.

Now let's analyse the much larger number $3^{3^{3^{3^3}}}$ ($3 \uparrow\uparrow 5$).
To fully understand the next section, consider a *googolplex* ($10^{10^{100}}$). This is simply a 1 followed by 10^{100} zeros. Since the universe 'only' has about 10^{80} atoms, there are vastly more digits in a *googolplex* than atoms in the universe. In other words, to even write a *googolplex*, (given infinite time) there simply is not enough space in the observable universe to write all 10^{100} zeros in this number.

We may now prove that $3 \uparrow\uparrow 5$ is vastly bigger than a *googolplex*:

Let $N = 3^{3^{3^{3^3}}}$, and $M = 10^{10^{100}}$

$$\implies \log_{10}(N) = \log(3^{3^{3^{3^3}}}) = (3^{3^{3^3}}) \times \log_{10} 3$$

We already know that $3 \uparrow\uparrow 4$ has 3.6 trillion digits so $\approx 10^{3,600,000,000,000}$

$$\implies \log_{10}(N) = 10^{3,600,000,000,000} \times \log_{10} 3$$
$$= 10^{3,600,000,000,000} \times \ 0.477$$

Furthermore:

$$\implies \log_{10}(M) = 10^{100} \times \log_{10} 10$$
$$= \log_{10}(M) = 10^{100}$$

Hence:

$$\log_{10}(N) = \log_{10}(3 \uparrow\uparrow 5) \gg \log_{10}(M) = \log_{10}(10^{10^{100}})$$

$$\therefore 3^{3^{3^{3^3}}} \gg 10^{10^{100}}, \text{By trillions of orders of magnitude}$$

We see that a number which cannot even *fit* in the universe when written out pales in comparison to $3 \uparrow\uparrow 5$

Note that this was only a power tower of 5 three's. In reality, $3 \uparrow\uparrow\uparrow 3$ was a tower of over 7.6 trillion three's, with each step down the tower taking us into a much bigger world. There is simply no way to truly capture its size.

We started with $3 \uparrow 3$, which equals 27, and jumped to $3 \uparrow\uparrow 3$, a number on the order of 7.6 trillion. A third expression, $3 \uparrow\uparrow\uparrow 3$, gave us a number so large that it dwarfs even a googolplex.

4.3 A fourth arrow

We will add one final arrow, and attempt to evaluate $3 \uparrow\uparrow\uparrow\uparrow 3$. Quadruple arrows is simply a repetition of triple arrows.

$$\implies 3 \uparrow\uparrow\uparrow\uparrow 3 = 3 \uparrow\uparrow\uparrow (3 \uparrow\uparrow\uparrow 3)$$

Remembering that triple arrows is repeated double arrows:

$$3 \uparrow\uparrow\uparrow\uparrow 3 = \underbrace{3 \uparrow\uparrow (3 \uparrow\uparrow \ldots (3 \uparrow\uparrow 3) \ldots))}_{3\uparrow\uparrow\uparrow 3}$$

$$\underbrace{3 \uparrow\uparrow 3 \uparrow\uparrow 3 \uparrow\uparrow \ldots 3 \uparrow\uparrow 3 \uparrow\uparrow 3}_{3\uparrow\uparrow\uparrow 3}$$

The coloured part $(3 \uparrow\uparrow 3)$ is equal to 7.6 trillion as seen before

$$\underbrace{3 \uparrow\uparrow 3 \uparrow\uparrow 3 \uparrow\uparrow \ldots 3 \uparrow\uparrow 3 \uparrow\uparrow 3}_{3\uparrow\uparrow\uparrow 3}$$

The coloured part is now $3 \uparrow\uparrow 7.6$trillion, which is $\underbrace{3^{3^{3^{\cdots}}}}_{7.6 \text{ trillion times}}$

$$\underbrace{3 \uparrow\uparrow 3 \uparrow\uparrow 3 \uparrow\uparrow \ldots 3 \uparrow\uparrow 3 \uparrow\uparrow 3 \uparrow\uparrow 3}_{3\uparrow\uparrow\uparrow 3}$$

The coloured part is now $3 \uparrow\uparrow (3^{3^{3^{\cdots}}})$, which is equal to $\underbrace{3^{3^{3^{\cdots}}}}_{\underbrace{3^{3^{3^{\cdots}}}}_{7.6 \text{ trillion}}}$

Note that what happens is that when we go from right to left, we start with 3, but this is the height of the power tower of the next number. This number is the height of the power tower of the NEXT number (remembering that $a \uparrow\uparrow b$ is a power tower of a's with height b). This process carries on, with each new number we calculate being the height of the NEXT power tower. We end up creating a tower of towers, and the height of this greater tower is $3 \uparrow\uparrow\uparrow 3$, which in itself is already unfeasible.

I will not bother trying to explain how gargantuan this number is, as there is simply no point. By the time we construct our fifth power tower, we have a number which dwarfs $3 \uparrow\uparrow\uparrow 3$ already. The speed at which the size of these towers grows is beyond anything I can think of.

To summarise:

$$3 \uparrow 3 = 27, \quad \text{a 5 year old could count to this number}$$

$$3 \uparrow\uparrow 3 \approx 7.6 \times 10^{12}, \quad \text{the number of seconds humanity has existed for}$$

$$3 \uparrow\uparrow\uparrow 3, \quad \text{cannot even be written in the universe given infinite time}$$

$$3 \uparrow\uparrow\uparrow\uparrow 3, \quad \text{if imagined in its entirety, your brain would collapse into a black hole.}$$

5 Reaching Graham's Number

This entire time, I have been emphasising how rapidly the number grows after adding just one arrow, and you are probably sick of hearing it. You will see why I keep mentioning this.

Let $g_1 = 3 \uparrow\uparrow\uparrow\uparrow 3$

g_1, which is already beyond fathomable, represents the number of ARROWS between two 3's in g_2!

$$g_2 = 3 \underbrace{\uparrow\uparrow\uparrow\uparrow\uparrow \text{......} \uparrow\uparrow\uparrow}_{3\uparrow\uparrow\uparrow\uparrow 3} 3$$

To even *attempt* to simply this:

$$g_2 = 3 \underbrace{\uparrow\uparrow\uparrow\uparrow\uparrow \text{......} \uparrow\uparrow\uparrow}_{g_1 - 1} (3 \underbrace{\uparrow\uparrow\uparrow\uparrow\uparrow \text{......} \uparrow\uparrow\uparrow}_{g_1 - 1} 3)$$

And this is the most we can really do.

Now, we define a recursive sequence. Let $g_{n+1} = 3 \underbrace{\uparrow\uparrow\uparrow\uparrow\uparrow \text{......} \uparrow\uparrow\uparrow}_{g_n} 3$

$$\implies g_3 = 3 \underbrace{\uparrow\uparrow\uparrow\uparrow\uparrow \text{......} \uparrow\uparrow\uparrow}_{3 \underbrace{\uparrow \text{......} \uparrow}_{3\uparrow\uparrow\uparrow\uparrow 3} 3} 3$$

Note that I am not even trying to explain how big these numbers are, as there is simply no point. Anything beyond this scale, the numbers are *practically* infinite. But we are not still even close We continue this process 64 times, until we reach g_{64}, and then finally, we reach the famous Graham's Number.

$$\text{Graham's Number } = g_{64} = 3 \underbrace{\uparrow\uparrow\uparrow\uparrow\uparrow \text{......} \uparrow\uparrow\uparrow\uparrow\uparrow}_{g_{63}} 3$$

This was the upper bound for the original problem. Since then, the bound has been reduced to $9 \uparrow\uparrow\uparrow 4$ (still an absurdly large number), but still an improvement.

6 Finding Graham's Number (kind of)

Mathematicians have cleverly used modular arithmetic to determine the last couple digits of Graham's Number. In order to determine the final k digits of g_{64}, we must work modulo 10^k.

Working modulo 10, we can prove that Graham's Number will end in a 7.

$$\text{Let } f_0(n) = 3^n \bmod 10.$$

$$\text{We see that } f_0(1) = 3, \quad f_0(2) = 9, \quad f_0(3) = 7, \quad f_0(4) = 1,$$

$$\text{and this sequence repeats with a period of 4.}$$

Thus, we can define $f_0(n)$ for all $k \in \mathbb{Z}^+$ as:

$$f_0(4k+1) = 3, \quad f_0(4k+2) = 9, \quad f_0(4k+3) = 7, \quad f_0(4k) = 1.$$

$$\text{Now let } f_1(n) = 3^{3^n} \mod 10.$$

Consider the case when $n = 4k + 1$ or $4k + 3$

$$\text{Since } f_0(4k+1) = 3, \implies 3^{4k+1} \equiv 7 (\text{mod } 10)$$

$$\text{Furthermore, } f_0(4k+3) = 7, \implies 3^{4k+3} \equiv 7 (\text{mod } 10)$$

Since $4k + 1$ and $4k + 3$ covers all odd integers (every odd integer is either 1 more or 3 more than a multiple of 7), $f_1(2k + 1) = 7$, $k \in \mathbb{Z}^+$

Now consider Graham's Number, which can be written in the form 3^N, for some large number N. N itself can be written in the form $N = 3^M$, so $g_{64} = 3^N = 3^{3^M}$. Since M is also a 3s power tower, it will not contain any factors of 2, and is therefore odd.

Since M is odd, $f_1(M) = 7$. From the definition of $f_1(n) = 3^{3^n} \mod 10$, it must follow that 3^{3^M} ends in a 7. It just so happens that $3^{3^M} = g_{64}$.

$$\therefore \text{The last digit of Graham's Number is a 7.}$$

Modular arithmetic for larger powers of 10 can get tedious, so it is ideal to use computers to minimize the effort.

Here are the steps in the algorithm to get the last k digits of Graham's Number:

1. Define $g_1 = 3 \uparrow\uparrow\uparrow\uparrow 3$, which represents a power tower of height 3 with base 3.
2. Recursively compute $g_n = 3 \underbrace{\uparrow\uparrow\uparrow\uparrow\uparrow \ldots\ldots \uparrow\uparrow\uparrow}_{g_{n-1}} 3$ for n = 2, 3 ,4 ...64.
3. Instead of computing G_{64} directly (which is impossible), compute it modulo 10^k to keep only the last k digits.
4. Use modular exponentiation to efficiently compute each step while maintaining only the last k digits.
5. Output the last k digits of Graham's number.

Here is the code written in *Python 3.13* which implements the algorithm.

```python
def tetration(base, height, mod):
    # Repeated exponentiation (power tower)
    if height == 1:
        return base % mod
    # Compute power step by step
    return pow(base, tetration(base, height - 1, mod), mod)

def get_digits(k):
    # Only keep the last k digits
    mod = 10**k
    return tetration(3, 64, mod)

k = int(input())
print(get_digits(k))
```

When running the code and inputting k=1, it indeed shows that the last digit is a 7.

I inputted 100 as the value of k, and in return, got the last 100 digits of Graham's Number:

....9079226639260620677321886132284373937967905496638003222348723967
0184851864390591045756272624641953 87

No matter how much computational power we accumulate, and regardless of how many trailing digits we identify, no one will ever determine and prove the leading digit of Graham's Number.

Finally, just because it was possible, I ran k=1000, and determined the final 1000 digits of Graham's Number:

4497164216564515110293688366488801885840362406281516352187527995851248870442048 9
4406191392518403719005058821239657261039265017522955213415326943251182155255020 3
2920513709145147370448730157387329035625349136149442703001326133544558947133169 0
2856088469029872613183097238999112980639401105896420353625985508105391185470005 6
7640466133519318576568837607807568308886172549185651696383359234359517363875801 3
3632938773322587092445345942913096478906095281766685618716899663820972358982750 5
6029768658637037179596957219138209777040833663170224271014929111936358483346015 5
3911953701324911817650841460630308466769655171264545034156593651121555649723002 4
7741473332678699784981625935234624040706234133426523936068499679177863383411711 3
1561310358229679784785728922975181226645706979188609578864246054293558096365018 7
0148988989482940240764912714167244682419676283143898793783776212601905182209658 3
6040030192364486918390792266392606206773218861322843739379679054966380032223487 2
396701848518643905910457562726246419538 7

7 Conclusion

Graham's Number held the title of the largest number ever used in a mathematical proof for several decades. However, in recent years, it has been surpassed by other massive numbers, such as TREE(3) and Rayo's Number, among others.

A common question that arises is: Why can't mathematicians simply add 1 each time a new large number is discovered, and let that become the "largest" number? The answer is that such a number would lose its utility. Graham's Number, for instance, was originally introduced as an upper bound in a specific mathematical proof. Increasing it arbitrarily would undermine its purpose and relevance in the context of the problem it was designed to address.

TREE(3), for example, is not just an arbitrary large number—it is used to describe the number of possible games that can be formed given 3 seeds and a set of rules. Similarly, Rayo's Number was defined within the context of first-order set theory for a specific purpose: to win a "big number contest." These numbers aren't just large for the sake of being large; they have particular applications and meanings within their respective fields.

One last thing to consider is that regardless of how ridiculously large Graham's Number seems, as $n \to \infty$, the probability that $n > g_{64}$ tends to 100 %.
Infinity is not a number in itself, but rather a concept that represents the idea of something continuing without limit, beyond any finite value, no matter how large.