

Counting Lattice Paths

Frank Rodriguez

May 4th, 2018

Contents

1	Introduction	1
2	Counting	2
2.1	Simple Counting	2
2.2	Law of Reduction	3
2.3	Law of Symmetry	5
3	Advanced Counting	6
3.1	Lattice Decomposition	6
3.2	Worked Example: $ 3 \times 3 $	8
4	A Total Solution	10
4.1	The $n \times 1$ Case	10
4.2	Sums of Sums of Sums...	12
4.3	Finite Calculus	13
4.3.1	The Falling Power	14
4.3.2	The Finite Integral	14
4.4	Expanding the Base Case	15
4.5	The Finite Pattern	17
4.6	Simplifying the Finite Pattern	19

1 Introduction

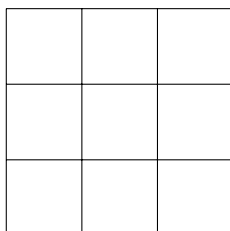


Figure 1: A lattice

This is a lattice. They are different from grids in that our focus is not on the squares themselves, but rather on the corners and edges. In the system we will discuss in this paper, the goal is to travel along the edges from the top left corner to the bottom right corner. However, there are only two legal moves: moving right, and moving down. Here are some example legal paths:

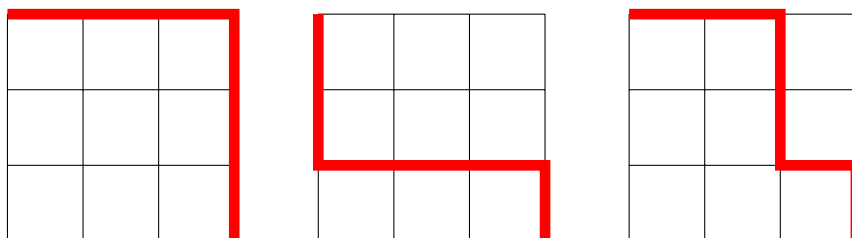


Figure 2: A few lattice paths

And here are some illegal paths:

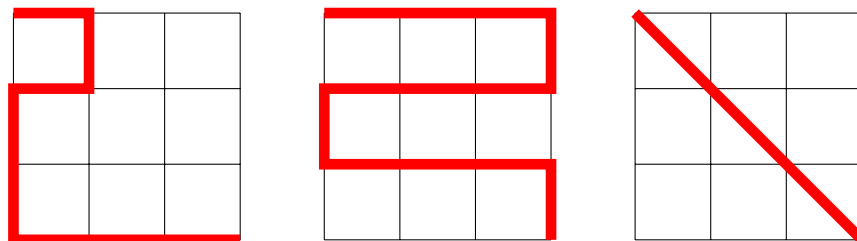


Figure 3: Some illegal lattice paths

The question then becomes, given any lattice, how can we calculate the number of legal paths it contains?

2 Counting

2.1 Simple Counting

The first idea one might have is to simply count the number of paths in a lattice by hand. There is nothing particularly flawed about this approach, and it can technically be used to solve any lattice. For example, here are the paths of a one-by-one lattice:



Figure 4: Counting all paths in a 1×1 lattice

Thus demonstrating that a 1×1 lattice has two paths. We will write this as $|1 \times 1| = 2$, or “the magnitude of a one-by-one lattice is two.” We can repeat this to solve for $|2 \times 2|$:

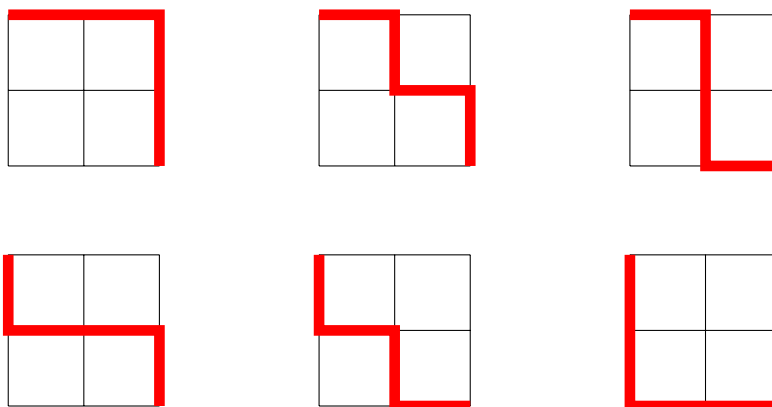


Figure 5: Counting all paths in a 2×2 lattice

This shows all six potential paths to solving our two-by-two lattice problem. What of a three-by-three lattice? While certainly possible to count all solution paths, it becomes quite easy to fumble. For even larger lattices, this becomes especially true. A better method of counting lattice paths is necessary.

2.2 Law of Reduction

Let us analyze a specific lattice case, the $n \times 1$ lattice. By starting small, we can count their paths with little difficulty and search for patterns. Starting with a 1×1 lattice, we see that it has two paths. If we append a square to it in order to create a 2×1 lattice, what changes?

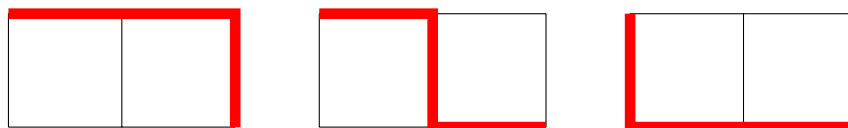


Figure 6: Counting all paths in a 2×1 lattice

While $|1 \times 1|$ is two, it seems as though adding another square added only a single path. Why is this?

In the 1×1 lattice, the only two paths were the outermost paths. As we are dealing with rectangular paths (paths which can be fully described as

$n \times p$), there will *always* be two outermost paths available. Thus, the 2×1 lattice also has these same two options. However, it has one new option – the edge formed where one square touches the other.

This explains why only one path was added when a new square was added. In fact, this pattern can be seen persisting as we add more and more squares to the lattice:

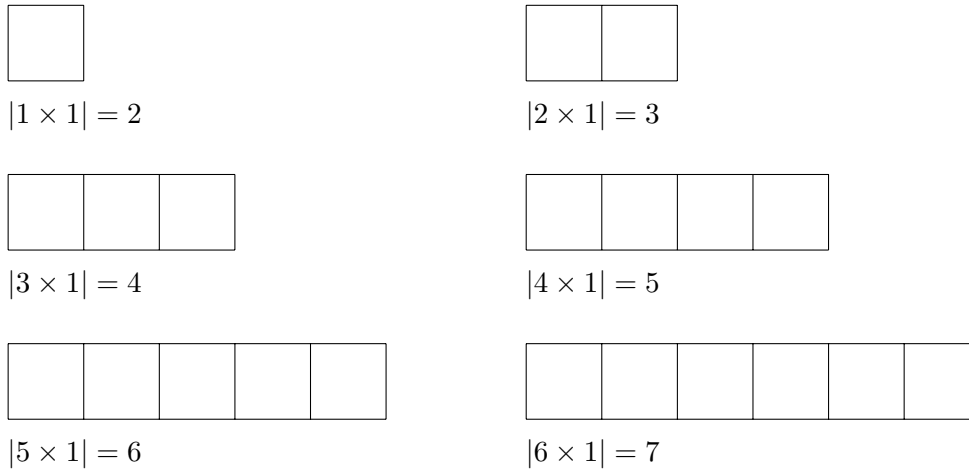


Figure 7: Succession of paths in $n \times 1$ lattices

The general pattern becomes clear, and we can generalize this rule to come up with our first “lattice law.” These will be a set of tools which will help us analyze lattices:

$$|n \times 1| = n + 1$$

2.3 Law of Symmetry

What if we rotate our lattices? Since they are rectangular, rotations of 180° will have no effect on the lattice – it would be indistinguishable from the original orientation. Therefore, only rotations 90° clockwise or counterclockwise need to be considered.

Square lattices are the exception, as any rotation would leave them indistinguishable from its original position.¹ We will start with the 2×1 lattice. Notice how rotating it 90° , be it clockwise or counterclockwise, results in the same shape of lattice – a 1×2 lattice (indeed, this is always the case):

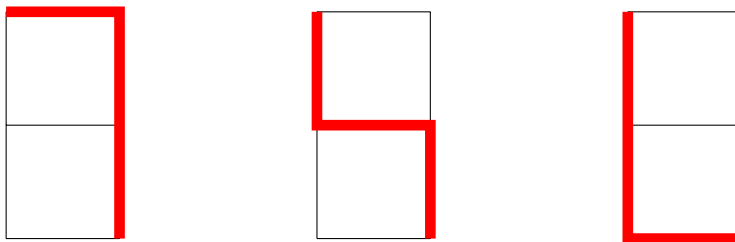


Figure 8: Counting all paths in a 1×2 lattice

The 1×2 lattice had three paths, and it appears that the 2×1 lattice also has three paths. Why?

We can represent a rotation as a switch in the placement of the numbers in the notation. In other words, a rotation of $n \times p$ can be written as $p \times n$. Note that, if $n \neq p$, then this new lattice is *not* the same as the original lattice, but do realize what the rotation has done. After rotation, all right path steps became down path steps, and all down path steps became right path steps. By writing out the rotation in terms of $n \times p$, this is much easier to see.

The significance of this is that, while $n \times p$ is not the same lattice as $p \times n$ for all $n \neq p$, the number of paths in the two lattices is the same. This gives us our second lattice law:

¹We only consider rotations in increments of 90° . In the context of the pathing problem we are solving, no other type of rotation makes sense.

$$|n \times p| = |p \times n|$$

This proves that our first lattice law applies to its rotated forms. In other words, because $|n \times 1| = n + 1$ is true, then $|1 \times n| = n + 1$ must also be true. It might be helpful to then reconsider our first lattice law to be:

$$|n \times 1| = |1 \times n| = n + 1$$

3 Advanced Counting

3.1 Lattice Decomposition

So far, we have two lattice laws:

$$\begin{aligned} |n \times 1| &= n + 1 \\ |n \times p| &= |p \times n| \end{aligned}$$

This helps us solve very easy lattices, but is still not powerful enough to tackle more complex ones. Consider our 2×2 lattice:

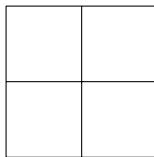


Figure 9: A 2×2 lattice.

We cannot count the number of paths it contains using *only* our lattice laws. We need a method of turning these larger lattices into more manageable chunks.

In order to do this, we need a realization. What happens when you move right?

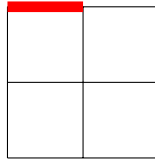


Figure 10: One step right.

Once you take a step to the right, you cannot move back, as left steps are not legal. Thus, when you move right, all possible paths are contained within the right side of the lattice:

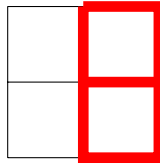


Figure 11: All possible consequences of moving right once.

Likewise, when you take a step down, all possible paths this can lead to are contained within the lower part of the lattice:

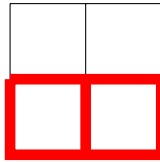


Figure 12: All possible consequences of moving down once.

Because these two pieces of the lattice contain all consequences to the final path from our first move, and because these two sets of paths are independent of each other, we can say that the whole lattice's paths are the sum of these two sub-lattice's paths.

When you move to the right, you end up with a 1×2 lattice. By the first law, this has three paths. When you move down, you end up with a 2×1 lattice. Again by the first law, this has three more paths.

Thus, $|2 \times 2| = |1 \times 2| + |2 \times 1| = 3 + 3 = 6$, and indeed this is what we counted earlier (figure 5).

We can generalize this further. Moving to the right in an $n \times p$ results in a sub-lattice which has the leftmost edge cut off. This is equivalent to an $(n - 1) \times p$ lattice.

Likewise, moving down in an $n \times p$ lattice produces a sub-lattice which has the top row cut off. This is equivalent to an $n \times (p - 1)$ lattice.

This pair of sub-lattices represents all possible paths in the parent lattice, giving us our third lattice law:

$$|n \times p| = |(n - 1) \times p| + |n \times (p - 1)|$$

Combined with the first two laws, we now have a high enough understanding of lattices to count the number of paths in more difficult configurations.

3.2 Worked Example: $|3 \times 3|$

First, a reminder of our lattice laws:

$$\begin{aligned} |n \times 1| &= n + 1 \\ |n \times p| &= |p \times n| \\ |n \times p| &= |(n - 1) \times p| + |n \times (p - 1)| \end{aligned}$$

Given this, counting the paths in a 3×3 lattice can be done graphically:

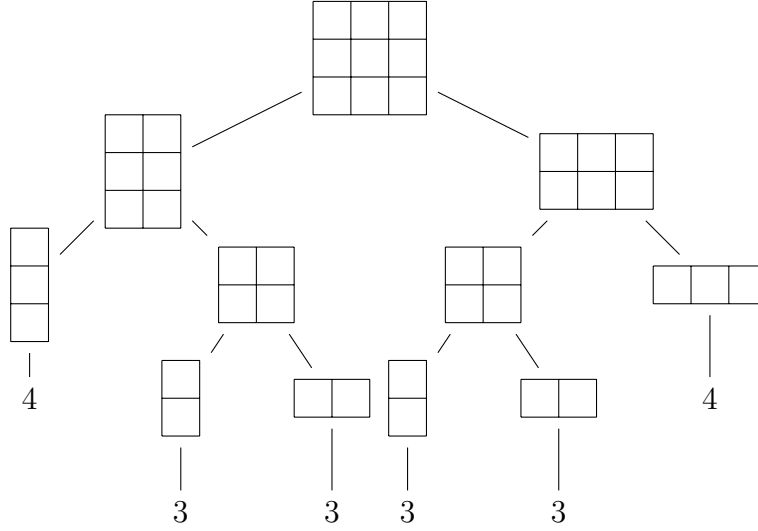


Figure 13: Graphical decomposition of a 3×3 lattice. The final answer is $4 + 3 + 3 + 3 + 3 + 4$, or 20.

It can also be done analytically:

$$\begin{aligned}
 |3 \times 3| &= |3 \times 2| + |2 \times 3| && \text{(Decomposition)} \\
 &= 2|3 \times 2| && \text{(Symmetry)} \\
 &= 2(|2 \times 2| + |3 \times 1|) && \text{(Decomposition)} \\
 &= 2(|2 \times 2| + 4) && \text{(Reduction)} \\
 &= 2(|1 \times 2| + |2 \times 1| + 4) && \text{(Decomposition)} \\
 &= 2(2(|1 \times 2|) + 4) && \text{(Symmetry)} \\
 &= 2(2(3) + 4) && \text{(Reduction)} \\
 &= 20 && \text{(Arithmetic)}
 \end{aligned}$$

This showcases the utility of the laws we derived. However, it also becomes unwieldy for even larger lattices, such as a 6×6 . While this method can make tackling lattices much easier to conceptualize than simply counting paths, it is still not the best way to tackle the task.

4 A Total Solution

4.1 The $n \times 1$ Case

The $n \times 1$ lattice case has a special property: we have a distinct formula for calculating the number of paths it contains, $|n \times 1| = n + 1$. As the size of the $n \times 1$ lattice increases, the difficulty in counting its paths does *not* increase. At a glance, we can determine that $|10,000 \times 1|$ is simply 10,001, or that $|999,999,999 \times 1|$ is simply 1,000,000,000. Can this seemingly unique case be applied to larger lattices?

Take for example an $n \times 2$ lattice. With our lattice laws alone, we have no shortcut, no formula that will immediately tell us the answer. We would have to decompose the lattice into smaller and smaller pieces, until we end with a series of $n \times 1$ lattices, which we can then extract real numbers from.

When doing this, can we find a pattern? Is there something more going on than simple decomposition? Let's try solving a 6×2 lattice, graphically:

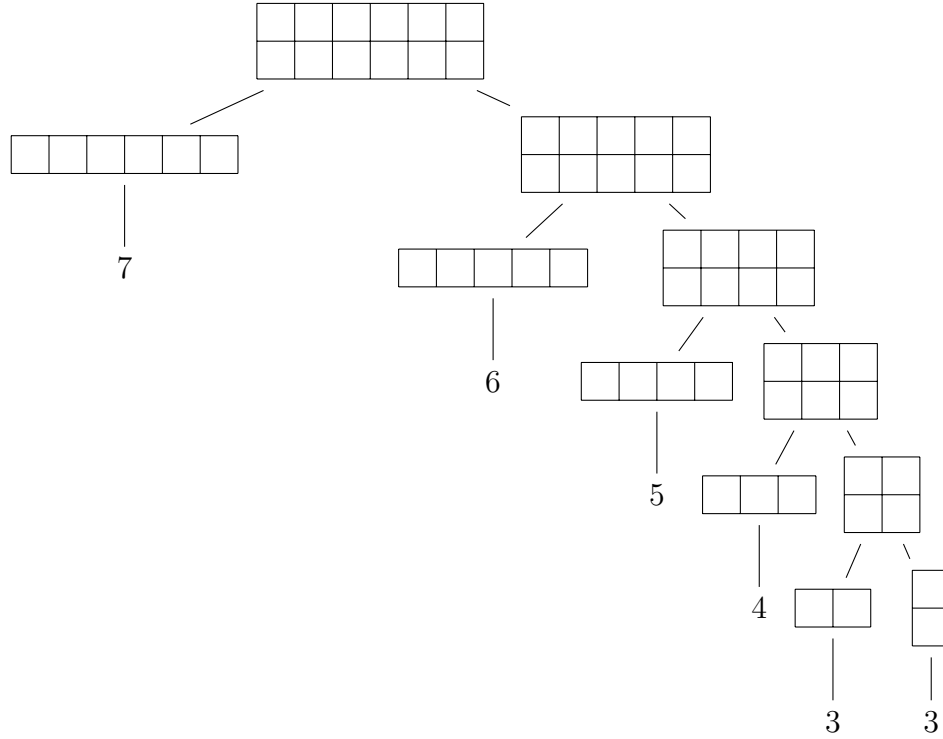


Figure 14: Graphical decomposition of a 6×2 lattice.

We end up with a neat little sum: $7 + 6 + 5 + 4 + 3 + 3$. This is even more significant when you realize that $3 = 2 + 1$, so we can rewrite the sum as $7 + 6 + 5 + 4 + 3 + 2 + 1$. This makes sense to always be the case: by the law of decomposition, decomposing any $n \times 2$ lattice will result in one $n \times 1$ lattice and one $(n - 1) \times 2$ lattice. This will continue until we end with a single 1×2 lattice, as we can then directly reduce it.

Thus, we have an interesting property. We can solve any $n \times 2$ lattice using a summation. In other words, $|n \times 2|$ will equal the sum of all numbers counting from 1 to $n + 1$. More succinctly:

$$|n \times 2| = \sum_{i=1}^{n+1} i$$

The logic here is sound, and the formula terse, but we can still do better.

4.2 Sums of Sums of Sums...

When we look at a partial decomposition of an $n \times 3$ lattice, we see another interesting pattern. For example, here is a 5×3 lattice:

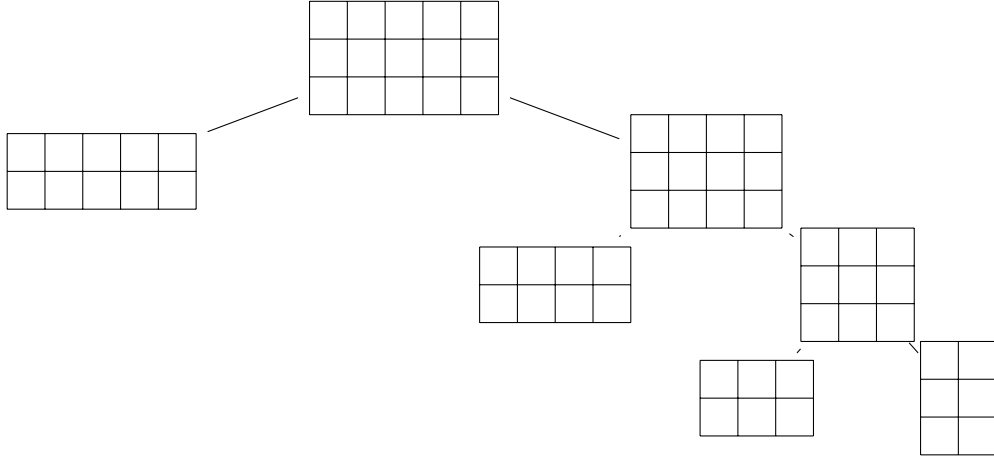


Figure 15: Partial graphical decomposition of a 5×3 lattice.

We find that an $n \times 3$ lattice can be broken up into a series of $n \times 2$ lattices. Again this can be shown directly from decomposition, as decomposing an $n \times 3$ lattice results in one $n \times 2$ lattice and one $(n - 1) \times 3$ lattice. Each $n \times 2$ lattice is a summation, as shown earlier. Thus, an $n \times 3$ lattice is a summation of summations.

As for the bounds, we end with a single 3×2 lattice and a single 2×3 lattice. However, just as with the $n \times 2$ lattices, we can again think of the last term differently.

The 2×3 lattice has 10 paths. If we treated it purely as the $n \times 2$ summations, we would finish the sequence with $\sum_{i=1}^3 i + \sum_{i=1}^2 i + \sum_{i=1}^1 i$, which is 10 – identical to what we counted. This means that it is safe to consider an $n \times 3$ lattice as a summation of summations, from 1 to n . In other words, our 5×3 lattice's paths could be written as:

$$\sum_{i=1}^6 i + \sum_{i=1}^5 i + \sum_{i=1}^4 i + \sum_{i=1}^3 i + \sum_{i=1}^2 i + \sum_{i=1}^1 i = 56$$

Figure 16: $|5 \times 3|$, expanded.

The “sum of sums” is now very clear. We can thus think of $|n \times 3|$ as being:

$$|n \times 3| = \sum_{j=1}^{n+1} \sum_{i=1}^j i$$

Similarly, an $n \times 4$ lattice will decompose into a series of $n \times 3$ lattices. Thus, $|n \times 4|$ can be thought of as a “sum of sums of sums” – quite a mouthful!

$$|n \times 4| = \sum_{k=1}^{n+1} \sum_{j=1}^k \sum_{i=1}^j i$$

The pattern continues, as an $n \times 5$ lattice decomposes into a series of $n \times 4$ lattice – a sum of sums of sums of sums!

4.3 Finite Calculus

These sums quickly become excruciating to calculate as they increase in complexity exponentially. Alas, they bear no resemblance to our beautifully unique case of $|n \times 1| = n + 1$, where increasing the size has negligible effect on the complexity of the solution.

Luckily, there is a way to extract such neat formulas from summations, a branch of mathematics known as “Finite Calculus.” The field is large, but we will only need to work with the parts regarding the transformation of sums of polynomials to closed-form functions. Let us look at some techniques.

4.3.1 The Falling Power

The first topic to cover is known as a “falling power.” Whereas a normal exponent is written as a^b , a falling exponent is written as $a^{\underline{b}}$. Falling powers are similar to normal exponents, but they “fall” by subtracting 1 each time. Consider $10^{\underline{5}}$:

$$10^{\underline{5}} = 10 \cdot (10 - 1) \cdot (10 - 2) \cdot (10 - 3) \cdot (10 - 4) = 10(9)(8)(7)(6) = 30,240$$

Or $9^{\underline{3}}$:

$$9^{\underline{3}} = 9 \cdot (9 - 1) \cdot (9 - 2) = 9(8)(7) = 504$$

Falling powers can be thought of as a factorial that terminates after a certain number of steps. It multiplies each number down to the power minus one: $a^{\underline{b}} = a(a - 1)(a - 2)(a - 3) \dots (a - (b - 1))$.

It is possible to translate regular exponents to falling exponents using something called *Stirling Numbers*. While interesting, we will only need to translate n^0 and n^1 to falling powers. Luckily, n^0 is equal to $n^{\underline{0}}$, and n^1 is equal to $n^{\underline{1}}$.

4.3.2 The Finite Integral

The infinite integral of a variable to a power works as follows:

$$\int x^n dx = \frac{x^{n+1}}{n+1} + C$$

A finite integral (also called a discrete integral) works in much the same way, but instead of an integral symbol, we are working with a finite integral symbol (which reflects the summation), and instead of a standard exponent, we are working with a falling exponent:

$$\sum n^{\underline{k}} \delta n = \frac{n^{\underline{k+1}}}{k+1} + C$$

That applies to an indefinite finite integral. For a definite finite integral, there is another very important distinction between an infinite integral and a finite integral: \sum_a^b would sum from a to $b - 1$, something which is usually unwanted. Thus, if one wishes to sum from a to b , he will have to take the finite integral from \sum_a^{b+1} .

As for constants, they work identically as they do in infinite integrals:

$$\begin{aligned}\int a \, dx &= x + C \\ \sum a \, \delta n &= n + C\end{aligned}$$

Figure 17: Comparing the infinite and finite integrals of constants.

4.4 Expanding the Base Case

We can now apply our knowledge of finite calculus. To find the closed-form solution to any $n \times 2$ lattice, we know we will be summing multiple $n \times 1$ lattices. We can thus take the definite finite integral of our $n \times 1$ formula, $n + 1$:

$$\begin{aligned}
|n \times 2| &= \sum_0^{n+1} n + 1 \delta n \\
&= \sum_0^{n+1} n^{\frac{1}{2}} + 1 \delta n \\
&= \frac{1}{2} n^{\frac{2}{2}} + n \Big|_0^{n+1} \\
&= \frac{1}{2} (n+1)^{\frac{2}{2}} + n + 1
\end{aligned}$$

Figure 18: Taking the finite integral of $n + 1$

And now we finally have a closed-form solution for the $n \times 2$ lattice. Simplifying it, we get:

$$\begin{aligned}
|n \times 2| &= \frac{1}{2} (n+1)^{\frac{2}{2}} + n + 1 \\
&= \frac{1}{2} (n+1)(n+0) + n + 1 \\
&= \frac{n(n+1)}{2} + n + 1 \\
&= \frac{(n+1)(n+2)}{2}
\end{aligned}$$

Figure 19: Simplifying the results.

Thus we can confidently say that $|n \times 2|$ is $\frac{(n+1)(n+2)}{2}$. While it isn't as clean as the $|n \times 1|$ case, it has the benefit of being easier to calculate than the summation – any calculator can handle it.

4.5 The Finite Pattern

Since we know that the formula for calculating $|n \times 3|$ results in a sum of multiple $|n \times 2|$ formulas, we can integrate the $|n \times 2|$ formula to find $|n \times 3|$:

$$\begin{aligned} |n \times 3| &= \sum_0^{n+1} \frac{1}{2}(n+1)^2 + n + 1 \delta n \\ &= \frac{1}{3} \frac{1}{2}(n+1)^3 + \frac{1}{2}n^2 + n \Big|_0^{n+1} \\ &= \frac{1}{6}(n+2)^3 + \frac{1}{2}(n+1)^2 + n + 1 \end{aligned}$$

Figure 20: Integrating the $|n \times 2|$ formula.

Likewise, we can take the finite integral of this new formula to derive the formula for $|n \times 4|$, then integrate that to find the formula for $n \times 5$, then $|n \times 6|$, etc. Listing them out, we find:

$$\begin{aligned} |n \times 1| &= n + 1 \\ |n \times 2| &= \frac{1}{2}(n+1)^2 + n + 1 \\ |n \times 3| &= \frac{1}{6}(n+2)^3 + \frac{1}{2}(n+1)^2 + n + 1 \\ |n \times 4| &= \frac{1}{24}(n+3)^4 + \frac{1}{6}(n+2)^3 + \frac{1}{2}(n+1)^2 + n + 1 \end{aligned}$$

Figure 21: The first four complete formulas.

A pattern can be seen. To make it more clear, we can rewrite these as follows:

$$\begin{aligned}
|n \times 1| &= \frac{1}{1!}n^1 + \frac{1}{0!}n^0 \\
|n \times 2| &= \frac{1}{2!}(n+1)^2 + \frac{1}{1!}n^1 + \frac{1}{0!}n^0 \\
|n \times 3| &= \frac{1}{3!}(n+2)^3 + \frac{1}{2!}(n+1)^2 + \frac{1}{1!}n^1 + \frac{1}{0!}n^0 \\
|n \times 4| &= \frac{1}{4!}(n+3)^4 + \frac{1}{3!}(n+2)^3 + \frac{1}{2!}(n+1)^2 + \frac{1}{1!}n^1 + \frac{1}{0!}n^0
\end{aligned}$$

Figure 22: The first four complete formulas, rewritten in terms of falling powers, n , and factorials.

Everything neatly counts down, starting from the non- n term of the lattice. This behavior can be fully described by a single summation, a total function which solves any lattice:

$$|n \times p| = \sum_{i=0}^p \frac{1}{i!} (n + (i-1))^i$$

Figure 23: A solution for any lattice.

4.6 Simplifying the Finite Pattern

There is one final simplification to be had, and we are done:

$$\sum_{i=0}^p \frac{1}{i!} (n + (i - 1))^i = \binom{n+p}{n}$$

Thus, the final answer which can solve for the magnitude of any lattice is as follows:

$$|n \times p| = \binom{n+p}{n}$$

■