Calculating the See & Say Sequence

Jake Saville

November 2017

1 Introduction

The See & Say Sequence[1] is a sequence of integers starting with 1 and from there, visually describing each term to get the next. For example, the first term, 1, contains 1 of the digit 1 and as such the next term is 11. This has 2 of the digit 1 and so the next term is 21. This has 1 of the digit 2 and 1 of the digit 1, and so the next term is 1211 and so on. Table 1 below shows the first few terms of the sequence.

Table 1: First 8 terms of the See & Say Sequence in Base 10

1	11	21	1211	111221	312211	13112221	1113213211

As with any pattern that relies on the way in which the numbers are written, this sequence will be different depending on the base. Table 1 only shows the base 10 version of the sequence. For example, in base 2, the second term, 11 (3 in base 10), is the same, but the third is different. Since there are 2 of the digit 1 in the second term, and 2 in base 2 is written as 10, the next term in base 2 would be 101. When translated back into base 10, this term is 5. Table 2 shows the first few terms in the base 2 version of the sequence (translated back into base 10).

Table 2: First 8 terms of the See & Say Sequence in Base 2

1	3	5	59	245	2491	235253	127756731

The purpose of this paper is to define a set of analytic functions $f_n(x)$ whose input x is some positive integer and whose output is the number associated with the visual description of x in base n. For example, $f_{10}(1211) = 111221$ and $f_2(59) = 245$. In order to define this set of functions, we must first define some other helper functions.

2 Helper Functions

2.1 Checking Equality

For these helper functions, we will allow for function names with multiple letters so as to improve readability. We can define a general function $equals(x, y)$ which returns 1 if $x = y$ and 0 otherwise. We can do this by checking whether the difference between x and y is 0. We can do this by first dividing this difference by itself plus 1. If this distance is zero, the result will be zero. If the distance is greater than zero, this value will be between 0 and 1 (non-inclusive). We can then round up to the next integer (giving 0 when the difference is zero and 1 otherwise), and subtract the result from 1 (giving 1 when the difference is zero and 0 otherwise).

$equals(x, y) = 1 - \lceil \frac{|x-y|}{|x-y|+1} \rceil$

[1]OEIS A005150

2.2 Getting the Number of Digits

In order to get the number of digits in a base n number, we need to use the powers of n. When you raise n to a power between $m-1$ (inclusive) and m (non-inclusive), you will get an m digit base n number. We need to reverse this process such that from that given m digit number, x, we can calculate m. To solve for $m-1$, round down the base n log of x to the next integer. To solve for m, add 1 to the result.

$numberofdigits_n(x) = \lfloor log_n(x) \rfloor + 1$

2.3 Getting a Particular Digit

2.3.1 Right to Left

Now we need to define a function which allows us to fetch a particular digit from a base n number. This function counts digits from right to left and - like all functions defined herein with respect to a particular digit of a number - indexes from 0 (i.e. the 0th digit is the right-most digit). In order to do this, first divide the number by the base to the power of the index of the digit you want to receive. This will result in the number being shifted such that the digit you want is in the unit's place. Round down to the next integer to truncate everything after the decimal point (or base n point), and then take the remainder after diving by the base to isolate the required digit.

$getdigitrtl_n(x,k) = \lfloor \frac{x}{n^k} \rfloor \bmod n$

2.3.2 Left to Right

For the purposes of this paper, it is more useful to get the kth digit of a number from left to right (with the 0th digit being the left-most). This is a relatively simple adjustment as we just need to find the RTL index from the LTR index by subtracting it from the total number of digits in the number. We also need to subtract an additional 1, because we are indexing from 0, and so the greatest index is the number of digits-1. Once we've converted the LTR index into an RTL index, we can just get the digit using the RTL function

$getdigitltr_n(x,k) = getdigitrtl_n(x, numberofdigits_n(x) - k - 1)$

3 Utility Functions

The functions defined in this section are designed specifically to serve a singular purpose in solving this problem.

3.1 Checking for Homogeneity

Since this problem requires the ability to identify strings of the same character, it is useful to define a function which, with a specified base n number, x, and two indices, j and k, can determine whether all of the digits between those two indices (inclusive) are the same. We can achieve this by checking whether each of the digits between the indices are equal to the digit at the first index. We can achieve this via the *equals* function and the *getdigitltr* function. If we have an iterator variable i which goes through every index between j and k (inclusive), we can get the two digits which we need to compare with the expressions $getdigitltr_n(x,i)$ (the current digit being checked) and $getdigitltr_n(x,j)$ (the first digit in the range). We can then form an expression of the form $equals(getdigitltr_n(x,i), getdigitltr_n(x,j))$. This expression will yield 1 if the current digit at index i matches the first digit in the range, and 0 if it doesn't. As such, if we take the product of this expression over all integer values of i between j and k (inclusive) then we will have a value of 1 if every digit in the range is the same, and a value of 0 otherwise. This is the function which tells us if the range is homogeneous.

$israngehomogeneous_n(x,j,k) = \prod_{i=j}^{k}(equals(getdigitltr_n(x,i), getdigitltr_n(x,j)))$

3.2 Calculating Length of a Homogeneous String

We will now define a function which, given a base n number, x, and an index k, will give the number of consecutive digits starting from the digit at k (inclusive) which are the same as the digit at k. Such a string of consecutive digits which are all the same will be referred to as a homogeneous string. In order to calculate this, we can generate every possible range of indices starting at k and ending at some index between k and the end of x (inclusive) and see how many of the ranges are homogeneous. If the homogeneous string is of length a, then

a of those ranges will be homogeneous. Since the function to check for the homogeneity of a range returns 1 when the range is homogeneous and 0 otherwise, we can take the sum of the value of this function over all the valid end points of the range. As stated above, these end points can be between k and the number of digits in x subtract 1 (the index of the last digit in x)

$lengthofstring_n(x,k) = \sum_{i=k}^{numberofdigits_n(x)-1}(israngehomogeneious_n(x,k,i))$

It is important to note that this will not always give the total length of a homogeneous string, but will instead give the length of it that occurs on or after the digit at k. We will later define a function for determining whether a given value of k actually lies at the start of a homogeneous string.

3.3 Creating a Chunk

The See & Say Sequence itself relies largely on a structure that will be referred to as a chunk. This is comprised of a number representing how many of a given digit there are in a row, followed by that digit itself. The function defined above gives us this first number. In order to concatenate the digit itself, we need to first shift the length of the string to the left by one place, leaving the unit's place free to hold the digit itself. This can be achieved by multiplying it by the base n. Then, simply add the value of the digit.

$chunk_n(x,k) = lengthofstring_n(x,k)n + getdigitltr_n(x,k)$

This will yield the number of consecutive digits at or after index k in the base n number x which are all the same, followed by the digit itself. Again, it is important to note that this will work at any valid index, not just those at the start of a homogeneous string.

3.4 Finding the Start of the Next Homogeneous String

In order to find out whether a given value of k is at the start of a homogeneous string, we must first calculate for a given value of k, at what index the next homogeneous string begins. If k is indeed the start of a homogeneous string, the result will be equal to k itself. In order to calculate this, we add to $k-1$ the length of the remainder of its homogeneous string. We use $k-1$ because the indices index from 0.

$nextstring_n(x,k) = k + lengthofstring_n(x,k-1) - 1$

3.5 Checking Whether an Index is at the Start of a Homogeneous String

As previously stated, if a value of k is indeed at the start of a homogeneous string (and so the chunk obtained by it is valid), the index of the start of the next homogeneous string will be equal to k itself. As such, the function for determining a chunk's validity can be written as follows:

$validchunk_n(x,k) = equals(k, nextstring_n(x,k))$

4 Concatenating Chunks

Using the tools defined above, we can generate a list of chunks which take the form of the number of a specific digit which appear in a row, followed by that digit itself. We also have a way of telling which of those chunks are actually valid (which ones are measuring the whole string, and not starting part-way through). We now need to deal with the issue of concatenation.

5 General Concatenation

We can start with a more general case in which we have a list of base n numbers $a_0, a_1, \ldots, a_p$ and we want to concatenate them in order from a_0 to a_p. We can achieve this by shifting each of them to the left by the combined number of digits in each of the numbers after it, and adding all of these results together. If we wanted to calculate the shifted version of a_k, then we need to calculate the combined number of digits in a_{k+1} to a_p (inclusive). We can do that using the *numberofdigits* function, and so an expression for this might look like

$\sum_{j=k+1}^{p}(numberofdigits_n(a_j))$

and in order to shift a_k by this many places to the left, we need to multiply it by n raised to this power as such.

$$a_k n^{\sum_{j=k+1}^{p}(numberofdigits_n(a_j))}$$

In order to calculate the total concatenation, we need to sum over this expression for all integer values of k between 0 and p (inclusive). An expression for this would take the following form:

$$\sum_{k=0}^{p}(a_k n^{\sum_{j=k+1}^{p}(numberofdigits_n(a_j))})$$

6 Selective Concatenation

Our case is slightly more complicated, as we don't need to concatenate the chunk for every value of k, but only ones for which $validchunk_n(x,k)=1$. As such, we need to insert a mechanism by which a chunk without this property has no effect on the sum. Since the $validchunk$ function will only ever yield a 0 or a 1, multiplying it by the value of the chunk will yield the value of the chunk when the chunk is valid, and 0 otherwise. The result of this will be referred to as the real value of the chunk. We need to utilize this method for selective evaluation of a chunk in two places in the concatenation process. The first is when shifting the value of a chunk. We need instead to shift the value of the real value of the chunk. This will result in this entire step in the sum being equal to zero if the chunk is invalid. The other place is when calculating the total number of digits in later chunks. We want instead to add up only the number of digits in the valid chunks, and so we multiply the length of each chunk by the value of the $validchunk$ function so that for invalid chunks, 0 is added. As such, the expression looks like this:

$$\sum_{k=0}^{numberofdigits_n(x)-1}(validchunk_n(x,k)chunk_n(x,k)n^{\sum_{j=k+1}^{numberofdigits_n(x)-1}(validchunk_n(x,j)numberofdigits_n(chunk_n(x,j)))})$$

This selective concatenation of only valid chunks is the definition of our function - a function which, given a base n number, x, finds all of the consecutive strings of the same digit and concatenates each of their lengths followed by the digits themselves, one after the other. In other words, we have found our function f_n.

$$f_n(x)=\sum_{k=0}^{numberofdigits_n(x)-1}(validchunk_n(x,k)chunk_n(x,k)n^{\sum_{j=k+1}^{numberofdigits_n(x)-1}(validchunk_n(x,j)numberofdigits_n(chunk_n(x,j)))})$$

7 Substitutions

There is one major issue with the definition above, and that is that it is defined in terms of several other functions which would have to be defined how they have been in this paper wherever this function is to be used. As such, below I will substitute the definitions of these functions back into our definition of f_n. Fully substituting every function gives the following:

$$\begin{array}{ll}
 & \lfloor \log_n(x)\rfloor\, k{=}0((1-\lceil \frac{\left|k-(k+(\sum_{i=k-1}^{\lfloor log_n(x)\rfloor}(\prod_{j=k-1}^{i}((1-\lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-j}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor+1-k}}\rfloor \bmod n\right|}{1+\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-j}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor+1-k}}\rfloor \bmod n\right|}\rceil))))-1)\right|}{1+\left|k-(k+(\sum_{i=k-1}^{\lfloor log_n(x)\rfloor}(\prod_{j=k-1}^{i}((1-\lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-j}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor+1-k}}\rfloor \bmod n\right|}{1+\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-j}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor+1-k}}\rfloor \bmod n\right|}\rceil))))-1)\right|}\rceil)((\sum_{i=k}^{\lfloor log_n(x)\rfloor}(\prod_{j=k}^{i}((1-\lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-j}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-k}}\rfloor \bmod n\right|}{1+\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-j}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-k}}\rfloor \bmod n\right|}\rceil))))n+\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-k}}\rfloor \bmod n) \\
f_n(x)= & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \sum \\
 & \sum_{j=k+1}^{\lfloor \log_n(x)\rfloor}(1-\lceil \frac{\left|j-(k+(\sum_{i=j-1}^{\lfloor log_n(x)\rfloor}(\prod_{l=j-1}^{i}((1-\lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-l}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor+1-j}}\rfloor \bmod n\right|}{1+\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-l}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor+1-j}}\rfloor \bmod n\right|}\rceil))))-1)\right|}{1+\left|j-(k+(\sum_{i=j-1}^{\lfloor log_n(x)\rfloor}(\prod_{l=j-1}^{i}((1-\lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-l}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor+1-j}}\rfloor \bmod n\right|}{1+\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-l}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor+1-j}}\rfloor \bmod n\right|}\rceil))))-1)\right|}\rceil \log_n((\sum_{i=j}^{\lfloor log_n(x)\rfloor}(\prod_{l=j}^{i}((1-\lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-l}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-j}}\rfloor \bmod n\right|}{1+\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-l}}\rfloor \bmod n-\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-j}}\rfloor \bmod n\right|}\rceil))))n+\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor-j}}\rfloor \bmod n)+1) \\
 & n \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad)
\end{array}$$

8 Conclusion

The below is the definition for a pure, standalone set of analytic functions with the properties outlined in the introduction. Running this function on any term in the base n See & Say Sequence yields the next term in the base n See & Say Sequence.

$$f_n(x) = \sum_{k=0}^{\lfloor \log_n(x)\rfloor} ((1 - \lceil \frac{\left| k - (k + (\sum_{i=k-1}^{\lfloor log_n(x)\rfloor} (\prod_{j=k-1}^{i} ((1 - \lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - j}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor + 1 - k}} \rfloor \bmod n\right|}{1 + \left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - j}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor + 1 - k}} \rfloor \bmod n\right|} \rceil)))) - 1)\right|}{1 + \left| k - (k + (\sum_{i=k-1}^{\lfloor log_n(x)\rfloor} (\prod_{j=k-1}^{i} ((1 - \lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - j}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor + 1 - k}} \rfloor \bmod n\right|}{1 + \left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - j}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor + 1 - k}} \rfloor \bmod n\right|} \rceil)))) - 1)\right|} \rceil)((\sum_{i=k}^{\lfloor log_n(x)\rfloor} (\prod_{j=k}^{i} ((1 - \lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - j}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - k}} \rfloor \bmod n\right|}{1 + \left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - j}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - k}} \rfloor \bmod n\right|} \rceil))))n + \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - k}} \rfloor \bmod n)$$

$$n^{\sum_{j=k+1}^{\lfloor \log_n(x)\rfloor} (1 - \lceil \frac{\left| j - (k + (\sum_{i=j-1}^{\lfloor log_n(x)\rfloor} (\prod_{l=j-1}^{i} ((1 - \lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - l}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor + 1 - j}} \rfloor \bmod n\right|}{1 + \left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - l}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor + 1 - j}} \rfloor \bmod n\right|} \rceil)))) - 1)\right|}{1 + \left| j - (k + (\sum_{i=j-1}^{\lfloor log_n(x)\rfloor} (\prod_{l=j-1}^{i} ((1 - \lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - l}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor + 1 - j}} \rfloor \bmod n\right|}{1 + \left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - l}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor + 1 - j}} \rfloor \bmod n\right|} \rceil)))) - 1)\right|} \rceil \log_n((\sum_{i=j}^{\lfloor log_n(x)\rfloor} (\prod_{l=j}^{i} ((1 - \lceil \frac{\left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - l}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - j}} \rfloor \bmod n\right|}{1 + \left|\lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - l}} \rfloor \bmod n - \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - j}} \rfloor \bmod n\right|} \rceil))))n + \lfloor \frac{x}{n^{\lfloor log_n(x)\rfloor - j}} \rfloor \bmod n) + 1)})$$