The 196 Problem

Matthew Houser

Abstract: If you take any two-digit integer, reverse its digits, and add it back to the original number, after enough iterations you will eventually sum to a palindromic number, in which the integer and its digits reversed are equal to each other. As you expand this to three-digit integers, the first integer that seemingly does not work is 196. Any number that can be proven to never converge to a palindrome is said to be a Lychrel number. In this paper, we explore the differences between candidate Lychrel and non-Lychrel numbers to try to better understand why numbers like 196 seemingly never converge to a palindrome. In doing so, we use an advanced algorithm to only check possible seed numbers, expand to bases other than base 10, and use Monte Carlo simulation to estimate properties of Lychrel and non-Lychrel numbers where it is otherwise technologically infeasible to check every integer.

Introduction:

Using the reverse add algorithm all two-digit numbers will eventually converge to a palindrome, and 196 is the first integer than seemingly does not. A number that will never converge is called a Lychrel number, but because 196 has never been proven to be a Lychrel number, it is instead called a candidate Lychrel. While 196 is the first Lychrel number in base 10, it is not the only one. For example, 295, 394, 493, 592 and 691 are also candidate Lychrel numbers. These numbers also generate the same sequences, or thread, as 196 and are the called first order kin numbers for the seed 196. But 196 is not the only three-digit seed, as 879 is also a seed that generates a unique thread. As the number of digits increase, so too do the number of seeds. There are 3 seeds in four-digit integers, 69 in five-digit integers, 99 in six-digit integers, and 1,728 in seven-digit integers. Heuristically, there appears to be no shortage of seeds as the number of digits increases, but of particular importance to us is 196, since it is the smallest seed.

History of the Problem:

The 196 Problem has been known since at least 1938, when D.H. Lehmer published an article in the now defunct Belgian recreation mathematics magazine *Sphinx*. In this article, Lehmer noted that no palindromes occur in less than 74 iterations for 196, and no palindromes occur in less than 76 iterations for 1997. In a 1967 article mentioning Lehmer’s work, the author mentioned that using an IBM 1401 computer, the highest number of iterations currently tested was 3,556. Since the advent of personal computer, the 196 Problem has been a hobby for professional and amateur computer scientists and mathematicians alike. In 1984, the computer magazine *Ahoy!*, aimed at users of the Commodore 64, challenged their readers to send in a solution to the 196 problem, and a year later a reader wrote back that after 28 days of searching and 12,954 iterations, no palindromes were found. An article in Scientific American about how computers use large numbers mentions the 196 Problem, and states that no palindrome was found within 50,000 iterations, and that at that point the integers are 21,000 digits long.

Since then, amateur computer scientists, such as John Walker, Jason Doucette, Wade Van Landingham, and Romain Dolbeau, have applied the reverse add algorithm to enormous lengths. In 1990, John Walker was the first to calculate an iterate of 196 to one million digits. Jason Doucette calculated an iterate to 13 million digits in 2000, which took 289 days to compute and was written about in the Canadian kid’s science magazine *Yes!*. Interest in the problem has dwindled since 2000, and recently has mostly consisted of using more efficient code and faster computers to compute more digits. In 2011 Romain Dolbeau applied the algorithm one billion times, resulting in a 413-million-digit integer.

Initial approach:

We began our research by implementing a reverse add algorithm in python, and then testing all numbers up to 1,000,000. Any integer that did not converge in less than 100 iterations was deemed a candidate Lychrel number. We chose 1,000,000 as a cutoff point due to limitations in hardware, though later we will discuss a workaround to this problem. We found that about 88% of the integers were a palindrome or converged to a palindrome, and about 12% were candidate Lychrel numbers. Figure 1 shows a histogram of the relative frequency of the number of iterations for integers that do eventually converge. The most common number of iterations was 2, which about 22% of all integers in this range needed to converge to a palindrome. The most common number of iterations was 1, which about 17% of integers needed to converge to a palindrome, followed by 3 with around 13%. Figure 2 has the same information as Figure 1, but in the form of a cumulative frequency graph. About 52% of all integers converge to a palindrome within 3 iterations, 75% within 6 iterations, 91% within 12 iterations, and 99% within 30 iterations.

Figure 3 shows the number of iterations to reach a palindrome for each integer. Candidate Lychrel numbers were plotted with a value of -2 for illustrative purposes. The green points start with the seed 150,296 and include all 125 kin numbers, which converge to the palindrome 682,049,569,465,550,121,055,564,965,940,286 in 64 iterations. The purple points start with the seed 336,999 and include all 195 kin numbers, which converge to the palindrome 4,668,731,596,684,224,866,951,378,664 in 51 iterations. The point of graphing the Lychrel candidates is as a basic check that there are no holes on the x axis. The seeds are graphed to illustrate the point that much of the information plotted is redundant; once we know where a seed is located, we know all the information necessary about its first order kin.

Our next step was to follow up on the distribution of Lychrel numbers for a given number of digits. Plotting them in Figure 3 showed that there were no holes, but there were so many points that we received no other useful information. Due to the technical limitations of checking every single integer in a given range, we instead use Monte Carlo simulation. For a specified number of digits, we generated a random integer with python’s random module. We then tested if this integer converged within 200 iterations, and if it did not, we incremented a counter. This counter was then divided by the number of trials in order to find the share of Lychrel numbers. We repeated this process for all integers up to 30 digits with 100,000 trials. A graph of the results is shown in Figure 4. The results show that as the number of digits increase, the share of integers with that many digits that converge to a palindrome decrease. It is important to note that there will always be palindromes, regardless of how many digits there are, and there will always be non-palindromes that converge to palindromes after one iteration. For example, for an even number of digits n, the integer 333…444, where the first n/2 digits are 3 and the rest are 4, will always converge to the palindrome 777…777 after one iteration since there are no carries.

Our next Monte Carlo simulation was to find 100,000 numbers that converged to a palindrome for any given number of digits, and then comparing the number of iterations it took to converge between digit lengths. Figure 5 shows the results for 5, 10, 15, and 20-digit integers. Most five-digit integers converge within relatively quickly, with 25% converging in 2 iterations, and 60% converge within 3 iterations. In contrast, the 10, 15, and 20-digit integers have shorter bins. Interestingly, there is not much difference between the 10, 15, and 20-digit histograms. The relative frequency peaks with 2 iterations at about 15% and then trails off. One interesting piece of information is that for the relative frequency increases slightly from four to five iterations. Given that there are 100,000 samples, and that the pattern occurs for the digit lengths greater than 10, this does not seem to just be chance, and could make an interesting topic for further research.

Filtering out kin numbers:

As shown in figure 3 by the green and purple points, the majority of the points are kin numbers. Being able to only test seed numbers would be a large breakthrough in testing for seeds. In order to do this, we first need to formalize this problem.

Definition 1: Thread

A thread is any sequence of integers following the reverse add algorithm. A thread ends if the last entry is palindromic and begins with a seed.

Definition 2: Seed

A seed is the smallest integer that generates a thread.

Definition 3: Kin

A Kin is any integer that generates a thread but is not its seed.

Take for example the thread generated by 196: (196, 887, 1675, 7436, 13783, …). 196 is the seed, since it is the smallest integer, that following the reverse add algorithm, sums to 887. 295 is a kin number to 196, since 295 also sums to 887, but is larger than 196. Additionally, 689 is a kin number because it sums to 1675, which is the third entry in the sequence. This leads us to the next definition:

Definition 4: First Order Kin

A kin number is said to be first order if you can add and subtracting a constant to a pair of digits to arrive at a possible seed.

295 is a first order kin number because subtracting 1 from the hundreds digit and adding 1 to the ones digit arrives at 196. The integer 689 is second order kin, since it is first order kin to 887, the second element of the thread.

Our approach is to filter out all the integers that cannot possibly be seeds. What this means is that we want the first digit to be one, but we are constrained by the last digit, as whatever constant we add to it cannot be larger than 10. For example, the integer 3,426,790 cannot be a seed, because you can subtract 2 from the millionths digit and add 2 to the ones digit to arrive at 1,426,792, which will sum to the same thing using the reverse add algorithm. The conversion algorithm to find a seed is as follows:

Kin to Seed Algorithm:

Let A be any integer with digits

Then, select , where and .

A possible seed related to the first order kin A will then be

Then , where and

Continue for

Generalizing the above algorithm, we see that possible seeds have certain properties, specified below.

Definition: Possible seeds

Let represent any digit pair where and . An integer A is a possible seed if the first digit pair , and any other digit pair

By only selecting and testing possible seeds, we are able to test all possible seeds up to 10 digits. This is a huge improvement in the scope of the algorithm. There are, however, some limitations in our implementation. First, we only filter out first order kin, and not second or higher order. This means that 689 is still included in our search, even though it is second order kin to 196. Additionally, all integers found in a thread might also be tested, even though we already know where they should belong. For instance, 13783 is found in the 196 thread, and is kin to 12,793. However, 12,793 still fits our description for a possible seed, so our algorithm will test it even though we already know its properties from the seed 196.

By implementing this algorithm, we are massively simplifying the scope of our search. By following the above definition, we can formulize how many integers need to be tested with:

This is because there are 18 possible outer digit pairs and 19 possible inner digit pairs, and inner pairs to be checked. If is odd, then the center digit cannot be paired, and leads to 10 times as many possible seeds. The following table demonstrates how many integers can be saved:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number of digits (n) | Total Integers | Possible Seeds: PS(n) | Ratio (%) | Every Nth to check |
| 3 | 900 | 180 | 20 | 5 |
| 4 | 9,000 | 342 | 3.8 | 26 |
| 5 | 90,000 | 3,420 | 3.8 | 26 |
| 6 | 900,000 | 6,498 | .722 | 138 |
| 7 | 9,000,000 | 64,980 | .722 | 138 |
| 8 | 90,000,000 | 123,462 | .136 | 728 |
| 9 | 900,000,000 | 1,234,620 | .136 | 728 |

In total, we checked 1,433,502 integers from 100 to 999,999,999, or about 1 in every 700. There were 880,327 convergent threads and 553,174 candidate Lychrel threads.

Results for this implementation of the reverse add algorithm are found in Figure 6 and Figure 7. Figure 6 is the relative frequency histogram for number of iterations and is similar in shape to the Monte Carlo simulation of 10-digit integers. Figure 7 plots each seed and its iterations to reach a palindrome. The maximum number of iterations this time was 98.

Proving a Lychrel Number:

While it has not yet been proven whether Lychrel numbers exist in base 10, they have been proven in several other bases. For example, in every base that is a power of 2, there is a “family” of Lychrel numbers that all have the same form. Additionally, there are other proven Lychrel numbers in base 11, 17, and 26. The first proven Lychrel number in base 2 is 10110. The way this is proven is by showing that after some point, every subsequent iteration can be formulized. And since the formula aren’t palindromic, and the iterations leading up the that point weren’t, the number is proven to be Lychrel.

Proof: 10110100 in base 2 is a Lychrel number

10110100 can be expressed as 10[n\*1]01[n\*0], where n begins with 2, and represents the number of consecutive 1’s or 0’s. Following the reverse add algorithm ,we have:

Iteration 0: 10110100, expressible as 10[n\*1]01[n\*0]

Iteration 1: 11100001, expressible as 11[(n-2)\*0]1000[(n-2)\*1]01.

Iteration 2: 101101000, expressible as 10[n\*1]01[(n+1)\*0].

Iteration 3: 110010101, expressible as 11[n\*0]10[(n-1)\*1]01.

Iteration 4: 1011101000, expressible as 10[(n+1)\*1]01[(n+1)\*0]

Iteration 4 is identical to iteration 0, except n is incremented by one. The same occurs for iterations 5 and 1, 6 and 2, etc. By induction, the entire thread consists of consists of repetitions of this cycle, and none of the elements are palindromes, so 10110100 is a Lychrel number.

While this proof is for 10110100, 10110100 is the 4th iteration of 10110, and since none of the iterations in between are palindromic, and we’ve proven none afterwards are, 10110 must also a Lychrel number.

Conjecture: Family of Lychrel numbers in bases of power of 2

In any base , then the integer:

where represents n consecutive digits x, is a Lychrel number with period

While each power of 2 can be proven to have a Lychrel of the above form, I hesitate to call it a theorem since there is no way to prove it using the general formula, since the number of periods it takes to complete a cycle is not held constant.

In addition to this family in bases that are a power of 2, there are additional Lychrel numbers that are sporadic. For example, in base 4 1033202000232[n\*2]2302333113230 cycles after 6 iterations, where n increases by 3 each time. Other examples include:

Base 11: 1246277[n\*A]A170352495681825A5026571A506181864A5143171[n\*0]0872542

After 6 iterations, same but with n increased by 1

Base 17: 10023AB83E3B983CFGEC556G4G010[n\*0]0FGCG10FG505GF020CGF[n\*G]GG11G4F655D DGGB299B3D38BB320G

After 6 iterations, same with n increased by 1

Base 26: 1N5ELA6C[n\*P]P6E7[n\*0]0D59ME5N

After 4 iterations, same with n increased by 1.

While no cycles have been proven to exist in base 10, there are examples self-similar sequences, which appear to cycle, only to unravel after a certain number of iterations. For example, 17,509,097,067 has the same first and last 3 digits following a period 8 cycle, but on the 96th iteration, the first 3 digits are 175 and the last 3 are 957, thus breaking what appeared to be a cycle. Another clue that this isn’t a cycle is that the interior digits do not follow any generalized formula, unlike in above examples. And while some candidate Lychrel numbers are self-similar, not all of them are. For example, 196 does not appear to follow any pseudo-cycle.

While unable to come to any definitive conclusions about the status of 196, we have explored some of the properties of candidate Lychrel numbers in base 10 and compared their frequency to palindromic seeds for a given number of digits. We expanded out initial search up to one million by filtering to only test seeds, allowing us to check all possible seeds up to one billion. We also showed how to prove if an integer is Lychrel by testing if its iterations follow a cycle. We also discussed why there appears to be cycles in base 10, but only for the cycle to unravel after a certain number of iterations. While 196 has been iterated using the reverse algorithm a billion times and never summed to a palindromic number, that is only evidence for a strong heuristic argument. At this point, it appears that there are no tools to prove if the underlying properties of 196 means it is a Lychrel number, and unfortunately there does not appear to be any sign that will change anytime soon.

References:

[1] Jason Doucette. World Records http://jasondoucette.com/worldrecords.html

[2] Fred Gruenberger, How to handle numbers with thousands of digits, and why one might want to, *Scientific American*, April 1984

[3] David Seal. Sporadic Solutions https://web.archive.org/web/20170314094542/http://www.p196.org/papers1.html

[4] C W. Trigg.: Palindromes by Addition Mathematics Magazine, Vol 40. No. 1, 26-28, (1967)

[5] Wade Van Landingham, World Records https://web.archive.org/web/20170127054430/http://www.p196.org/lychrel\_records.html

Appendix:













