The Mathematics Teacher
(INDIA)

OFFICIAL JOURNAL OF THE ASSOCIATION OF
MATHEMATICS TEACHERS OF INDIA

Volume: 56 Issues: 3 & 4 Year: 2021

Dr. S.Muralidharan
Editor

EDITORIAL

This issue starts with an exploration on self Numbers by Manu Param. These numbers are defined and the pattern in these numbers is investigated. Some pointers for further exploration are also given.

The 62th International Mathematical Olympiad was hosted by St Petersberg. Due to the COVID pandemic, many countries including India participated remotely using a strict protocol of online monitoring. The questions posed on the two days of the competition and the performance of the Indian team are presented.

We next present the texts of some of the talks delivered at the 54th Annual Conference of AMTI at Kanyakumari. The talks include Keynote address by the President of AMTI, Prof R.C. Gupta memorial lecture, Theme talk, Prof A Narasinga Rao memorial lecture and Prof P.L. Bhatnagar memorial lecture.

We also present the gist of Association activities.

It has been an eventful 8 years since 2014, as the Editor of this journal. It was a great learning experience for me. Due to my other commitments, I am stepping down from the editor responsibility. I wish this journal and AMTI all the best in the service of Mathematics enthusiasts.

CONTENTS

Self Numbers

Manu Param

8th Grade, Mentored by
Raising a Mathematician Foundation

Self Number

Let us define the function $sum(x) =$ the sum of the digits of any real number x. Define $f(x) = x + sum(x)$. Starting with a real number x, we apply the function f repeatedly to get an infinite sequence $S(x) = \{x, f(x), f(f(x)), \ldots, \}$. Any number N that is *NOT* in the above list $S(x)$ for any $x, 0 \leq x < N$ is called a self number. For example, $1, 3, 5, 7$ are self numbers whereas 2 is not, since $2 = f(1)$. In this paper, we study various algorithms to find self numbers. We will also compute the efficiency of these algorithms. There appears to be some pattern in these numbers and we will indicate those as well.

Algorithms for computing self numbers

We describe algorithms to compute self numbers in a given range 0 to N, where N is a positive integer:

Set Algorithm This algorithm follows the definition of self number by calculating $f(x)$ for all of the numbers in the range. We put these values in a set and find out the values that are in the range but are not in the set.

Though it calculates $f(x)$ only once per number, this is not as fast as it might seem, since it requires looping through the set again to find the values that are not in the set.

Calc Algorithm This is a specialization of the Set algorithm where it focuses on calculating whether a given number is a self number or not. The determination of whether a specific number N is a self number or not is done as Set Algorithm by taking the range as 0 to $N-1$ and calculating $f(x)$ for this range. The key difference is that we do not create the set nor do a search in the set but do the comparison and determination as we calculate $f(x)$. If any $f(x) = N$ then N is not a self number and if in all $f(x)$ of range 0 to $N-1$ did not result in N, then N is a Self number.

Optimal Calc Algorithm A variation of the Calc Algorithm can improve the computation drastically for large numbers. This improvement is based on the fact that $x + sum(x) < N$ for all $x < 9 \times \text{number of digits in } N$. Hence rather than calculating $f(x)$ from 0 to $N-1$ as in Calc algorithm, we can calculate from $N - 9 \times \text{number of digits in } N$. This will result in a reduced number of calculations. For example, for $N = 1000$, the number of calculations will reduce from 1000 to 36.

The Optimal Calc algorithm helps to generate Self numbers for higher order numbers without having to start from 1. This also helps to evaluate the patterns at self numbers beyond 10^9.

Running time is the time it takes for an algorithm to run. It is in terms of the input size n.

1. The set algorithm that has an elimination function is $O(n^2)$ because we have to loop through the set n times for each of the n elements.

2. The Calc algorithm, is $O(n^2 \log n)$ because it takes $\log n$ for adding digits and n squared for looping through the numbers twice. So, this explains why it is slower than the other method.

3. The optimial calc algorithm takes $\log n$ instead of n for each number so the running time is $O(n \log\log n)$.

The following table compares the efficiency of these three algorithms (all times in seconds):

Range	Set Alg	Calc Alg	Opt Calc Alg
1	0	0	0
1 – 1000	0.11293	1.56530	0.03999
1 – 10000	12.04911	204.30051	0.67079
1 – 100000	1136.80346	–	10.49942
Is N a self number?	10^{12} calcs	10^{12} calcs	108 calcs
Running Time	$O(n^2)$	$O(n^2 \log n)$	$O(n \log\log n)$

Patterns in self numbers

Identifying a pattern to incrementally generate the self number will give an efficient algorithm. The pattern for self numbers changes depending in the range of the numbers. The pattern for Self numbers has an underlying formula of $10^k + 1$. We have split the pattern around these boundary points. There does not appear to be a uniform pattern for all the numbers and the pattern changes depending on the number of digits K in the number.

1. $K = 0$: The least self number is 1 and subsequent numbers can be obtained by adding $10^k + 1 = 2$ to this

number. Thus the self numbers in this range are the odd numbers.

2. $K = 1:$ Pattern continues with the addition of 11, 9 times from 9, until we get to 108

3. $K = 2:$ A 2 is added to get the next self number 110. This is followed by the $K = 1$ pattern of 11's, 9 times. Combining the two we get $2 + 9 \cdot 11 = 101$ our base number.

   This pattern of 101 repeats 8 times till we get to 916.

4. $K = 3:$ The pattern has following steps in this part

   (a) A 2 is added

   (b) Then the pattern continues with increments of 11's for 8 times

   (c) Then 15 is added

   (d) Followed by increments of 11s, 8 times

   (e) The above four steps are followed by the $K = 2$ pattern listed in $K = 2$, 8 times.

   These 5 steps above gives the base number

   $$(2 + 8 \cdot 11) + (15 + 8 \cdot 11) + 8 \cdot (2 + 9 \cdot 11) = 1001$$

   This overall pattern of 1001 repeats 9 times till the next transition point.

5. $K = 4:$ The pattern has the similar steps as in $K = 3$ pattern plus an additional step

   (a) A 2 is added

(b) Then the pattern continues with increments of 11's for 7 times

(c) Then 28 is added

(d) Followed by increments of 11s, 7 times

(e) This is followed by the 101 pattern listed in $K = 2$, 8 times

(f) And finally, the 1001 pattern listed in $K = 3$, 9 times

These 6 steps gives us the next base number

$(2+7\cdot 11)+(28+7\cdot 11)+8\cdot(2+9\cdot 11)+9\cdot(1001) = 10001$

This overall pattern of 10001 repeats 9 times to reach the next transition.

6. $K = 5$: The pattern has the similar steps as in $K = 3$ and $K = 4$ pattern plus an additional step.

   (a) A 2 is added

   (b) Then the pattern continues with increments of 11's for 6 times

   (c) Then 41 is added

   (d) Followed by increments of 11s, 6 times

   (e) This is followed by the 101 pattern listed in K=2, 8 times

   (f) Then the 1001 pattern listed in $K = 3$, 9 times

   (g) And finally, the 10001 pattern repeats 9 times

These 7 steps gives us the next base number

$$(2 + 6 \cdot 11) + (41 + 6 \cdot 11) + 8 \cdot (2 + 9 \cdot 11) \\ + 9 \cdot (1001) + 9 \cdot 10001 = 100001$$

This overall pattern of 100001 repeats 9 times to reach the next transition.

Generalizing the $K = 3, 4, 5$ pattern, following steps repeat for $K = 3, 4, 5, \ldots 10$:

1. First a 2 is added
2. Then the pattern continues with increments of 11's for $9 - (K - 2)$ times
3. Then $2 + 13(K - 2)$ is added
4. Followed by increments of 11s for $9 - (K - 2)$ times
5. This is followed by the $K = 2$ steps 8 times
6. And every pattern before from $K = 3$ to $K - 1$
7. And all the above steps are repeated 9 times.

Conclusion

On Evaluating the numbers including the higher order numbers, there seems to be a repeating scheme but it has some anomalies that makes it difficult to determine a global pattern, thus far. The open questions that are worth looking into are:

1. Evaluating patterns for numbers beyond 10^12 . It looks like the $2+10 \cdot 13 = 132$ is split into 12 11's and hence 11's

are just added. These higher orders have to be generated and analyzed.

2. Find a global pattern.