Actuarial Review May-June 2021
It's a Puzzlement Solutions – Identifiable Sequences

John Bergland

If we define "slowest growing" as not leaving out any possible sum, there are many "slowest growing" sequences. They told us to use strictly increasing positive integers, but the math is easier to explain if we use non-negative integers. We will be able to add a constant to each term of the sequences to make them be positive.

The simplest sequences to describe are these:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A(n)	0	1	4	5	16	17	20	21	64	65	68	69	80	81	84	85	256	257
B(n)	0	2	8	10	32	34	40	42	128	130	136	138	160	162	168	170	512	514

Here B(n) is twice A(n) for all n. To see the pattern in A(n), look at the first 2, 4, 8, 16... numbers in the sequence. Take the first 2^q numbers, add 2^2^q to them to get the next 2^q numbers. For example, the first 2^2 = four numbers are 0, 1, 4, 5. We add 2^2^2 = 16 give us the next four numbers: 16, 17, 20, 21. An easier way to see this is to write the numbers in binary, and look at n-1 instead of n.

n-1	A(n)	B(n)	n-1 in binary	A(n) in binary	B(n) in binary
0	0	0	0	0	0
1	1	2	1	1	10
2	4	8	10	100	1000
3	5	10	11	101	1010
4	16	32	100	1 0000	10 0000
5	17	34	101	1 0001	10 0010
6	20	40	110	1 0100	10 1000
7	21	42	111	1 0101	10 1010
8	64	128	1000	100 0000	1000 0000
9	65	130	1001	100 0001	1000 0010
10	68	136	1010	100 0100	1000 1000
11	69	138	1011	100 0101	1000 1010
12	80	160	1100	101 0000	1010 0000
13	81	162	1101	101 0001	1010 0010

14	84	168	1110	101 0100	1010 1000
15	85	170	1111	101 0101	1010 1010
16	256	512	1 0000	1 0000 0000	10 0000 0000
17	257	514	1 0001	1 0000 0001	10 0000 0010
18	260	520	1 0010	1 0000 0100	10 0000 1000

Notice how to get A(n) from n-1 we just put a zero between each binary digit. For example, 1011 becomes 1**0**0**0**1**0**1. Also note how A(n) has only odd powers of two. B(n) has only even powers of two. Any non-negative integer can be broken apart into odd and even powers of two in only one way. Thus, any such number can be expressed as the sum of A(n) and B(n) in exactly one way. This shows that A(n) and B(n) are identifiable sequences and that they are slowest growing.

How to compute efficiently?
I found the easiest way for these was to use the programming calculator on my computer. It translates between binary and decimal instantly. So to find A(1000), I would put 1000-1 = 999 in as a decimal and get 1111100111 as binary. I would manually add zeros between the binary digits to get 1010101010000010101. When I type this into my calculator as binary it spits out 349205 as decimal.

Since they wanted us to begin with A(n) as 1 and B(n) as 2, we can add 1 and 2 to each sequence. (Or vice versa if you like.) This would end up giving us A(100) = 5126, A(200) = 20502, and A(1000) = 349206.

The growth rates are polynomial – in fact, very close to n^2.

But there are other sequences that are also slowest growing. Here's one that has A'(m) grow as slow as possible for the first three terms.

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A'(m)	0	1	2	9	10	11	36	37	38	45	46	47	144	145	146	153	154	155
B'(m)	0	3	6	18	21	24	72	75	78	90	93	96	288	291	294	306	309	312

This example is grouped by threes. We can see that the first 3 numbers of A'(m) and B'(m) can add up to any sum from 0 to 8. The relationship between these sequences and our original sequences is that every third number of A'(m) is 9 times A(n) for the appropriate m and n. So, the original sequence A(n) began 0,1,4,5,16 which will translate to 0,9,36,45,144 as every third number in A'(m). If you are really into equations, you could write

A'(m) = 9*A(floor[(m-1)/3]+1)+[(m-1) mod(3)]
B'(m) = 9*B(floor[(m-1)/3]+1)+3*[(m-1) mod(3)]

There is nothing special about three. We could have picked any positive integer, k and defined:
A”(m) = k^2*A(floor[(m-1)/3]+1)+[(m-1) mod(k)]
B”(m) = k^2*B(floor[(m-1)/3]+1)+k*[(m-1) mod(k)]

In this case, we could take k = 1000, and it's very easy to find A”(1000)... it's just 1000-1=999.

Fun problem.

We could extend the problem:
Find three sequences A(n), B(n) and C(n) that you give to three people. They each pick a number and tell you the sum. You tell each of them which number they picked. This might be called a set of triple identifiable sequences. Find a slowest growing example.

For an answer, we could do the same trick – just having the binary numbers of these forms:
A(n) is ...00x00x00x00x00x
B(n) is ...0x00x00x00x00x0
C(n) is ...x00x00x00x00x00
where the x's are either 0 or one, independently.

Ken Klinger

I believe this problem is related to the Mian-Chowla sequence as described in https://oeis.org/A005282 whose values are listed as a(n).

A(n) can be the odd numbered elements of the sequence and B(n) the even. That is A(n) = a(2n-1) and B(n)= a(2n).

Here A(100) = 170984, A(200)=1136410, and A(1000)=96579998.

Eamonn Long

Scheme

To compose A[n], take the binary representation of n and consider this instead a base 4 number.

For example, if n == 11, the binary representation is 1011. The base 4 number 1011 in base 10 is 69.
So A[11] = 69. A[100] = 5136. A[200] = 20544. A[1000] = 349248.

N.B.: There are only 1s and 0s in the base 4 representation of A[n].

Choose B[n] = 2* A[n] so that there are only 2s and 0s in the base 4 representation of B[n].

<u>Uniqueness of sums</u>

In this scheme, A[i] + B[j] is unique and this can be seen in base 4.

In the base 4 representation of A[i] + B[j], look at the kth digit of A[i] + B[j].

- If the kth digit is 0, this means that the kth digit in base 4 of A[i] = kth digit in base 4 of B[j] = 0
- If the kth digit is 1, this means that the kth digit in base 4 of A[i] = 1, while the kth digit in base 4 of B[j] = 0
- If the kth digit is 2, this means that the kth digit in base 4 of A[i] = 0, while the kth digit in base 4 of B[j] = 2
- If the kth digit is 3, this means that the kth digit in base 4 of A[i] = 1, while the kth digit in base 4 of B[j] = 2
- There are no other possibilities for the kth digit in base 4
- This decomposition works for all digits in base 4 of A[i] + B[j], hence there is only one possible decomposition

<u>Slowest possible</u>

Where n = 2^m – 1

A[n] + B[n] = 4^m – 1 or is approximately n^2

However, A[n] + B[n] will grow at least as fast as n^2 since there are in general n^2 numbers in the array [A[i] + B[j] for I,j <= n] and these n^2 numbers will all be different.

Hence, asymptotically this is the slowest growing sequence possible.

<u>Tomasz Serbinowski</u>

Answer:
A(100) = 41829
A(200) = 292394
A(500) = 3609240

A(1000) = ?

Growth rate: n^2.77
based on values A(200) through A(500)

Solution:
I'm not sure there is the slowest growing pair of identifiable sequences. I assumed that the idea was to start with A(1) = 1 and B(1) = 2 and construct each sequence so that each element was the smallest possible.

Essentially, given the A(1) through A(n) and B(1) through B(n) we try to find the smallest number k so that A(n+1) = A(n) + k works.

We form all n^2 sums A(i) + B(j) where i, j = 1...n.

For a given k, we look at all sums
A(n) + k + B(j)
j = 1...n

If none of these n sums is already on our list of our n^2 sums A(i) + B(j) we have our next member of the sequence.

I wrote a short program in R (IdentifiableSequences.R), but it is too slow to get A(1000). It took about 12 sec for A(100), about 5 min for A(200), 40 min for A(300) and almost 10 hours for A(500).

I stopped the program at A(508). The first 508 members of each sequence are in the attached Excel file (IdentSeq500).

IdentSeq500.xlsx   IdentifiableSequences.R