# EASILY INFERRED SEQUENCES

by

Dana Angluin

Memorandum No. ERL-M499

31 December 1974

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

# Easily Inferred Sequences

by Dana Angluin

WARNING: THIS PAPER IS DEVOID OF THEOREMS

## intent

This is an attempt to describe what sequences of numbers have an obvious, or easily inferred, pattern. The test of such a sequence is that from a few successive terms of it a person very easily guesses a correct rule of generation for it. Characterizing these sequences was chosen as a (hopefully small) step towards understanding the process of learning-from-examples; this is an informal first approximation to that characterization.

## definitions and nondefinitions

The sequences we shall consider will be infinite sequences of non-negative integers (or, at least, strings of decimal digits) which are "computable," that is, for each sequence it must be possible to write a computer program which will print out the successive terms of the sequence starting with the first one and continuing *ad infinitum*. A person will be said to have "inferred" such a sequence when he produces a computer program (or equivalent rule, making liberal use of Church's Thesis) which will so generate all the terms of the sequence.

No definitions will be offered of "obvious", "easy", "few", or "person", so the class of sequences under consideration will remain quite ill-defined. (Note that if we set a bound (e.g., 40 digits, commas, and blanks) on the size of the finite presentation of terms, we get a bound ($12^{40}$) on the

---

Research sponsored by National Science Foundation Grant GJ-35604X1.

number of sequences in the class, though not a particularly useful one.)

## method

The method of description is the enumeration of some tricks for constructing easily inferred sequences, together with some example sequences for each trick. I have generally tended to start each group of examples with the most straightforward applications of the trick and to progress to more difficult variants to suggest its limits and its interactions with other features. I have also provided a few cautions and counter examples where it seemed that an unrestricted variation of a construction led to uninferrability.

Thus, in general plan and degree of analyticity, this paper lies somewhere between "THE JOY OF COOKING" and an outline review of German grammar.

## plea

The current list of tricks is unsatisfactorily organized and probably woefully incomplete. I would greatly appreciate constructive suggestions on either count. Another paper will sketch algorithmic methods for inferring these sequences; I would welcome comments on that topic as well.

## a partial glossary of notations

| | |
|---|---|
| x+y | addition |
| x-y | subtraction |
| x*y | multiplication |
| x/y | division |
| x↑y | exponentiation ("x to the power y") |
| explode(x) | a list of the digits of x |
| reverse(x) | the digits of x in reverse order |
| length(x) | the number of digits of x |
| firstdigit(x) | the leading digit of x |
| lastdigit(x) | the trailing digit of x |
| tail(x) | the digits of x with the first one removed |
| sort(x) | the digits of x in increasing order of value |
| conc(x,y) | the digits of x concatenated with those of y |
| last(g) | the last term of the group g |
| sum(g) | the sum of the terms of g |
| conc(g) | the concatenation of the terms of g |

examples:

explode(1024) = 1, 0, 2, 4

reverse(1024) = 4201

length(1024) = 4

firstdigit(1024) = 1

lastdigit(1024) = 4

tail(1024) = 024

sort(15129) = 11259

conc(64, 128) = 64128

last((4, 16, 256)) = 256

sum((4, 16, 256)) = 276

conc((4, 16, 256)) = 416256

CONTENTS

CONTENTS, continued

1 Simple arithmetic sequences

<u>constant</u>:

1, 1, 1, 1, 1, ...

47, 47, 47, 47, 47, ...

<u>constant almost everywhere</u>:

4, 3, 2, 1, 0, 0, 0, 0, 0, 0, ...

13, 2, 2, 2, 2, 2, 2, ...

$x_{i+1} = f(x_i)$, where $f(x) = x+c$, $x*c$, $x\uparrow c$, $x*c+d$:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, ...

2, 4, 6, 8, 10, 12, 14, 16, ...

2, 4, 8, 16, 32, 64, 128, 256, ...

7, 10, 13, 16, 19, 22, 25, ...

2, 6, 18, 54, 162, 486, ...

2, 4, 16, 256, 65536, ...

2, 9, 23, 51, 107, 219, ...

<u>squares and other quadratic</u>

1, 4, 9, 16, 25, 36, 49, 64, ...

1, 4, 10, 19, 31, 46, 64, ...

2, 5, 10, 17, 26, 37, 50, 65, ...

1, 2, 4, 7, 11, 16, 22, ...

<u>cubes</u>

1, 8, 27, 64, 125, 216, 343, ...

(Other cubic, and higher degree polynomial, sequences are generally not quickly inferred, though determined differencing eventually yields a constant sequence.)

## 2 Intertwining

### 2.1 two "independent" sequences intertwined $x_i, y_i$.

<u>one or both constant:</u>

1, 3, 1, 3, 1, 3, 1, 3, 1, ...

1, 1, 1, 2, 1, 3, 1, 4, 1, 5, ...

0, 1, 0, 1, 0, 2, 0, 1, 0, 2, 0, 3, 0, 1, 0, 2, 0, 3, 0, 4, ...

3, 4, 1, 2, 1, 2, 1, 2, 1, ...

(constituents constant almost everywhere)

<u>signs of difference-sequence alternating:</u>

0, 16, 1, 17, 2, 18, 3, 19, 4, 20, 5, 21, ...

10, 6, 100, 12, 1000, 24, 10000, 48, 100000, 96, ...

9, 5, 16, 6, 25, 7, 36, 8, 49, 9, 64, ...

<u>other:</u>

11, 12, 111, 123, 1111, 1234, 11111, 12345, ...

2, 3, 4, 9, 8, 27, 16, 81, 32, 243, ...

1, 1, 2, 1, 2, 2, 3, 1, 3, 2, 3, 3, 4, 1, 4, 2, ...

2, 4, 4, 9, 8, 16, 16, 25, 32, 36, 64, ...

4, 1, 9, 8, 16, 27, 25, 64, 36, 125, 49, ...

<u>period (number of sequences intertwined) > 2:</u>

8, 4, 7, 8, 4, 7, 8, 4, 7, 8, 4, ...

1, 2, 3, 11, 22, 33, 111, 222, 333, 1111, 2222, ...

2, 4, 4, 4, 9, 8, 6, 16, 16, 8, 25, 32, 10, 36, 64, 12, ...

(period not evident near the start)

2.2 two "dependent" sequences intertwined $x_i, f(x_i)$

arithmetic f, f(x) = x+c, x*c, x↑c, c↑x, x*c+d:

4, 3, 9, 8, 16, 15, 25, 24, 36, 35, 49, 48, ...

2, 4, 5, 25, 8, 64, 11, 121, 14, 196, 17, ...

2, 4, 3, 8, 4, 16, 5, 32, 6, 64, 7, 128, ...

36, 73, 48, 97, 510, 1021, 612, 1225, ...

other f, f(x) = explode(x), reverse(x), firstdigit(x), lastdigit(x), length(x), conc(x), sort(x):

32, 3, 2, 64, 6, 4, 128, 1, 2, 8, 256, 2, 5, 6, 512, ...

9, 9, 16, 61, 25, 52, 36, 63, 49, 94, 64, 46, ...

25, 5, 49, 9, 81, 1, 121, 1, 169, 9, 225, 5, ...

8, 8, 16, 1, 32, 3, 64, 6, 128, 1, 256, 2, 512, 5, ...

1, 1, 8, 1, 27, 2, 64, 2, 125, 3, 216, 3, 343, 3, 512, 3, ...

2, 0, 4, 0, 16, 00, 256, 000, 65536, 00000, ...

6, 69, 12, 1236, 24, 2481, 48, 48144, 96, ...

9, 9, 916, 169, 91625, 12569, 9162536, 1235669, ...

concatenation, reversal variants:

36, 48, 510, 612, 714, 816, ...

99, 1661, 2552, 3663, 4994, 6446, ...

99, 166, 255, 366, 499, 644, 811, ...

11, 9, 27, 25, 51, 49, 83, 81, 123, 121, ...

2.3 intertwining three sequences, with binary operator, $x_i, y_i, f(x_i, y_i)$ or $x_i, x_{i+1}, f(x_i, x_{i+1})$

f(x,y) = x+y, x*y, conc(x,y):

2, 4, 6, 3, 8, 11, 4, 16, 20, 5, 32, 37, 6, ...

2, 3, 6, 4, 5, 20, 6, 7, 42, 8, 9, 72, 10, ...

6, 12, 612, 24, 48, 2448, 96, 192, 96192, ...

all three concatenated:

347, 5611, 7815, 91019, 111223, ...

## 3 Sequences of groups

A "group" will be a finite ordered set of elements which are either numbers or groups. Sequences of groups, for example,

(2), (2, 2), (2, 2, 2), (2, 2, 2, 2), ...

(1), (1, 2), (1, 2, 3), (1, 2, 3, 4), ...

((1)), ((1, 1), (2, 2)), ((1, 1, 1), (2, 2, 2), (3, 3, 3)), ...

will be employed in the description of other sequences, but no parentheses may appear in the finished product. If we were simply to drop the parentheses from these three sequences, we could expect someone to guess where to replace them in the latter two examples, but not in the first. If, however, we were to intertwine the first sequence with the constant zero sequence and then drop parentheses, we would get

2, 0, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 2, 0, ...

which reflects the original grouping quite clearly. Techniques will be given for construction and for manipulation of sequences of groups.

### 3.1 constructing sequences of groups

#### 3.1.1 breaking a single sequence into groups

the sequence of lengths-of-groups should be easily inferred:

(0), (0, 0), (0, 0, 0), (0, 0, 0, 0), ...

(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12), ...

(1), (1, 1), (1, 1, 1, 1), (1, 1, 1, 1, 1, 1, 1, 1), ...

(2, 4), (8, 16, 32), (64, 128, 256, 512), ...

As indicated above, these sequences are generally further manipulated (intertwined with other sequences or the groups concatenated) before the parentheses are dropped.

3.1.2 the "one-loop" sequences

3.1.2.1 construction

Consider the variations of the simple loop:

```
         k ← 1
loop:    FOR x = 1 TO k [reversed or not]
              PRINT [x or k]
         ENDFOR
         k ← k + 1
         GOTO loop
```

which produce the sequences of groups:

(1), (2, 2), (3, 3, 3), (4, 4, 4, 4), ...

(1), (1, 2), (1, 2, 3), (1, 2, 3, 4), ...

(1), (2, 1), (3, 2, 1), (4, 3, 2, 1), ...

which in turn may be concatenated or not and stripped of parentheses to produce:

1, 2, 2, 3, 3, 3, 4, 4, 4, 4, ...

1, 22, 333, 4444, ...

1, 1, 2, 1, 2, 3, 1, 2, 3, 4, ...

1, 12, 123, 1234, ...

1, 2, 1, 3, 2, 1, 4, 3, 2, 1, ...

1, 21, 321, 4321, ...

all eminently inferrable. (The last is the form of "The Twelve Days of Christmas" for example.)

3.1.2.2 using the "one-loop" sequences as patterns

These basic sequences may be used to "pattern" other, relatively arbitrary, sequences. For example, if $x_1, x_2, x_3, x_4, x_5, \ldots$ is strictly increasing, then it may fairly easily be recovered from the pattern $x_1, x_2, x_1, x_3, x_2, x_1, x_4, x_3, x_2, x_1, x_5, \ldots$

examples:

2, 44, 888, 16161616, 3232323232, ...

0, 0, 2, 0, 2, 4, 0, 2, 4, 6, 0, 2, 4, 6, 8, ...

4, 45, 459, 45914, 4591423, ...

642, 8642, 108642, 12108642, ...

1, 11, 11, 111, 111, 111, 1111, 1111, 1111, 1111, 11111, ...

3, 8, 3, 15, 8, 3, 24, 15, 8, 3, 35, 24, 15, 8, 3, ...

3, 4, 5, 6, 5, 6, 7, 8, 7, 8, 7, 8, 9, 10, 9, 10, 9, 10, 9, 10, ...

(This last is 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, ... patterning of (3, 4), (5, 6), (7, 8), (9, 10), ... but smudged in effects.)

3.1.3 the "two-loop" sequences

In an analogous way, we may generate all variants of a simple pair of nested loops:

```
        k ← 1
loop:   FOR x = 1 to k [reversed or not]
            FOR y = 1 TO [k or x]
                PRINT [k or x or y]
            ENDFOR
        ENDFOR
        k ← k + 1
        GOTO loop
```

which produces the following collection of sequences (the variant with concatenation of the outer loop grouping has been systematically omitted):

1, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, ...

1, 22, 22, 333, 333, 333, ...

1, 1, 1, 2, 2, 1, 1, 1, 2, 2, 2, 3, 3, 3, 1, 1, 1, 1, ...

1, 11, 22, 111, 222, 333, 1111, ...

1, 1, 2, 1, 2, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 4, ...

1, 12, 12, 123, 123, 123, 1234, ...

1, 2, 2, 2, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, ...

1, 2, 22, 3, 33, 333, 4, 44, 444, 4444, 5, ...

1, 22, 2, 333, 33, 3, 4444, 444, 44, 4, 55555, ...

1, 1, 2, 2, 1, 2, 2, 3, 3, 3, 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, ...

1, 1, 22, 1, 22, 333, 1, 22, 333, 4444, ...

1, 1, 1, 2, 1, 1, 2, 1, 2, 3, 1, 1, 2, 1, 2, 3, 1, 2, 3, 4, ...

1, 1, 12, 1, 12, 123, 1, 12, 123, 1234, ...

1, 2, 1, 2, 1, 3, 2, 1, 3, 2, 1, 3, 2, 1, 4, 3, 2, 1, ...

1, 21, 21, 321, 321, 321, 4321, ...

1, 1, 2, 1, 1, 2, 1, 3, 2, 1, 1, 2, 1, 3, 2, 1, 4, 3, 2, 1, ...

1, 1, 21, 1, 21, 321, 1, 21, 321, 4321, ...

1, 2, 2, 1, 1, 3, 3, 3, 2, 2, 2, 1, 1, 1, 4, 4, 4, 4, ...

1, 22, 11, 333, 222, 111, 4444, 3333, ...

1, 2, 2, 1, 3, 3, 3, 2, 2, 1, 4, 4, 4, 4, 3, 3, 3, ...

1, 22, 1, 333, 22, 1, 4444, 333, ...

1, 1, 2, 1, 1, 2, 3, 1, 2, 1, 1, 2, 3, 4, 1, 2, 3, ...

1, 12, 1, 123, 12, 1, 1234, 123, 12, 1, ...

1, 2, 1, 1, 3, 2, 1, 2, 1, 1, 4, 3, 2, 1, 3, 2, 1, ...

1, 21, 1, 321, 21, 1, 4321, 321, 21, 1, ...

These are all relatively easy to infer, and, like the "one-loop" sequences, may be used to pattern other sequences, for example:

3, 388, 388151515, 38815151524242424, ...

("Three (and more)-loop" sequences seem to be of diminished return.)

3.1.4 sequences of constant, linear, geometric groups

One generalization of the "one-loop" and "two-loop" sequences is the notion of a sequence of groups, internally in constant, linear, or geometric progression, in which the derived sequences of (say) first elements, lengths, and differences (or ratios) are easily inferred.

3.1.4.1 "independent" sequence of first elements program - form:

```
        t ← 1
loop:   x ← f(t)
        ℓ ← g(t)
        c ← h(t)
        FOR i = 1 TO ℓ
            PRINT x
            x ← x+c (or x*c)
        ENDFOR
        t ← t+1
        GOTO loop
```

where f, g, h are easily inferred sequences.

examples:

1, 1, 2, 3, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 7, 1, ...

16, 17, 18, 32, 33, 34, 64, 65, 66, 128, 129, 130, ...

1, 4, 8, 9, 18, 36, 16, 32, 64, 128, 25, 50, 100, ...

1, 2, 4, 3, 9, 27, 4, 16, 64, 256, 5, 25, 125, ...

1, 2, 2, 4, 4, 4, 4, 8, 8, 8, 8, 8, 8, 8, 8, 16, ...

1, 2, 4, 3, 6, 9, 4, 8, 12, 16, 5, 10, 15, ...

concatenation and reversal variants:

456, 91011, 161718, 252627, 363738, ...

6, 2, 27, 9, 3, 108, 36, 12, 4, 405, 135, 45, ...

(some property must mark groups, for example)

2, 3, 4, 5, 6, 7, 7, 8, 9, 10, 11, 12, 13, 14, 15, 13, 14, 15, 16, 17, 18, 17, ... is <u>not</u> readily grouped to

(2), (3, 4), (5, 6, 7), (7, 8, 9, 10), (11, 12, 13, 14, 15), (13, 14, 15, 16, 17, 18), ...

3.1.4.2 first element derived from preceding last element

<u>program - form</u>:

```
        t ← 1
        x ← x_o
loop:   ℓ ← g(t)
        c ← h(t)
        FOR i = 1 TO ℓ
            PRINT x
            x ← x+c (or x*c)
        ENDFOR
        x ← f(x) (or f(x-c) or f(x/c))
        t ← t+1
        GOTO loop
```

where g, h are easily-inferred sequences and f is a simple function, e.g., f(x) = x+d or x*d.

<u>examples</u>:

2, 4, 5, 10, 20, 21, 42, 84, 168, 169, 338, ...

1, 2, 4, 5, 6, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, ...

(note difference-sequence: 1, 2, 1, 1, 2, 1, 1, 1, 2, ...)

1, 2, 3, 3, 4, 5, 6, 6, 7, 8, 9, 10, 10, 11, 12, 13, ...

1, 2, 4, 5, 6, 12, 13, 14, 15, 30, 31, 32, 33, 34, 68, ...

<u>concatenation and reversal variants</u>:

12, 357, 8111417, 1822263034, ...

3, 11, 9, 37, 35, 33, 117, 115, 113, 111, 359, 357, ...

3.1.4.3 alternating simple rules

As a special case of deriving first from last when each group is of length 1, we get a rule

$$x_{i+1} = \begin{cases} f(x_i) & \text{if } i \text{ is odd} \\ g(x_i) & \text{if } i \text{ is even} \end{cases}$$

and in this case we may have f(x), g(x) = x+c, x*c, x↑c, x*c+d, reverse(x).

<u>examples</u>:

1, 2, 5, 10, 21, 42, 85, 170, ...

1, 3, 9, 11, 121, 123, 15129, 15131, ...

91, 93, 39, 41, 14, 16, 61, 63, 36, 38, ...

3.2 operations with sequences of groups

3.2.1 intertwining with a sequence of numbers

<u>$g_i$, $y_i$ with g a grouping of constant sequence</u>:

0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, ...

0, 0, 0, 4, 0, 0, 0, 9, 0, 0, 0, 16, 0, 0, 0, 25, 0, 0, ...

1, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1, 1, 1, 4, 1, 1, 1, 1, 1, 5, 1, ...

<u>$g_i$, $y_i$ with y a constant sequence</u>:

1, 0, 2, 3, 0, 4, 5, 6, 0, 7, 8, 9, 10, 0, 11, 12, ...

1, 0, 1, 2, 0, 1, 2, 3, 0, 1, 2, 3, 4, 0, 1, 2, ...

0, 1, 2, 0, 1, 0, 1, 2, 0, 1, 0, 1, 0, 1, 2, 0, 1, ...

1, 2, 4, 1, 2, 3, 4, 1, 2, 3, 4, 4, 1, 2, 3, 4, 5, 4, ...

<u>$g_i$, $f(g_i)$, f(g) = last(g), last(g) + 1, reverse(last(g)), sum(g), conc(g):</u>

1, 2, 3, 3, 4, 5, 6, 7, 7, 8, 9, 10, 11, 12, 12, 13, ...

4, 5, 9, 16, 17, 25, 36, 49, 50, 64, 81, 100, 121, 122, ...

1, 2, 3, 3, 1, 2, 3, 4, 4, 1, 2, 3, 4, 5, 5, 1, 2, 3, ...

16, 61, 32, 64, 46, 128, 256, 512, 215, 1024, ...

2, 4, 24, 8, 16, 32, 81632, 64, 128, 256, 512, 64128256512, ...

2, 3, 5, 4, 5, 6, 15, 7, 8, 9, 10, 34, 11, 12, 13, ...

(This last generalizes $x_i$, $x_{i+1}$, $x_i + x_{i+1}$ of Section 2.3.)

<u>$g_i$, $y_i$ y "independent":</u>

1, 2, 16, 3, 4, 5, 32, 6, 7, 8, 9, 64, 10, 11, ...

1, 2, 1, 2, 5, 1, 2, 3, 10, 1, 2, 3, 4, 17, 1, 2, ...

10, 11, 22, 100, 111, 222, 333, 1000, 1111, 2222, ...

3.2.2 intertwining with another sequence of groups

<u>$g_i$, $h_i$ one a grouping of a constant sequence:</u>

1, 2, 1, 1, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2, ...

0, 1, 0, 0, 1, 2, 0, 0, 0, 1, 2, 3, 0, 0, 0, 0, 1, ...

1, 2, 1, 1, 3, 4, 1, 1, 1, 5, 6, 1, 1, 1, 1, 7, 8, ...

<u>$g_i$, $h_i$ one identity or reverse of other:</u>

1, 2, 1, 2, 1, 2, 3, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 1, ...

2, 4, 4, 2, 8, 16, 32, 32, 16, 8, 64, 128, 256, 512, ...

2, 4, 2, 8, 16, 32, 16, 8, 64, 128, 256, 512, 256, ...

(one set of end points identified)

1, 2, 1, 2, 3, 2, 1, 2, 3, 4, 3, 2, 1, 2, 3, 4, 5, 4, ...

(both sets of end points identified)

<u>other $g_i$, $h_i$</u>

1, 2, 1, 4, 3, 4, 5, 1, 4, 9, 6, 7, 8, 9, 1, 4, 9, 16, 10, ...

1, 2, 3, 1, 9, 4, 1, 8, 27, 1, 81, 16, 1, 32, 243, 1, ...

36, 38, 47, 49, 64, 66, 79, 81, 100, 102, 119, 121, ...

1, 2, 2, 3, 5, 3, 4, 7, 11, 13, 17, 5, 6, 19, 23, 29, 31, 37, ...

<u>("marked", for example, with 0's)</u>:

1, 2, 0, 4, 9, 0, 3, 4, 5, 0, 16, 25, 36, 0, ...

1, 0, 4, 9, 0, 2, 3, 4, 0, 16, 25, 36, 49, 0, ...

<u>concatenation variants</u>:

511, 10111, 171111, 2611111, 37111111, ...

123, 3, 2, 1, 1234, 4, 3, 2, 1, 12345, 5, ...

33, 388, 381515, 38152424, ...

41234, 512345, 6123456, ...

121, 12321, 1234321, 123454321, ...

247, 4814, 358, 61016, 469, 81218, 5710, ...

## 4 concatenation

### 4.1 constant groups concatenated

Concatenated constant groups are readily inferrable:

1, 11, 111, 1111, 11111, ...

2, 22, 2222, 22222222, ...

4747, 474747, 47474747, ...

1, 22, 333, 4444, ...

and they also concatenate with other things fairly inferrably:

211, 3111, 41111, 511111, 6111111, ...

711, 13111, 251111, 4911111, 97111111, ...

13, 123, 1223, 12223, 122223, ...

2112, 411114, 61111116, 8111111118, ...

2114, 31116, 411118, 51111110, ...

90009, 16000061, 250000052, 3600000063, ...

2113, 411115111116, 7111111181111111191111111110, ...

215154, 31515156, 4151515158, 5151515151510, ...

2200, 333111, 44442222, 5555533333, ...

1, 122, 122333, 1223334444, ...

11211, 1113111, 111141111, 11111511111, ...

(See also section on repeating digits.)

### 4.2 concatenation of pairs

<u>conc($x_i$, $x_{i+1}$) or conc($x_i$, $y_i$) where x and y are constant, linear, geometric sequences or sequences of groups</u>:

12, 34, 56, 78, 910, 1112, 1314, ...

36, 1224, 4896, 192384, ...

1232, 1234, 1238, 12316, 12332, ...

16, 212, 324, 448, 596, 6192, ...

21, 411, 8111, 161111, 3211111, ...

41234, 512345, 6123456, 71234567, ...

<u>conc($x_i$, $y_i$) where $y_i$ = $x_i$, reverse($x_i$), firstdigit($x_i$), lastdigit($x_i$), $x_i$+c:</u>

1212, 123123, 12341234, 1234512345, ...

33, 88, 1515, 2424, 3535, 4848, ...

88, 1661, 3223, 6446, 128821, ...

272, 818, 3433, 7297, 21872, ...

366, 811, 1444, 2255, 3244, 4411, ...

2628, 6365, 124126, 215217, 342344, ...

<u>arbitrary conc($x_i$, $y_i$) is generally not very inferrable:</u>

53, 107, 1715, 2631, 3763, 50127, ...

29, 2865, 126217, 344513, ...

36, 1118, 2738, 5166, 83102, ...

4.3 concatenation of nonconstant groups

<u>internally in linear or geometric progression:</u>

1, 23, 456, 78910, 1112131415, ...

357, 9111315, 1719212325, ...

1, 24, 81632, 64128256512, ...

24, 3612, 481632, 510204080, ...

12151821, 24273033, 36394245, ...

intertwined constant with constant or linear:

343, 56565, 7878787, 9109109109109, ...

01, 0203, 040506, 070809010, ...

2326, 49412415, 618621624627, ...

232527, 39311313, 415417419, ...

arbitrary group concatenation is generally not too inferrable:

71321, 314357, 7391111, 133157183, ...

3748, 59610711, 81291310141115, ...

3478, 5691078, 111291013141112, ...

(This last fares a little better? Perhaps a special effect of conc(x, x+1)?)

4.4 concatenation, other patterns of matching

Already it has been noted that "patterns" like $x_1x_1$, $x_2x_2$, $x_3x_3$, $x_4x_4$, ... or $x_1$, $x_1x_2$, $x_1x_2x_3$, $x_1x_2x_3x_4$, ... permit recovery of the sequence $x_1$, $x_2$, $x_3$, $x_4$, ...

other patterns, within one term, xyx, xxy, xyyx:

838, 16416, 32532, 64664, 1287128, ...

363673, 484897, 5105101021, 6126121225, ...

12612, 241224, 482448, 964896, 19296192, ...

918189, 27363627, 45545445, 63727263, ...

other patterns, across two terms, (xy, yz), (xyz, yzw), (xyz, zwv):

38, 815, 1524, 2435, 3548, ...

3811, 81119, 111930, 193049, 304979, ...

235, 5711, 111317, 171923, 232931, ...

5 digits

explode ($x_i$) intertwined with a "marking" sequence:

1, 0, 4, 0, 9, 0, 1, 6, 0, 2, 5, 0, 3, 6, 0, 4, 9, ...

6, 1, 1, 1, 2, 1, 1, 2, 4, 1, 1, 4, 8, 1, 1, 9, 6, 1, 1, 1, 9, 2, ...

digits intertwined with constants or constant groups:

2, 4, 8, 106, 302, 604, 10208, 20506, 50102, ...

205, 4009, 80001, 10000200001, 1000006000009, ...

2126, 2225, 2326, 2429, 2624, 2821, 212020, ...

digits repeated as a function of position or value:

1166, 3366, 6644, 110000, 114444, 119966, ...

3, 66, 111222, 22224444, 4444488888, ...

8, 166, 322, 644, 122888, 255666, 511222, ...

1666666, 2255555, 333666666, 4444999999999, ...

"tailing" groups:

6753, 753, 53, 3, 6754, 754, 54, 4, 6755, 755, ...

32, 3, 64, 6, 128, 12, 1, 256, 25, 2, 512, 51, 5, 1024, ...

explode($x_i$), reverse($x_i$), sort($x_i$), firstdigit($x_i$) generally not easily inferred:

3, 4, 6, 8, 1, 1, 1, 1, 2, 2, 2, 3, 3, 4, 4, 4, 5, ...

8, 72, 46, 521, 612, 343, 215, ...

1289, 13468, 23678, 35566, 011237, ...

1, 2, 8, 2, 5, 6, 5, 1, 2, 1, 0, 2, 4, 2, 0, 4, 8, 4, 0, ...

## 6 miscellany

Some tricks which don't fit elsewhere, some sequences depending primarily on literal memorization.

permutation:

12, 21, 123, 231, 312, 1234, 2341, 3412, 4123, ...

binary counter:

1, 2, 1, 4, 1, 2, 1, 8, 1, 2, 1, 4, 1, 2, 1, 16, 1, 2, 1, 4, ...

factorial:

1, 2, 6, 24, 120, 720, 5040, ...

5, 10, 30, 120, 600, 3600, 25200, ...

Fibonacci:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

1, 4, 5, 9, 14, 23, 37, 60, 97, ...

1, 1, 1, 3, 5, 9, 17, 31, 57, 105, ...

2, 3, 4, 6, 9, 14, 22, 35, 56, ...

n ↑ n:

1, 4, 27, 256, 3125, 46656, ...

factors:

7, 8, 2, 4, 9, 3, 10, 2, 5, 11, 12, 2, 3, 4, 6, 13, 14, 2, 7, ...

binary coding:

1, 10, 11, 100, 101, 110, 111, 1000, 1001, ...

11, 110, 1001, 1100, 1111, 10010, 10101, ...

2, 23, 22, 233, 232, 223, 222, 2333, ...

11, 1022, 1122, 100333, 101333, 110333, 111333, ...

11, 104, 119, 10016, 10125, 11036, 11149, 100064, ...

<u>primes</u>:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, ...

1, 2, 2, 4, 2, 4, 2, 4, 6, 2, 6, ...

5, 13, 17, 29, 37, 41, 53, ...

<u>digits of real numbers of beloved memory</u>:

3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3, 2, 3, 8, 4, 6, ...

<u>subway stops (nonalgorithmic)</u>:

14, 18, 23, 28, 34, 42, 50, 59, 66, 72, 79, 86, 96, 103,
110, 116, 125, 137, 145, 157, 168, 181, 191, 207, 215, 225,
231, 238, 242 (from Sloane*)

*See N.J.A.Sloane's "A Handbook of Integer Sequences" (Academic Press, 1973) for other "puzzle sequences" and a wealth (over 2,300) of very interesting and on the whole rather difficult-to-infer sequences.