

# Project Euler #751: Concatenation Coincidence

Solved 23 December, 2021

A non-decreasing sequence of integers  $a_n$  can be generated from any positive real value  $\theta$  by the following procedure:

$$\begin{aligned}b_1 &= \theta \\b_n &= \lfloor b_{n-1} \rfloor (b_{n-1} - \lfloor b_{n-1} \rfloor + 1) \quad \forall n \geq 2 \\a_n &= \lfloor b_n \rfloor\end{aligned}$$

where  $\lfloor \cdot \rfloor$  is the floor function.

For example,  $\theta = 2.956938891377988 \dots$  generates the Fibonacci sequence:  
2, 3, 5, 8, 13, 21, 34, 55, 89,  $\dots$

The *concatenation* of a sequence of positive integers  $a_n$  is a real value denoted  $\tau$  constructed by concatenating the elements of the sequence after the decimal point, starting at  $a_1$ :  $a_1.a_2a_3a_4\dots$

For example, the Fibonacci sequence constructed from  $\theta = 2.956938891377988 \dots$  yields the concatenation  $\tau = 2.3581321345589 \dots$ . Clearly,  $\tau \neq \theta$  for this value of  $\theta$ .

Find the only value of  $\theta$  for which the generated sequence starts at  $a_1 = 2$  and the concatenation of the generated sequence equals the original value:  $\tau = \theta$ . Give your answer rounded to 24 places after the decimal point.

## Solution

generate\_series (generic function with 2 methods)

```
• function generate_series( $\theta$ , dec_places = 100)
•   # set starting conditions
•   B, A = [ $\theta$ ], [floor(Int,  $\theta$ )]
•   str_A_dec = ""
•   # iterate until sufficiently many digits calculated
•   while length(str_A_dec) < dec_places
•       # generate b and a
•       push!(B, floor(B[end]) * (B[end] - floor(B[end]) + 1))
•       push!(A, floor(B[end]))
•       # update string representation of decimal places
•       str_A_dec *= string(A[end])
```

concatenation\_coincidence (generic function with 2 methods)

```
• function concatenation_coincidence(a1, dec_places, iters=1000)
•     # set starting conditions
•     θ, str_θ = a1, string(a1)
•     # iterate towards solution
•     for _ in 1:iters
•         # generate series concatenation from current θ
•         str_τ = generate_series(θ, dec_places)
•         # if generated concatenation equals θ, return solution
•         str_τ == str_θ && return str_θ
•         # update θ to series concatenation
•         θ, str_θ = parse(BigFloat, str_τ), str_τ
•     end
•     return "No θ found in $iters iterations."
end
```

"2.223561019313554106173177"

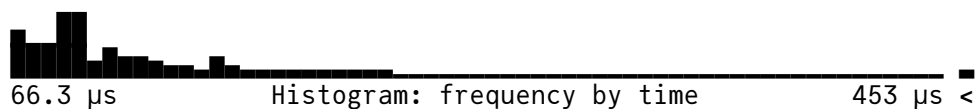
• concatenation\_coincidence(2, 24)

## Benchmark

• using BenchmarkTools

BenchmarkTools.Trial: 10000 samples with 1 evaluation.

Range (min ... max):	66.300 μs ... 8.614 ms	GC (min ... max):	0.00% ... 93.93%
Time (median):	96.200 μs	GC (median):	0.00%
Time (mean ± σ):	133.715 μs ± 220.412 μs	GC (mean ± σ):	5.04% ± 3.43%



Memory estimate: 65.02 KiB, allocs estimate: 1295.

• @benchmark concatenation\_coincidence(2, 24)

## Validation

• @assert generate\_series(2.956938891377988)[1:15] == "2.3581321345589"