COLLATZ ITERATION AND EULER NUMBERS?

MARKUS SIGG

For an integer $n > 0$ and an odd integer r, $1 \leq r < 2^n$, consider $c_{n,r}(1) := 2^n x + r$ with x being an indeterminate positive integer. Because $c_{n,r}(1)$ is odd, its first two Collatz iterates are $c_{n,r}(2) = 3 \cdot 2^n x + 3r + 1$ and $c_{n,r}(3) = 3 \cdot 2^{n-1} x + (3r+1)/2$. As long as the parity of $c_{n,r}(k)$ is known, we can continue with the iteration, up to an index $k_{n,r}$ where the parity of $c_{n,r}(k_{n,r})$ depends on the actual value of x. This means that $k_{n,r}$ is the first index k for which the coefficient of x in $c_{n,r}(k)$ is odd, which defines the set

$$T_{n,r} := \{c_{n,r}(1), \dots, c_{n,r}(k_{n,r})\}.$$

Let $R(n)$ be the set of those r for which there exist $u, v, w \in T_{n,r}$ with $u + v = w + 1$. We are interested in $\varrho(n) := |R(n)|$. As $r, 2^n + r \in R(n+1)$ for $r \in R(n)$, we have $\varrho(n+1) \geq 2\varrho(n)$.

It is easy to see that $R(1) = R(2) = \emptyset$ and so $\varrho(1) = \varrho(2) = 0$.

For $n = 3$ we have $r \in \{1, 3, 5, 7\}$ with

$$\begin{aligned}
T_{3,1} &= \{8x+1, 24x+4, 12x+2, 6x+1, 18x+4, 9x+2\},\\
T_{3,3} &= \{8x+3, 24x+10, 12x+5, 36x+16, 18x+8, 9x+4\},\\
T_{3,5} &= \{8x+5, 24x+16, 12x+8, 6x+4, 3x+2\},\\
T_{3,7} &= \{8x+7, 24x+22, 12x+11, 36x+34, 18x+17, 54x+52, 27x+26\}.
\end{aligned}$$

The identity $(6x+1) + (18x+4) = (24x+4) + 1$ shows that $1 \in R(3)$. No such identity exists for the other cases of r, so $R(3) = \{1\}$ and $\varrho(3) = 1$.

Similarly, for $n = 4$ one sees that $R(4) = \{1, 9, 11, 13\}$, so $\varrho(4) = 4$.

A Python program (see below; not optimised for speed) was used to generate the initial 20 terms of the sequence ϱ:

1	:	0	6	:	26	11	:	1013	16	:	32752
2	:	0	7	:	57	12	:	2036	17	:	65519
3	:	1	8	:	120	13	:	4083	18	:	131054
4	:	4	9	:	247	14	:	8178	19	:	262125
5	:	11	10	:	502	15	:	16369	20	:	524268

Surprisingly, these seem to be the Euler numbers A000295, i.e. $\varrho(n) = 2^{n-1} - n$. A proof of this for all n, presumably by establishing the recursion $\varrho(n) = 2\varrho(n-1) + n - 2$, has yet to be given.

Date: August 2, 2024.

Listing 1. Python program for generating the initial 20 terms of ϱ

```
def checkTrajectory(T):
    for u in T:
        for v in T:
            for w in T:
                if u[0] + v[0] == w[0] and u[1] + v[1] == w[1] + 1:
                    return 1

    return 0

for n in range(1, 21):
    rho = 0

    for r in range(1, 1 << n, 2):
        c = (1 << n, r)
        T = [ c ]

        while c[0] % 2 == 0:
            if c[1] % 2 == 0:
                c = (c[0] // 2, c[1] // 2)
            else:
                c = (3 * c[0], 3 * c[1] + 1)

            T.append(c)

        rho += checkTrajectory(T)

    print(n, ":", rho)
```

Freiburg, Germany

Email address: mail@markussigg.de