

$(3n+1)$ -Conjecture

August 12, 2020

1 # Python Program to check the $(3n+1)$ Conjecture

Powered by: Dr. Hermann Völlinger, DHBW Stuttgart(Germany); August 2020

See Wikipedia: https://en.wikipedia.org/wiki/Collatz_conjecture

The Collatz conjecture is a conjecture in mathematics that concerns a sequence defined as follows: start with any positive integer n . Then each term is obtained from the previous term as follows: if the previous term is even, the next term is one half of the previous term. If the previous term is odd, the next term is 3 times the previous term plus 1.

The conjecture is that no matter what value of n , the sequence will always reach 1.

The conjecture is named after Lothar Collatz, who introduced the idea in 1937, two years after receiving his doctorate.[1] It is also known as the $3n + 1$ problem, the $3n + 1$ conjecture, the Ulam conjecture (after Stanisław Ulam), Kakutani's problem (after Shizuo Kakutani), the Thwaites conjecture (after Sir Bryan Thwaites), Hasse's algorithm (after Helmut Hasse), or the Syracuse problem.[2][4] The sequence of numbers involved is sometimes referred to as the hailstone sequence or hailstone numbers (because the values are usually subject to multiple descents and ascents like hailstones in a cloud),[5][6] or as wondrous numbers.[7]

Paul Erdős said about the Collatz conjecture: "Mathematics may not be ready for such problems." [8] He also offered US\$500 for its solution.[9] Jeffrey Lagarias in 2010 claimed that based only on known information about this problem, "this is an extraordinarily difficult problem, completely out of reach of present day mathematics." [10]

2 Problem Solution

1. Create a function `collatz` that takes an integer n as argument.
2. Create a loop that runs as long as n is greater than 1.
3. In each iteration of the loop, update the value of n .
4. If n is even, set n to $n/2$ and if n is odd, set it to $3n + 1$.
5. Print the value of n in each iteration.

3 Program Explanation

1. The user is asked to input n .
2. The sequence is printed by calling `collatz` on n .

```
[2]: # Program/Source Code
# Here is the source code of a Python program to test Collatz conjecture for a
# → given number.
# The program output is shown below.

def collatz(n):
    while n > 1:
        print(n, end=' ')
        if (n % 2):
            # n is odd
            n = 3*n + 1
        else:
            # n is even
            n = n//2
        print(1, end='')

n = int(input('Enter n: '))
print('Sequence: ', end='')
collatz(n)
```

```
Enter n: 5555
Sequence: 5555 16666 8333 25000 12500 6250 3125 9376 4688 2344 1172 586 293 880
440 220 110 55 166 83 250 125 376 188 94 47 142 71 214 107 322 161 484 242 121
364 182 91 274 137 412 206 103 310 155 466 233 700 350 175 526 263 790 395 1186
593 1780 890 445 1336 668 334 167 502 251 754 377 1132 566 283 850 425 1276 638
319 958 479 1438 719 2158 1079 3238 1619 4858 2429 7288 3644 1822 911 2734 1367
4102 2051 6154 3077 9232 4616 2308 1154 577 1732 866 433 1300 650 325 976 488
244 122 61 184 92 46 23 70 35 106 53 160 80 40 20 10 5 16 8 4 2 1
```

```
[4]: # print current date and time

import time
print("date",time.strftime("%d.%m.%Y %H:%M:%S"))
print ("end")
```

```
date 12.08.2020 22:29:31
end
```