(3n+1)-Conjecture

September 28, 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.

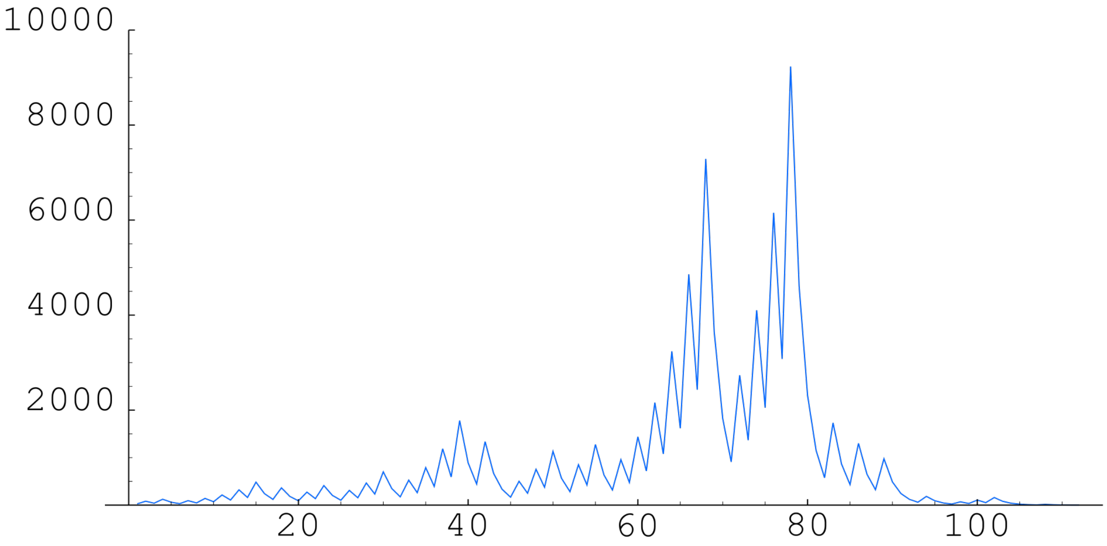
The conjecture is named after Lothar Collatz, who introduced the idea in 1937, two years after receiving his doctorate.[1] It is also known as the 3n + 1 problem, the 3n + 1 conjecture, the Ulam conjecture (after Stanisław Ulam), Kakutani's problem (after Shizuo Kakutani), the Thwaites conjecture (after 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]

[1]:
```
# Program Example for Number=27


print (" The sequence for n = 27, listed and graphed below, takes 111 steps (41␣
 ↪steps through odd numbers, in large font),")
print (" climbing to a high of 9232 before descending to 1.")

print␣
 ↪("****************************************************************************************


from IPython.display import Image

Image('collatz5-No27.png')
```

```
 The sequence for n = 27, listed and graphed below, takes 111 steps (41 steps
through odd numbers, in large font),
 climbing to a high of 9232 before descending to 1.
********************************************************************************
***************************************
```

[1]:



1.1 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.

1.2 Program Explanation

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

[2]:

```
print (" The sequence for n = 27, listed and graphed below, takes 111 steps (41␣
 ↪steps through odd numbers, in large font),")
print (" climbing to a high of 9232 before descending to 1.")
```

```
 The sequence for n = 27, listed and graphed below, takes 111 steps (41 steps
through odd numbers, in large font),
 climbing to a high of 9232 before descending to 1.
```

[3]:

```
# 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: 25
Sequence: 25 76 38 19 58 29 88 44 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

[5]:

```
# print current date and time

print ("***** current date and time***********")

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

```
***** current date and time***********
date 28.09.2020 18:27:29
end
```