

# Euler 0014

## The Problem:

We are trying to find the longest Collatz sequence under 1000000.

## Considerations:

The collatz sequence follows 2 simple rules:

- if  $n$  is even, then  $n/2$
- if  $n$  is odd, then  $n*3 + 1$

5 -> 16 -> 8 -> 4 -> 2 -> 1

6 -> 5 -> 4 -> 3 -> 2 -> 1

## The Approach:

The approach I'll take to this is to set up a dictionary of lengths at key  $n$ . If the key doesn't have a value then it generates a collatz sequence until it returns 1 or a key that has a length value

## The Code:

```
highest_starting = 1000000
lengths_dict = {1:1}
max_length = 1
length_key = 1

#the recursive function to generate collatz.
#Base case is satisfied by the lengths_dict above already having 1 initialized.
def update_lengths(number, lengths):
```

```
#print(number)
if lengths.get(number) == None:
    if number % 2 == 0:
        one_down = number/2
    else:
        one_down = number*3 + 1
    lengths = update_lengths(one_down, lengths)
    lengths[number] = lengths[one_down] + 1

return lengths

for i in range(1, highest_starting + 1):
    if lengths_dict.get(i) == None:
        lengths_dict = update_lengths(i, lengths_dict)

    if lengths_dict.get(i) > max_length:
        length_key = i
        max_length = lengths_dict.get(i)

print(length_key, max_length)
```

---

Revision #2

Created 2025-09-16 14:40:15 UTC by Maxwell

Updated 2025-09-16 14:43:01 UTC by Maxwell