POSET INTERPRETATION OF KNOWN COLLATZ SEQUENCES

Jack D’Angelo
December 6, 2022

1. Introduction

In this paper I will seek to investigate the relationships between odd numbers in known Collatz sequences and will use these relationships to develop a partial ordering of odd numbers. Finally, I develop a recursion that traverses these ordering’s antichains in to more quickly calculate the number of odd values in a Collatz sequence. This recursive rule can be used to generate a tree structure of infinite integer sets in which the depth of a node corresponds to its elements’ distance away from 1 in its Collatz sequence.

2. Chains from Collatz Sequences

(2.1) Collatz Conjecture definition. The Collatz Conjecture asks whether a simple recursion will always transform every positive integer into the number 1. The recursion is given as

$$a_n = \begin{cases} \frac{a_{n-1}}{2} & if\ a_{n-1} \equiv 0\ (mod\ 2) \\ 3a_{n-1} + 1 & if\ a_{n-1} \equiv 1\ (mod\ 2) \end{cases}$$

(1) $if\ a_0 = 17, then\ the\ sequence\ (until\ 1\ is\ reached)\ is\ as\ follows$:

17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1….

It has yet to be proven that all sequences produced by this recurrence eventually reach the value 1 as claimed in the conjecture and it is possible that there are infinite sequences or cycles for some input values. However, it has been confirmed by computer that all values up to 2^{68} do in fact converge to 1 after a certain number of steps.[1]

(2.1.1) We consider the <u>stopping time</u> of the sequence to be the smallest value *i* such that $a_i = 1$. For example, the sequence (1) above has a stopping time of 12.

For all known converging sequences, the stopping time is equivalent to the steps required to reach 1 because once the value 1 is reached, a cycle occurs:

(2) $if\ a_0 = 1, then\ the\ sequence\ is\ as\ follows$:

1, 4, 2, 1, 4, 2, 1 …

This produces an infinite cycle that is not relevant for the stopping time of any known sequence. Therefore, for the purposes of this paper, all sequences will terminate at 1. Further, this paper will limit its scope to the sequences such that $a_0 \in \mathbb{Z}^+ < 2^{68}$ for which the Collatz Conjecture has been empirically shown to hold true.

(2.2) Reduction to chains. Given that each sequence under consideration is known to begin at a value a_0 and terminate at 1, each sequence can be interpreted as a finite chain in which each vertex in the chain consists of the odd values in the sequence.

It is possible to ignore the even values of the sequences in these chains because it can be shown that any odd value in a Collatz sequence will lead to another odd value.
$Let\ there\ be\ an\ a_i\ such\ that\ a_i \equiv 1\ (mod\ 2)$, or a_i is odd. This implies:

$$a_{i+1} = 3a_i + 1$$

It is true that any two odd values multiplied together produce another odd. Once 1 is added to this result, it is ensured that a_{i+1} is even. a_{i+1} is then subsequently divided by 2 until an odd number is reached at a_{i+q} where q $\in \mathbb{Z}^+$.

As a result, it is possible for each sequence considered to be reduced to a chain of odd numbers that eventually ends in 1.

(1) Using the above example $a_0 = 17$, the original sequence

17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1

Is reduced to a sequence of only odd values while still retaining the sequence's properties:

(2) 17, 13, 5, 1

(2.2.1) A <u>chain</u> will refer to a single path down to 1 with vertices consisting only of the odd values in the Collatz sequence. Chains will be denoted in the form C_n such that *n* is the first odd value in the Collatz sequence.

(3) The chain of C_{17} is represented visually as such:

$$\begin{array}{c} 17 \\ | \\ 13 \\ | \\ 5 \\ | \\ 1 \end{array}$$

It is important to note that any value of the form $2^k * n$ where k $\in \mathbb{Z}^+$ yields the same C_n. For example, Collatz sequences that start with $a_0 = 17$ and $a_0 = 34$ both correspond to C_{17}. As such, for the purposes of this paper, only odd a_0 will be under consideration.

(2.2.2) It is often the case that one chain is the subset of another chain. For example, $\{C_{13}, C_5, C_1\} \subset C_{17}$. However, $C_{17} \not\subset C_{13}$

(4) This relation is shown below:

$$\begin{array}{cccc} C_{17} & & & \\ 17 & C_{13} & & \\ | & & & \\ 13 & 13 & C_5 & \\ | & | & & \\ 5 & 5 & 5 & \\ | & | & | & C_1 \\ 1 & 1 & 1 & 1 \end{array}$$

All chains considered will eventually meet at 1 meaning that $C_1 \subset C_n$ for all *n* considered. However, chains often meet earlier than 1 and will collapse into other chains. For instance, $C_{69} \not\subset C_{17}$ but $C_{13} \subset C_{17} \cap C_{13} \subset C_{69}$.

The Collatz sequence with $a_0 = 69$ produces the following:

69, 208, 104, 52, 26, **13**, 40, 20, 10, **5**, 16, 8, 4, 2, **1**

This is then reduced to the chain C_{69} from the bolded odd elements above:

$$\begin{array}{c} 69 \\ | \\ 13 \\ | \\ 5 \\ | \\ 1 \end{array}$$

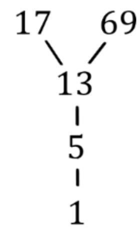
3. Posets from Collatz Sequences

(3.1) Chains to partial orderings. As shown above, it is possible for different odd numbers to lead to the same next odd number in a Collatz sequence, and hence for different chains to share other chains as a subset. This relationship can be used to establish a partial ordering:

Let A be a set of odd positive integers produced under the Collatz recurrence that is known to terminate at 1.

A partial ordering can be created under the relation *a* < *b* if $C_a \subseteq C_b$
Let the relation be represented by R.

Without loss of generality, let one such A = {69, 17, 13, 5, 1} represented visually as the poset



R = { (69, 69), (69,17), (69, 13), (69, 5), (69, 1), (17, 17), (17, 13), (17, 5), (17, 1), (13, 13), (13, 5), (13, 1), (5, 5), (5, 1), (1, 1) }

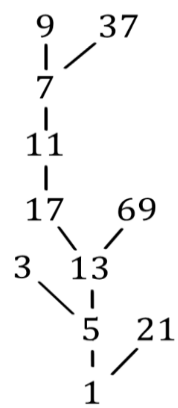
R is reflexive as the (69, 69), (17, 17), (13, 13), (5, 5), and (1, 1) are all elements of the relation.

R is antisymmetric because for any $m, n \in A, if\ (m,n) \in R\ \cap (n,m) \in R\ then\ m = n$

R is transitive because for any $m, n, k\ \in A, if\ (m,n) \in R\ \cap (n,\ k) \in R\ then\ (m,k)\ \in R$

Because R fulfills these three properties, R is a partial order on set A.
If we define C as the set of all possible chains C_n, a finite poset P can be created from any combination of elements of set C:

(1) Let P be the poset created by combining chains $C_3, C_9, C_{21},\ C_{69}, and\ C_{37}$



Organizing different chains as part of the same poset allows us to consider the length of a Collatz sequence in a different way than the traditional stopping time method. The <u>height</u> of the poset corresponds to how many odd values the largest chain is away from terminating, including 1. This is equivalent to the number of odd numbers in the Collatz sequence the poset represents. The height of the above poset with the longest chain of C_9 is 7.

(3.2) Lemma. Given the Collatz recursion, we can find which odd values will lead to same subsequent odd value, just as 17 and 69 both lead to 13.

Let x be an odd value and f(x) be the Collatz recursion applied as a function with input x.
If x is odd, then f(x) = 3x + 1. This will always produce an even number, implying:

$f^2(x)$= f(3x+1) = $\frac{3x+1}{2}$

Let g(x) = 4x+1 for odd x:

$$f(g(x))$$

$$f(4x+1) = 12x+4$$

$$f(12x+4) = 6x+2$$

$$f(6x+2) = 3x+1$$

$$f(3x+1) = \frac{3x+1}{2}$$

$$f^4(g(x)) = \frac{3x+1}{2}$$

Therefore,

$$f^4(g(x)) = f^2(x)$$

It follows that all values of a sequence of the form $b_n = 4b_{n-1} + 1$ when b_0 is odd will produce the same subsequent odd value under the Collatz recurrence if b_0 is odd.

NOTE: For purposes addressed later in this paper, even numbers are valid b_0 values despite not maintaining this property. For any even number x, $\frac{x-1}{4}$ will always produce a non-integer value.

(3.3) Antichains. The elements of a sequence $b_n = 4b_{n-1}$+1 represent incomparable elements of the same height in each poset. They will never be part of the same chain but will always lead to the same next vertex. Defining antichains as such is advantageous because it allows us to produce an infinite number of odd values with the same height, and hence same distance of odd values away from the 1 in a Collatz sequence.

(3.3.1) Because the values considered in the poset are defined to only be odd, each antichain has a head value b_0 such that $\frac{b_0-1}{4}$ results in a non-integer value.

Valid b_0 values:

$$7 = 4 * 1.5 + 1$$
$$2 = .25 * 4 + 1$$
$$19 = 4 * 4.5 + 1$$

Invalid b_0 values:

$$13 = 4 * 3 + 1$$
$$5 = 4 * 1 + 1$$
$$69 = 4 * 17 + 1$$

This recurrence has a closed form of $f_n = 4^n b_0 + 4^{n-1} + \frac{4^{n-1}-1}{3}$ so that one may quickly calculate very large values with a known height.

4. Garden of Eden Values

(4.1) Garden of Eden Values. Following empirical verification via a Python program, it was observed that all values belonging to the sequence $g_n = 6n + 3$ were always flagged as having no predecessor.

Following this observation, these values belonging to the sequence g_n were input into another program which sorted the values by height.

Let set *G* = {3, 9, 15, 21, 27 … }

The following table ranks the elements of *G* in order of the height of their chains:

Height	2	3	4	5	6	7	…
C_n	21	3	69	45	15	9	…

Upon discovering this ordering, this sequence's importance was confirmed using the OEIS. The sequence $A349873$[2] details this discovery, which was first posted by Johannes Koelman in 2021. Koelman's entry onto the OEIS names the elements of set *G* the Garden of Eden (GoE) values, a designation which will be adopted for the duration of this paper.

(4.2) Lemma. The GoE values are values which do not have an odd predecessor under the Collatz mapping.

Let $g \in G.\ \nexists C_n\ such\ that\ C_g\ \subset\ C_n$. In other words, chains starting with a GoE value are a subset of no other chain but itself.

$For\ C_g$ to be a strict subset of another chain would require that some odd integer n eventually leads to g under a Collatz mapping, which is impossible.

Proof by Contradiction:

Let y ∈ $\mathbb{Z}^+$ such that g = 6y + 3, the definition of a GoE value

Let *x* ∈ $\mathbb{Z}^+$ $such\ that\ x\ is\ odd$

Suppose that $\frac{3x+1}{2^k} = 6y + 3$

Then $3x + 1\ \equiv 6y + 3\ (mod\ 2^k)$

So $3x\ \equiv 6y + 2\ (mod\ 2^k)$

And hence $x\ \equiv 2y + \frac{2}{3}\ (mod\ 2^k)$

The final statement is logically inconsistent as no odd integer divided by 2^k can ever have a remainder with a denominator of 3. Therefore, GoE chains cannot be a strict subset of any other chain.

5. Antichain Recurrence

(5.1) Antichain Traversal. The antichain relationship described by $b_n = 4b_{n-1}$+1 and the Garden of Eden numbers described by the sequence $g_n = 6n + 3$ appear to have an interesting relationship. By moving along an antichain, it allows for the enumeration of odd values that have the same height has the b_0 value, which has already been discussed. It turns out that it is possible to use the GoE relationship to jump between antichains.

(5.2) Lemma. For values of $b_0 = 4b + 1\ such\ that\ b\ is\ not\ an\ integer$, if the height of b_0 is *h*, then the height of $6b_0 + 3$ is *h – 1.*

Let x be an odd positive integer such that x = 4b + 1, where b is not odd, making x a b_0 value. Let f(x) be the Collatz recursion applied as a function with input x.

Suppose the transformation 6x+3 is applied to the same x. Then:

$$f(6x + 3) = 18x + 10$$
$$f(18x + 10) = 9x + 5$$
$$f(9x + 5) = \frac{9x + 5}{2}$$

Next, suppose:

$$f(x) = 3x+1$$
$$f(3x+1) = \frac{3x+1}{2}$$

At this step, we must consider the two cases to see if $\frac{3x+1}{2}$ *is odd or even.*

Case 1: $\frac{x-1}{4}\ is\ even$

If x is of the form x = 4n+1 where n is even, then:

$$\frac{3(4n + 1) + 1}{2}$$

$$\frac{12n + 4}{2}$$

$$6n+2$$

For Case 1, the value will always be even, meaning that we cannot draw any conclusions in comparison to the 6x+3 transformation. This failed case will be considered in section (5.3).

Case 2: $\frac{x-1}{4}$ *is an integer plus* $\frac{1}{2}$

x is of the form x = 4(n + .5) + 1 = 4n + 3, where n is a positive integer

$$\frac{3(4n + 3) + 1}{2}$$

$$\frac{12n + 10}{2}$$

$$6n + 5$$

For Case 2, the value will always be odd, so we continue the proof assuming that x exists such that x = 4n + 1, where n is a non-integer value. As just shown, this implies that $\frac{3x+1}{2}$ *is odd, meaning that we complete the proof with:*

$$f\left(\frac{3x+1}{2}\right) = \frac{9x+5}{2}$$

(5.3) The Failed Case. The conclusion made above does not hold for values of b_0 when b_0 is even. In this case, it was empirically discovered that the transformation $3b_0 + 1$ will make the appropriate jump between antichains.

Claim: the transformation $3b_0 + 1$ will produce a value in an antichain one closer to value 1, similar to the GoE jumps.

Let x = 2k be and even integer and f(x) be the Collatz recursion applied as a function with input x. First, the initial shift must occur:

$$3(2k) + 1$$
$$6k + 1$$

We must show that 6k+1 has a height of one less than the odd numbers on the same antichain as x. The first such value is 4x+1.

4(2k) + 1 = 8k + 1
f(8k + 1) = 24k + 4
f(24k + 4) = 12k + 2
f(12k + 2) = 6k + 1

Recall that only odd values are significant for a chain's height. In other words, the height of a Collatz sequence is only updated when the 3x+1 condition is triggered. As shown above, it only took one of such steps for the values to become equivalent. Therefore, when a b_0 value is even, instead of applying the transformation $6b_0 + 3$ to reach the next odd number in the Collatz sequence, the transformation $3b_0 + 1$ should be applied.

(5.4) Define New Recursion. By combining these properties, it is possible to define a new recursion that traverses a Collatz poset's antichains rather than chains:

$$d_n = \begin{cases} \dfrac{(d_{n-1} - 1)}{4} & if \ \dfrac{(d_{n-1} - 1)}{4} \equiv 1 \ (mod\ 2) \\ 3d_{n-1} + 1 & if \ d_{n-1} \equiv 0 \ (mod\ 2) \\ & else \\ 6d_n + 3 & \end{cases}$$

$$where\ d_0\ is\ an\ odd\ positive\ integer < 2^{68}\ and\ the\ sequence\ terminates\ if\ 1\ is\ reached$$

(5.4.1) The height of recurrence d_n is defined as the number of antichains traversed. As explained before, once a b_0 value is reached, the Garden of Eden relation is used to skip to the antichain with the next lowest height. In terms of the recurrence, this means that the height of d_n is equivalent to the number of times that the transformation $6d_n + 3$ is triggered.

It is important to note that by the way we defined antichains, 1 is a b_0 value on the same antichain as 5, 21, 85, etc. This results in the following relationship between heights:

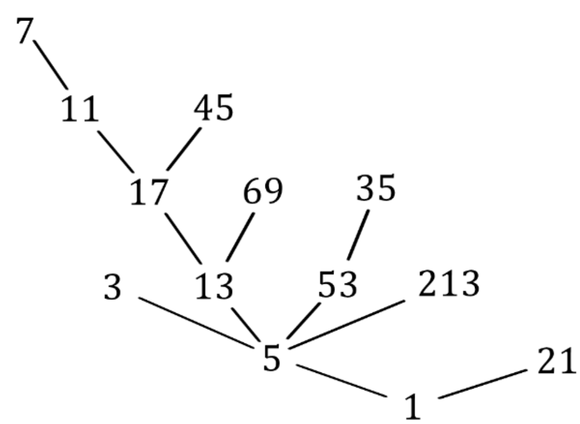
Let the height of a Collatz sequence is given as h_C and the height of the antichain sequence defined in (5.4) to be h_d. It follows that:

$$h_C = h_d + 1 \text{ for an odd starting value} > 1$$

***Note that if 1 is the input value, both sequences have a height of 1.*

(5.4.2) Both the Collatz sequence a_n and the sequence d_n can be visualized as posets defined earlier in this paper via Hasse diagrams. For example, while the Hasse diagram for the Collatz sequence starting with $a_0 = 7$ would only consist of the chain C_7, it is easy to enumerate values belonging to the antichains of C_7 by using the $b_n = 4b_{n-1} + 1$ recursion to produce a poset consisting of a combination of many other chains as well:

Let the poset P below consist of the combination of the chains $C_7, C_{45}, C_{69}, C_{35}, C_3, C_{213}, and\ C_{21}$:

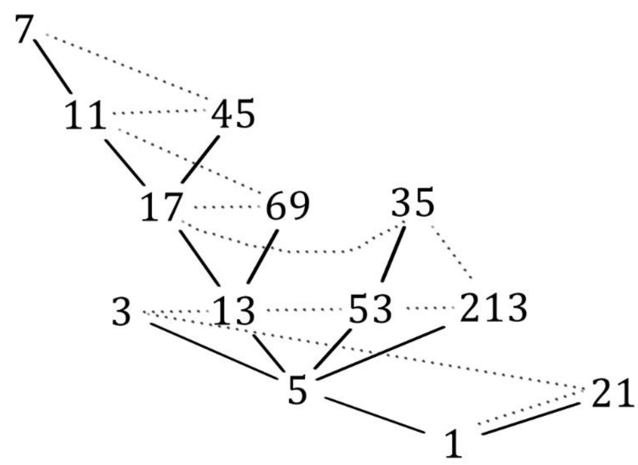


***Note that 21, 3, 213, and 45 are all GoE values, which cannot have any predecessors*

If we enumerate the antichain sequence beginning with $d_0 = 7$, we will get the sequence:

7, 45, 11, 69, 17, 35, 213, 53, 13, 3, 21, 5, 1

This sequence can be represented below as a path by dotted lines on Hasse diagram:



6. Transfinite Tree Generation

NOTE: This section is currently a work in progress and lacks the same formality as the rest of the paper. For now, it will be used as a record for any ideas and progress made that is incomplete.

(6.1) Tree Definition. Given the observations made when organizing these sequences into partially ordered sets, it is possible to reinterpret and reverse engineer these values into an infinitely generative tree structure spawning from a single source node.

Let $\mathcal{T} = \{N\}$, a tree with a set of nodes *N*. Each node $n \in N$, is defined by an infinite set defined by the recursion that produces odd numbers of the same height in the Collatz sequences.

$$n = \{a_n |\ a_n = 4a_{n-1} + 1\}$$

The source of the tree will be the node $n_0 = \{a_n |\ a_n = 4a_{n-1} + 1, a_0 = 1\}$. Enumerated, the elements of this infinite source set are:

1, 5, 21, 85, 341...

(6.2) Generation Rule. The generation of the children at each depth of the tree will be determined by the inverses of the recursive rules from above. At a given depth *d*, the children at *d+1* are generated by specific elements of each node set.

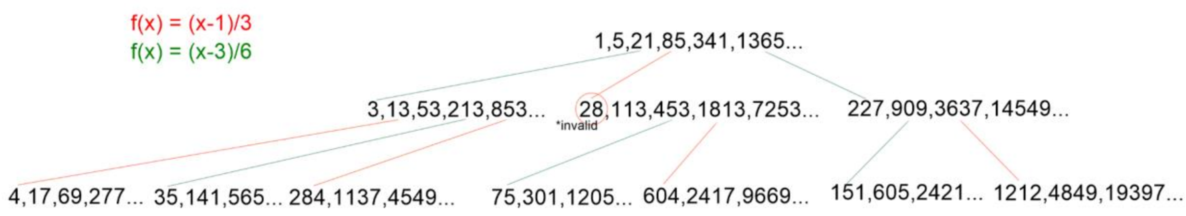
For each $e \in n$ where $e = 6x + 3\ or\ e = 3x + 1\ for\ some\ x \in \mathbb{N}$, a child is formed at *d+1* such that $a_0 = x$.

There is a limiting condition on which elements can be selected to generate a child. If a given element *e* can be traced directly back to an element of the source node by either the reverse of 6x+3 or 3x+1, then it cannot be used for generation. This is extremely important as it maintains the tree structure and keeps a 1 to 1 correspondence between the tree's depth and the height of the element in its Collatz sequence. All such values cause a collision with the source and are a result of the repeated recursion of 6x+3 or 3x+1 with a starting value that is an element of the source node. It also is invalid if it is a mix of said recursions. These elements are invalid candidates to spawn children. Here are some examples of invalid candidates:

$$\frac{129-3}{6} = 21, 21 \in n_0$$
$$\frac{769-1}{3} = 256, \frac{256-1}{3} = 85, 85 \in n_0$$
$$\frac{28-1}{3} = 9, \frac{9-3}{6} = 1, 1 \in n_0$$

It is worth noting that there is no explicit method other than reverse engineering the recursion that can easily/quickly verify whether an element is invalid. However, for the purposes of the construction of the tree, we will assume that the validity of each candidate is known.

Because of the nature of the generation, after *d = 0* (which only has a single source node), each depth should have infinite nodes because each set has infinite elements that are valid candidates. Below is a limited view of a subtree to demonstrate how generation works:



(6.3) Questions.

- Can it be shown that n_0 is unique in its ability to be a source?
- Is the cardinality of this tree $\aleph_0$?
- Are there integers that are not generated by this tree? If so, does this imply infinite integers on a graph that shares no elements with $\mathcal{T}$?

7. Conclusions

(7.1) Computer Verification. To verify height property and observe the relationship between a_n and d_n I used the Python program discussed earlier to test a large volume of input values. Specifically, I tested the first 1,000,000 odd numbers and confirmed that for each of these excluding 1, the height property discussed in (5.4.1) proves true. Further, the program showed that for 786,216 of the 1,000,000 odd numbers tested, the stopping time of the d_n sequence was less than or equal to the stopping time of the corresponding Collatz sequence.
(7.2) Discussion. While the interpretation of the Collatz sequences as finite posets is intriguing, it is limited by the unresolved nature of the Collatz Conjecture. However, the antichain recursion discovered certainly reveals a lot about the properties of the Collatz sequences and provides much insight into how these numbers interact, particularly that reducing the problem to just considering odd integers may be a step in the right direction. This recursion's relationship with the Collatz sequences requires further investigation.

I have shown the antichain recursion to retain the height property of the Collatz sequences, although because of the modulus arithmetic involved, this method takes more time to calculate. However, further investigation is warranted as my algorithm represents a new lens to apply to the Collatz Conjecture.

References

1. Barina, D. Convergence verification of the Collatz problem. *J Supercomput* **77**, 2681–2688 (2021). https://doi.org/10.1007/s11227-020-03368-x
2. oeis.org
3. **See code below

Python code; utilized to test inputs with up to 45 digits (limitations due to overflow errors). Code produces heights and stopping times of first 1,000,000 positive odd integers for both Collatz sequences and defined antichain sequences, then outputs a comparison:

```
i = 1
greaterDif = 0
equal = 0
while (i < 2000000):
    height = 0
    a = i
    stepCount = 0
    original = a
    width = []
    widthCount = 0
    primeCount = 0
    while(a > 1):
        if((a-1) % 4 == 0 and ((a-1)/4) % 2 != 0):
            widthCount += 1
            a = (a-1)//4
            stepCount += 1
        elif((a-1) % 4 == 0 and ((a-1)/4) % 2 == 0):
            a = 2 * a + 1
            widthCount += 1
            stepCount += 1
        else:
            width.append(widthCount)
            widthCount = 0
            a = 6*a+3
            height += 1
            stepCount += 1
        width.append(widthCount);
    myCount = stepCount
    a = i
    heightCollatz = 0
    stepCount = 0
    while(a > 1):
        if(a % 2 == 0):
            a = a/2
            stepCount += 1
        else:
            stepCount += 1
```

```python
            heightCollatz += 1
            a = 3 * a + 1
    collatzCount = stepCount
    if(myCount > collatzCount):
        greaterDif += 1
    if(height + 1 == heightCollatz):
        equal += 1
    i += 2
print("This many were larger than Collatz: ", greaterDif)
print("This many had same height: ", equal)
```

C++ code; used to quickly enumerate Collatz sequences. Code represents a “Seed” class in which the seed represents the input to the Collatz function. It counts both the “reduced” stopping time and stopping time. Full program allows for these seed objects to easily be manipulated and displayed via an interactive menu.

```cpp
class Seed
{
public:

      Seed();
      void setSeedNum();
      void expansion();
      void setSeedNum(int);
      void standardConvergence(int);
      void factoredConvergence(int);
      int getSeedNum();
      int getIterationCountStandard();
      int getIterationCountFactored();
private:
      int seedNum = 0;
      int iterationCountStandard = 0;
      int iterationCountFactored = 0;
};

void Seed::setSeedNum()
{
      int temp = 0;
      cout << "Input new seed: ";
      cin >> temp;
      seedNum = temp;
}


void Seed::setSeedNum(int sd)
{
      seedNum = sd;
}

int Seed::getSeedNum()
{
      return seedNum;
```

```
}

int Seed::getIterationCountStandard()
{
	return iterationCountStandard;
}

int Seed::getIterationCountFactored()
{
	return iterationCountFactored;
}


void Seed::expansion()
{
	for (int i = 0; i < 1000; i++)
	{
		cout << (3 * log2(pow(2, i) + 1) + 1) / (pow(4, i)) << endl;

	}
}

void Seed::standardConvergence(int seed)
{
	while (seed != 1)
	{
		if (seed % 2 == 0)
		{
			seed = seed / 2;
		}
		else
		{
			seed = 3 * seed + 1;
		}
		iterationCountStandard++;
	}
}

void Seed::factoredConvergence(int seed)
{
	while (seed != 1)
	{
		for (int i = 0; pow(2, i) <= seed; i++)
		{
			int power = pow(2, i);
			if (power == seed)
			{
				seed = seed / power;
				iterationCountFactored++;
				break;
			}
			else if (seed % power == 0)
			{
```

```
                        continue;
                    }
                    else
                    {
                        seed = seed / pow(2, i - 1);
                        if (pow(2, i - 1) != 1)
                        {
                            iterationCountFactored++;
                        }
                        break;
                    }
                }
                if (seed % 2 != 0 && seed != 1)
                {
                    seed = 3 * seed + 1;
                    iterationCountFactored++;
                }
            }
            iterationCountFactored++;
        }
```