

k -Parent Aliquot Numbers: Counting Pre-Images Under the Sum-of-Proper-Divisors Function

Gavin Guinn

Supervisor: Michael J. Jacobson

December 2021

1 Introduction

If we examine numbers in the set $\{6, 28, 496, 8128, \dots\}$ we notice that every member n is equal to the sum of its divisors with the exclusion of itself,

$$1 + 2 + 3 = 6.$$

This of course is the definition of a *perfect number*, a property that has fascinated number theorists since antiquity. The *sum-of-divisors* function is defined for $n > 1$ by

$$\sigma(n) = \sum_{d|n} d.$$

The *proper-divisors* of n refers to the subset of the divisors which excludes itself, the *sum-of-proper-divisors* is defined for $n > 1$ by

$$s(n) = \sigma(n) - n,$$

thus n is a perfect number if and only if $s(n) = n$. The sum-of-proper-divisors is a natural way to consider perfect numbers but is certainly not limited to that domain. For instance consider $s(9) = 4$ which has the property $n < s(n)$, such numbers are *deficient* while numbers that satisfy $n > s(n)$ are *abundant*.

Aliquot sequences are produced by iterating $s(n)$, for example consider the sequence starting at 12:

$$s(12) = 1 + 2 + 3 + 4 + 6$$

$$s(16) = 1 + 2 + 4 + 8$$

$$s(15) = 1 + 3 + 5$$

$$s(9) = 1 + 3$$

$$s(4) = 1 + 2$$

$$s(3) = 1$$

In this case the sequence has terminated as $s(n)$ is only defined for $n > 1$, this sequence is also bounded as no value is greater than 16. A sequence does not have to terminate to be bounded, since perfect numbers are defined by $s(n) = n$ an aliquot sequence that reaches a perfect number will cycle. The members of a cycle of length 2 are referred to as an *amicable pair*, for instance

$$s(1184) \rightarrow s(1210) \rightarrow s(1184) \rightarrow \dots$$

It seems that cycles can form with arbitrary length [3] note a cycle of length 24, numbers in such cycles are *sociable*. Aliquot sequences are sometimes said to terminate in a cycle, we will only refer to aliquot sequences that reach 1 to have terminated while referring to the latter case as a bounded sequence. Aliquot sequences are formalized by [1] in the following way:

Definition 1.1. The aliquot sequence starting at $a_0 > 1$ is defined by

$$a_0, a_1 = s(a_0), a_2 = s^2(a_0), \dots$$

Proposition 1.1. An aliquot sequence is bounded if and only if

$$\exists a_n, \forall a_m \text{ s.t. } a_n \geq a_m$$

The aliquot sequence of 276,

$$s(276) \rightarrow s(396) \rightarrow s(696) \rightarrow s(1104) \rightarrow s(1872) \rightarrow \dots$$

is the smallest example that has not been shown to be bounded, [12] has followed the it's growth over 2090 iterations to values with over 200 digits! The dichotomy between the seemingly unbounded growth of the 276 sequence and the boundedness of the 12 sequence gets to the most important open question relating to aliquot sequences:

Conjecture 1.1 (Catalan-Dickson). All aliquot sequences are bounded and thus must terminate or enter a cycle. [2]

Conjecture 1.2 (Guy-Selfridge). The Catalan-Dickson conjecture is false, perhaps most aliquot sequences are unbounded. [6]

The computational exploration of these sequences that has taken place since Catalan originally proposed his conjecture in 1888 has certainly given credence to the Guy-Selfridge point-of-view. However a trend in the data is far from a proof, under our current understanding it is entirely possible that the 276 sequence could trip over a prime or enter a cycle on some future iteration.

1.1 Motivation

What statistics can be compiled on the behaviour of $s(n)$ to argue for or against the Calatan-Dickson Conjecture? The authors of [3] consider numerical and analytical approaches to determine the average of $s(n)/n$. The average order of an arithmetic function is a simple function that has the same behaviour on average as the less tractable function. Define $\mathcal{S}(n)$ to be the average order of $s(n)$, when computing the average of $s(n)/n$ we are numerically estimating $\mathcal{S}(n)$. When considering this metric [3] restrict the domain to evens to examine $\mathcal{S}(2n)$. I believe the impetus for this decision can be traced to [6] who heuristically argue that unbounded sequences exist:

- **if** $\mathcal{S}(2n) > 2n$ and,
- **if** other effects on the growth of a sequence are relatively minor compared to the tendency for aliquot sequences to preserve parity,
- **then** we would expect most even aliquot sequences that reach large values to be unbounded.

This argument seems to match the numerical evidence as seemingly unbounded sequences such as $s(276)$ [12] appear to maintain constant even parity.

The method used to either analytically determine or numerically compute $\mathcal{S}(2n)$ impacts the parameters value. While not explicitly stated by Guy and Selfridge, [1] indicate that arithmetic mean of $s(2n)/2n$ was used to determine $\mathcal{S}(2n)$ in constructing their heuristic argument for unboundedness.

Theorem 1.1. Arithmetic Mean of $s(2n)/2n$

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \frac{s(2n)}{2n} = \left(\frac{5\pi^2}{24} - 1 \right) = 1.0562...$$

With this metric we have $\mathcal{S}(2n) = (2n)(1.0563...)$ and thus $\mathcal{S}(2n) > 2n$. However as pointed out by [1] the sequence $\{5, \frac{1}{5}, 5, \frac{1}{5}, \dots\}$ is bounded but its arithmetic mean tends to 2.6, a better metric for evaluating the boundedness of a sequence is to consider the geometric mean.

Theorem 1.2. Geometric Mean of $s(2n)/2n$

$$\mu = \lim_{N \rightarrow \infty} \exp \left(\frac{1}{N} \sum_{n=1}^N \log s(2n)/2n \right) = 0.969...$$

With this more accurate metric we have $\mathcal{S}(2n) < 2n$ which suggests that even aliquot sequences will in fact decrease on average! This result certainly throws cold water on the heuristic argument of Guy and Selfridge. However the monumental growth of sequences like $s(276)$ suggests that this metric does not completely describe sequence behaviour.

To improve this model of sequence behaviour the authors of [3] consider the image of $s(2n)$ when computing $\mathcal{S}(2n)$. Consider the solutions to $s^{-1}(316)$,

$$s(192) = s(304) = s(344) = s(412) = 316,$$

since multiple sequences are tributary at 316 it seems reasonable to conclude that the value of $s(316)/316$ impacts average sequence growth more than numbers with fewer pre-images. [3] explores this by constructing an average of $s(2n)/2n$ weighted by $\#s^{-1}(2n)$. Counting the number of pre-images under $s(2n)$ was certainly on the mind of Dr. Guy when he posed the question that motivates this project:

Think of a number!! Say 36%, which is nice and divisible. It appears that about 36% of the even numbers are "orphans".

Divide by 1. For about 36% of the (even) values of n there is just one positive integer m such that $s(m) = n$. These values of n have just one "parent".

Divide by 2. About 18% of the even values of n have exactly two parents.

Divide by 3. About 6% of the even values of n have three parents.

Divide by 4. About 1.5% of the even values of n have just 4 parents.

This suggests that $1/(e \cdot p!)$ of the even numbers have p parents.

Experiments suggests that these values are a bit large for small values of p and a bit small for larger values of p . Can anything be proved?

Definition 1.2. n is a k -parent aliquot number if $s^{-1}(n)$ has k solutions.

These percentages refer to the density of even k -parent numbers or the probability of encountering a k -parent number when choosing randomly from all even natural numbers. There are a suite of colourful names to refer to numbers outside of the range of $s(\cdot)$ including *non-aliquots* and *untouchable numbers*, for the sake of consistency I will be referring to all such numbers as *non-aliquot*.

2 Modeling the Density of k-Parent Numbers

2.1 Pollack-Pomerance Model of Non-Aliquot Density

Consider the following set,

$$N_a = \{2, 5, 52, 88, 96, 120, 124, 146, 162, 188, 206, 210, 216, 238, 246, 248, \dots\}$$

which contains non-aliquot numbers [11], if $n \in N_a$ then $\nexists m$ s.t. $s(m) = n$, in symbols $\#s^{-1}(n) = 0$. The name non-aliquot implies that these numbers do

not appear in aliquot sequences, however every sequence begins with a member of N_a ; if $\#s^{-1}(n) > 0$ then there must be some preceding number in the same sequence.

We are interested in the natural density of N_a ,

$$d(N_a) = \lim_{n \rightarrow \infty} \frac{|N_a \cap [1, n]|}{n},$$

a method of determining the proportion of numbers that are non-aliquot. Using probabilistic methods [9] precisely model the density of non-aliquot numbers.

Conjecture 2.1. *The set of non-aliquot numbers has asymptotic density Δ ,*

$$\Delta = \lim_{y \rightarrow \infty} \frac{1}{\log y} \sum_{\substack{a \leq y \\ 2|a}} \frac{1}{a} e^{-a/s(a)}.$$

This model is an application of the "balls into bins" statistics problem, estimating the result after throwing m balls randomly into n bins;

$$\#[\text{empty bins}] = n(1 - \frac{1}{n})^m.$$

To apply this technique the authors begin by defining a series of disjoint subsets,

$$A_y = \text{lcm}[1, 2, \dots, y]$$

and for each divisor $a|A_y$ uniquely define a set T_a ,

$$T_a = \{n : \gcd(n, A_y) = a\}.$$

For example consider the case $A_3 = 6$, we have $a \in \{1, 2, 3, 6\}$ and the following sets T_a :

a	T_a
1	$\{1, 5, 7, 11, 13, 17, 19, \dots\}$
2	$\{2, 4, 8, 14, 16, 20, 22, \dots\}$
3	$\{3, 9, 15, 21, 27, 33, 39, \dots\}$
6	$\{6, 12, 18, 24, 30, 36, 42, \dots\}$

Define bounds on the set,

$$T_a(x) = T_a \cap [1, x].$$

Utilizing the asymptotic properties of T_a the authors set up assumptions needed to apply *balls in bins*:

- Assume that $s(\cdot)$ maps T_a to T_a for each $a|A_y$, asymptotically true if $y \rightarrow \infty$ and $T_a = \{n > e^{e^y} \mid \gcd(n, A_y) = a\}$.

- Assume $\forall n \in T_a$ that $\sigma(n)/n \approx \sigma(a)/a$, asymptotically true up to sets of vanishing density as $y \rightarrow \infty$.
- Assume that $s(\cdot)$ is a random map.

To apply this model we need values for the number of *bins* n and *balls* m ; naturally we use the bounded sets T_a as a analog for the number of *bins*,

$$n = |T_a(x)|.$$

For an *balls* m we need to find a range of T_a such that $s(\cdot)$ maps these numbers into $T_a(x)$. Utilizing our assumptions that that $s(\cdot)$ maps T_a to T_a and $\sigma(n)/n \approx \sigma(a)/a$ we can choose a value for m such that

$$\forall h \in T_a \leq m, s(h) \in T_a(x).$$

We are choosing m to be the largest value in the set T_a such that $s(\cdot)$ maps all values less or equal to m into $T_a(x)$. It is useful to observe that every member of a T_a by assumption shares the ratio $a/s(a)$,

$$\frac{a}{s(a)} = \frac{1}{(\sigma(a)/a) - 1}.$$

Utilizing this assumption we set our number of *balls* m to

$$m = |T_a(x \cdot (a/s(a)))|.$$

Plugging in values we have the following expression,

$$\left(1 - \frac{1}{n}\right)^m = \left(1 - \frac{1}{|T_a(x)|}\right)^{|T_a(x \cdot a/s(a))|}$$

which models the probability that some member of $T_a(x)$ is non-aliquot.

The authors note a useful asymptotic equivalence,

$$|T_a(x)| \sim \frac{\phi(A_y) \cdot x}{A_y \cdot a}.$$

Since we are interested in the asymptotic behavior we take the limit of the bound x and substitute in the equivalence,

$$\begin{aligned} \lim_{x \rightarrow \infty} \left(1 - \frac{1}{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot a}\right)}\right)^{(\phi(A_y) \cdot x)/(A_y \cdot s(a))} \\ = e^{-a/s(a)}. \end{aligned}$$

This expression models the probability that a member of the unbounded set T_a is non-aliquot. An intuition for this simplification can be gained by considering a definition of Euler's number,

$$\lim_{x \rightarrow \infty} \left(1 - \frac{1}{x}\right)^x = e.$$

At this point we have a model for non-aliquot density within any set T_a , we would like to extend this to model non-aliquot density over all naturals. To accomplish this we again utilize the equivalence,

$$|T_a(x)| \sim \frac{\phi(A_y) \cdot x}{A_y \cdot a},$$

noticing that the ratio $\phi(A_y)/(a \cdot A_y)$ expresses the probability that some natural is in T_a , see [4] for a similar treatment. Supposing $n \in T_a$ and $s^{-1}(n)$ are independent events model the joint probability,

$$\mathbb{P}[T_a \text{ and non-aliquot}] = \frac{\phi(A_y)}{A_y \cdot a} \cdot e^{-a/s(a)}.$$

Since each set T_a is disjoint we can apply the additive property to find the natural density of non-aliquot numbers:

$$\begin{aligned} \Delta &= \lim_{y \rightarrow \infty} \sum_{\substack{a|A_y \\ 2|a}} \frac{\phi(A_y)}{a \cdot A_y} \cdot e^{-a/s(a)} \\ &= \lim_{y \rightarrow \infty} \frac{\phi(A_y)}{A_y} \sum_{\substack{a|A_y \\ 2|a}} \frac{1}{a} e^{-a/s(a)} \end{aligned}$$

We restrict a to even values as a 's parity determines the parity of the entire set T_a and we know that the density of odd aliquot orphans quickly vanishes

2.2 k-Parent Generalization

Note: Unedited copy from final report for NSERC workterm.

Todo: Clean up non-formal language.

Luckily people interested in statistics have done the work for us, say we have n 'bins', m 'balls', and want to know the probability of a specific bin has exactly k balls.

$$\mathbb{P}[k \text{ balls in bin}] = \binom{m}{k} \frac{1}{n^k} \left(1 - \frac{1}{n}\right)^{m-k}$$

This gives a fairly simple strategy to generalize the Pollack/Pomerance heuristic model, simply swap out the probability of getting 0-balls in a bin for the general equation. Plugging in the same values of:

$$n = \frac{\phi(A_y) \cdot x}{A_y \cdot a}$$

And:

$$m = \frac{\phi(A_y) \cdot x}{A_y \cdot s(a)}$$

We get the following monster:

$$\mathbb{P}[k \text{ parents}] = \lim_{x \rightarrow \infty} \binom{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)}}{k} \frac{1}{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot a}\right)^k} \left(1 - \frac{1}{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot a}\right)}\right)^{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} - k}$$

This expression cleans up quite nicely (see section Probability Simplification), the probability of some $n \in T_a$ having k parents:

$$\mathbb{P}[k \text{ parents}] = \frac{a^k}{k! \cdot s(a)^k} \cdot e^{-a/s(a)}$$

We then simply slap this probability expression into the Pollack and Pomerance result giving the density of even k parent aliquot numbers over all naturals:

$$\Delta_k = \lim_{y \rightarrow \infty} \frac{\phi(A_y)}{A_y} \sum_{\substack{a|A_y \\ 2|a}} \frac{a^{k-1}}{k! \cdot s(a)^k} \cdot e^{-a/s(a)}$$

The authors continue to establish an easier to compute expression for Δ :

$$\Delta = \lim_{y \rightarrow \infty} \frac{\phi(A_y)}{A_y} \sum_{\substack{a|A_y \\ 2|a}} \frac{1}{a} e^{-a/s(a)} \quad (3.1)$$

$$= \lim_{y \rightarrow \infty} \frac{1}{\log y} \sum_{\substack{a \leq y \\ 2|a}} \frac{1}{a} e^{-a/s(a)} \quad (3.4)$$

This suggests that the density of k parent numbers can be simplified to:

$$\Delta_k = \lim_{y \rightarrow \infty} \frac{1}{\log y} \sum_{\substack{a \leq y \\ 2|a}} \frac{a^{k-1}}{k! \cdot s(a)^k} \cdot e^{-a/s(a)}$$

Computed estimates of this limit are given in Table 1.

Δ_k	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$y = 10^2$	0.143753	0.162516	0.103907	0.049607	0.019508	0.006589
$y = 10^3$	0.155222	0.164297	0.100006	0.046153	0.017759	0.005918
$y = 10^4$	0.161059	0.165306	0.098209	0.044498	0.016892	0.005575
$y = 10^5$	0.164577	0.165923	0.097137	0.043506	0.016373	0.005369
$y = 10^6$	0.166923	0.166336	0.096423	0.042846	0.016027	0.005232
$y = 10^7$	0.168599	0.166631	0.095912	0.042373	0.015779	0.005133
$y = 10^8$	0.169857	0.166853	0.095530	0.042019	0.015594	0.005060
$y = 10^9$	0.170834	0.167025	0.095232	0.041744	0.015449	0.005003
$y = 10^{10}$	0.171617	0.167162	0.094994	0.041524	0.015334	0.004957
$y = 2(10^{10})$	0.171822	0.167198	0.094931	0.041466	0.015303	0.004945

Table 1: Computed estimates of k -parent density heuristic model, note Δ_0 matches [9] nonaliquot density at reported bound $y = 2(10^{10})$.

2.3 Open Work

1. Prove:

$$\begin{aligned}
\Delta_k &= \lim_{y \rightarrow \infty} \frac{\phi(A_y)}{A_y} \sum_{\substack{a|A_y \\ 2|a}} \frac{a^{k-1}}{k! \cdot s(a)^k} \cdot e^{-a/s(a)} \\
&= \lim_{y \rightarrow \infty} \frac{1}{\log y} \sum_{\substack{a \leq y \\ 2|a}} \frac{a^{k-1}}{k! \cdot s(a)^k} \cdot e^{-a/s(a)}
\end{aligned}$$

2. Given that:

$$\Delta_k = \lim_{y \rightarrow \infty} \frac{1}{\log y} \sum_{\substack{a \leq y \\ 2|a}} \frac{a^{k-1}}{k! \cdot s(a)^k} \cdot e^{-a/s(a)}$$

Estimates the density of **even** k parent numbers over all naturals, does:

$$\Delta_k = \lim_{y \rightarrow \infty} \frac{1}{\log y} \sum_{a \leq y} \frac{a^{k-1}}{k! \cdot s(a)^k} \cdot e^{-a/s(a)}$$

Estimate the density of **all** k -parent numbers over all naturals? (note that the condition $2|a$ is removed from the sum)

3 Tabulation of k-Parent Numbers

3.1 Brute-Forcing Preimages

Consider the equation $s(n) = m$, where the integer m is the *image* of n under the sum-of-proper-divisors function. Computing $s(n)$ is equivalent to the prime-factorization of n , using the method of [5]. A harder problem is to compute the *preimages* of m , all integers n such that $s(n) = m$ expressed as $s^{-1}(m)$. Restricting to even values of m this problem can be brute-forced by computing $s(n)$ for ranges of n , a conjecture similar to [7] is useful to select the values of n .

Proposition 3.1. *$s(n)$ is even if and only if*

$$n = (2m + 1)^2, \text{ or} \\ 2|n \text{ and } (n \neq m^2 \text{ or } n \neq 2m^2)$$

Conjecture 3.1. *If $2|m$ and $s(n) = m$ then $n \in \{x \mid \text{even } x, 2 \leq x \leq 2m\} \cup \{x^2 \mid \text{odd } x, 1 \leq x \leq m\}$*

- If $2|n$ then $s(n) = 1 + 2 + \frac{n}{2} + x$ so $s(n) > \frac{n}{2}$;
thus $s(n) > m$ if $n > 2m$.
- If $2 \nmid n$ then $s(n) = m$ if and only if n is an odd square,
thus $s(n) = 1 + \sqrt{n} + x$ and $s(n) > \sqrt{n}$. So $s(n) > m$ if $n > m^2$.

This approach is expensive if computing $s^{-1}(m)$ for a single value of m , most values will be disregarded when searching for $s(n) = m$. It is more palatable to compute preimages for a range of m , as determining $s^{-1}(m)$ requires $\{s^{-1}(x) \mid \text{even } x, 1 \leq x \leq m\}$ anyway. This computation is useful to count the occurrence and density of k -parent numbers, these counts can be compared to the predictions of the k -parent heuristic model to provide evidence for correctness.

3.2 Pomerance-Yang Algorithm

We can certainly do better than this brute-force approach for enumerating preimages, a technique of [10] is a useful improvement. Algorithm 1 presents the Pomerance-Yang algorithm modified to determine $\#s^{-1}(n)$ for all even integers n less than or equal to the bound x .

Algorithm 1 Pomerance-Yang

```
1: procedure ENUMERATE_KPARENT( $x$ )
2:    $f[i] \leftarrow 1, \forall i \in [1, x]$ 
3:   for all odd  $m \in [1, x]$  do
4:     if  $2 \mid \sigma(m)$  then
5:        $t \leftarrow 3\sigma(m) - 2m$ 
6:       while  $t \leq x$  do
7:          $f[t] \leftarrow f[t] + 1$ 
8:          $t \leftarrow 2t + \sigma(m)$ 
9:       end while
10:    end if
11:    if  $\sigma(m) = m + 1$  then  $\triangleright$  if  $m$  is prime
12:       $f[m + 1] \leftarrow f[m + 1] + 1$ 
13:    end if
14:  end for
15:  for all odd composite  $m \in [1, x^{2/3})$  do
16:    if  $s(m^2) \leq x$  then
17:       $f[s(m^2)] \leftarrow f[s(m^2)] + 1$ 
18:    end if
19:  end for
20:  return  $f$ 
21: end procedure
```

Algorithm 1 was computed on a large scale by [3] who used the result to compute the geometric mean of $s(n)/n$ weighted by $\#s^{-1}(n)$. To gather data on the density of k -parent numbers I have re-implemented this algorithm utilizing techniques that improve performance and data compression. These improvements have enabled the collection of non-aliquot counts up-to 2^{44} from the previous maximum result of 2^{40} while utilizing substantially less CPU time. In concrete terms [3] utilize 662 CPU days to compute $\#s^{-1}(n)$ for even $n \leq 2^{40}$, my implementation required 3.6 CPU days.

The performance improvement can largely be credited to avoiding any disc operations. The Pomerance-Yang algorithm requires $\sigma(m)$ for odd m less than or equal to the bound x , these values are computed using a sieving technique related to the well known Segmented Sieve of Eratosthenes. This input is computationally expensive, [3] choose to amortize this cost by storing the result to disc allowing reuse over multiple runs of the Pomerance-Yang algorithm. However in practice it appears more efficient to sieve ranges of $\sigma(m)$ and to immediately pass this array as input to the Pomerance-Yang algorithm. Suggesting that re-sieving $\sigma(m)$ is less expensive than loading the pre-computed result from disc to memory.

This variant of the Pomerance-Yang algorithm outputs a buffer with each index corresponding to an even integer n and with the value storing $\#s^{-1}(n)$. Like the input this buffer is managed completely in memory to avoid the overhead

of disc operations. One challenge of this approach is fitting the output buffer into the constrained size of memory, the simplest solution is to allocate a byte for each even n up-to the bound x . This requires at least $x/2$ bytes of memory, for instance at $x = 2^{40}$ the algorithm requires approximately 550GB which is well within the 3TB provided by the **bigmem** partition on uCalgary’s Advanced Research Computing (ARC) cluster. However allocating a byte for each n is excessive for most use cases, enumerating non-aliquots requires only a single bit for each n . A bit-array does not come naturally to modern low-level languages as the smallest addressable unit of memory is a byte, thus in C the following expression is true,

```
sizeof(uint8_t) == sizeof(bool).
```

The **PackedArray** data structure [8] provides an interface to manipulate an array with values stored in a user definable number of bits. Thus a 1-bit **PackedArray** can be defined which can be used to enumerate non-aliquots with a memory requirement of $x/16$, thus enumerating up-to 2^{44} requires 1.1TB rather than 8.8TB for a byte array.

Like the Pomerance-Yang implementation of [3] my implementation takes advantage of the natural parallelism of the algorithm, each iteration of the **for** loops on lines 3 and 15 of Algorithm 1 are totally independent. A challenge in my **PackedArray** implementation is managing data races caused multiple threads simultaneously incrementing values in the output buffer. In the byte array implementation this problem can be trivially solved by declaring updates to the output buffer as **atomic**. An array of mutex’s provides a performant mechanism to synchronize shared **PackedArray** access, each mutex guarding access to a range of indices in the output buffer. This approach yields significant performance improvements compared to using a single mutex to guard all updates to the output **PackedArray**.

$(P_k \cap [1, y])/y$	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
$y = 2^3$	0.250000	0.125000	0.250000	0.000000	0.000000	0.000000	0.000000	0.000000
$y = 2^4$	0.125000	0.187500	0.250000	0.000000	0.000000	0.000000	0.000000	0.000000
$y = 2^5$	0.062500	0.250000	0.218750	0.000000	0.000000	0.000000	0.000000	0.000000
$y = 2^6$	0.046875	0.234375	0.203125	0.031250	0.000000	0.000000	0.000000	0.000000
$y = 10^2$	0.050000	0.260000	0.160000	0.040000	0.000000	0.000000	0.000000	0.000000
$y = 2^7$	0.054688	0.257813	0.156250	0.031250	0.007813	0.000000	0.000000	0.000000
$y = 2^8$	0.062500	0.222656	0.171875	0.039063	0.007813	0.000000	0.000000	0.000000
$y = 2^9$	0.074219	0.214844	0.156250	0.042969	0.007813	0.005859	0.000000	0.000000
$y = 10^3$	0.089000	0.202000	0.148000	0.048000	0.009000	0.005000	0.000000	0.000000
$y = 2^{10}$	0.087891	0.203125	0.148438	0.047852	0.008789	0.004883	0.000000	0.000000
$y = 2^{11}$	0.098633	0.199219	0.137207	0.045898	0.011230	0.007813	0.000488	0.000000
$y = 2^{12}$	0.111572	0.188721	0.132568	0.045166	0.014160	0.005615	0.001465	0.000488
$y = 2^{13}$	0.117676	0.192017	0.120605	0.046631	0.014893	0.004883	0.001709	0.001221
$y = 10^4$	0.121200	0.186700	0.120800	0.047600	0.015000	0.005200	0.001700	0.001500
$y = 2^{14}$	0.125977	0.183838	0.119568	0.046448	0.015442	0.004883	0.001770	0.001526
$y = 2^{15}$	0.131927	0.183319	0.114044	0.045197	0.015900	0.005219	0.002197	0.001343
$y = 2^{16}$	0.136444	0.182129	0.110535	0.044846	0.015854	0.005508	0.002411	0.001297
$y = 10^5$	0.138630	0.182120	0.108380	0.044560	0.015700	0.005830	0.002420	0.001260
$y = 2^{17}$	0.140381	0.181831	0.106712	0.044388	0.015945	0.005852	0.002464	0.001282
$y = 2^{18}$	0.144272	0.180084	0.105057	0.043594	0.015995	0.005978	0.002430	0.001289
$y = 2^{19}$	0.147343	0.178972	0.103687	0.042801	0.015949	0.006029	0.002562	0.001266
$y = 10^6$	0.150232	0.177507	0.102694	0.042456	0.015772	0.006030	0.002555	0.001261
$y = 2^{20}$	0.150362	0.177533	0.102528	0.042461	0.015800	0.006013	0.002549	0.001256
$y = 2^{21}$	0.152802	0.176569	0.101274	0.042312	0.015703	0.005986	0.002563	0.001240
$y = 2^{22}$	0.155051	0.175726	0.100225	0.041888	0.015695	0.006006	0.002564	0.001269
$y = 2^{23}$	0.157040	0.174969	0.099278	0.041626	0.015638	0.006017	0.002573	0.001253
$y = 10^7$	0.157497	0.174789	0.099095	0.041556	0.015596	0.006027	0.002582	0.001249
$y = 2^{24}$	0.158811	0.174203	0.098593	0.041352	0.015574	0.006037	0.002567	0.001241
$y = 2^{25}$	0.160381	0.173577	0.097879	0.041162	0.015550	0.006035	0.002561	0.001229
$y = 2^{26}$	0.161788	0.173005	0.097286	0.040969	0.015539	0.006020	0.002546	0.001220
$y = 10^8$	0.162469	0.172762	0.097017	0.040867	0.015504	0.006005	0.002537	0.001214
$y = 2^{27}$	0.162956	0.172583	0.096816	0.040797	0.015496	0.005987	0.002532	0.001211
$y = 2^{28}$	0.164051	0.172156	0.096368	0.040672	0.015452	0.005971	0.002520	0.001197
$y = 2^{29}$	0.165032	0.171778	0.095983	0.040542	0.015425	0.005952	0.002503	0.001185
$y = 10^9$	0.165827	0.171478	0.095672	0.040444	0.015397	0.005935	0.002489	0.001172
$y = 2^{30}$	0.165912	0.171446	0.095640	0.040432	0.015395	0.005932	0.002487	0.001171
$y = 2^{31}$	0.166707	0.171148	0.095331	0.040343	0.015367	0.005910	0.002468	0.001158
$y = 2^{32}$	0.167413	0.170897	0.095064	0.040256	0.015340	0.005886	0.002450	0.001144
$y = 2^{33}$	0.168055	0.170667	0.094832	0.040182	0.015310	0.005861	0.002431	0.001130
$y = 10^{10}$	0.168187	0.170619	0.094786	0.040166	0.015303	0.005856	0.002427	0.001128
$y = 2^{34}$	0.168634	0.170465	0.094627	0.040114	0.015280	0.005838	0.002412	0.001117
$y = 2^{35}$	0.169161	0.170284	0.094447	0.040053	0.015252	0.005815	0.002393	0.001103
$y = 2^{36}$	0.169640	0.170125	0.094286	0.039999	0.015225	0.005791	0.002374	0.001090
$y = 10^{11}$	0.169881	0.170045	0.094206	0.039972	0.015210	0.005778	0.002364	0.001083
$y = 2^{37}$	0.170076	0.169982	0.094144	0.039949	0.015198	0.005767	0.002355	0.001077
$y = 2^{38}$	0.170474	0.169856	0.094018	0.039904	0.015172	0.005744	0.002337	0.001064
$y = 2^{39}$	0.170838	0.169744	0.093907	0.039863	0.015146	0.005721	0.002318	0.001051
$y = 10^{12}$	0.171129	0.169658	0.093821	0.039831	0.015125	0.005701	0.002303	0.001040
$y = 2^{40}$	0.171173	0.169645	0.093808	0.039826	0.015122	0.005698	0.002300	0.001038
$y = 2^{41}$	0.171480	0.169556	0.093719	0.039792	0.015099	0.005676	0.002282	0.001026
$y = 2^{42}$	0.171763	0.169478	0.093641	0.039761	0.015076	0.005654	0.002265	0.001014
$y = 2^{43}$	0.172023	0.169408	0.093571	-	-	-	-	-
$y = 2^{44}$	0.172264	-	-	-	-	-	-	-

Table 2: Densities of k -parent numbers up-to bound y , $P_k = \{n \mid s^{-1}(n) = k\}$.

A k-Parent Density Model Simplification

Prove that:

$$\begin{aligned}\mathbb{P}[k \text{ parents}] &= \lim_{x \rightarrow \infty} \binom{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)}}{k} \frac{1}{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot a}\right)^k} \left(1 - \frac{1}{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot a}\right)}\right)^{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} - k} \\ &= \frac{a^k}{k! \cdot s(a)^k} \cdot e^{-a/s(a)}\end{aligned}$$

Proof:

$$\begin{aligned}\mathbb{P}[k \text{ parents}] &= \lim_{x \rightarrow \infty} \binom{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)}}{k} \frac{1}{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot a}\right)^k} \left(1 - \frac{1}{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot a}\right)}\right)^{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} - k} \\ &= \lim_{x \rightarrow \infty} \binom{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)}}{k} \left(\frac{A_y \cdot a}{\phi(A_y) \cdot x}\right)^k \cdot \lim_{x \rightarrow \infty} \left(1 - \frac{A_y \cdot a}{\phi(A_y) \cdot x}\right)^{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} - k} \\ &= \lim_{x \rightarrow \infty} \mathbb{A} \cdot \lim_{x \rightarrow \infty} \mathbb{B}\end{aligned}$$

Part A:

$$\begin{aligned}
\lim_{x \rightarrow \infty} \mathbb{A} &= \lim_{x \rightarrow \infty} \left(\frac{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)}}{k} \right) \left(\frac{A_y \cdot a}{\phi(A_y) \cdot x} \right)^k \\
&= \lim_{x \rightarrow \infty} \frac{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} \right)!}{k! \left(\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} - k \right)!} \left(\frac{A_y \cdot a}{\phi(A_y) \cdot x} \right)^k && \text{(Def. Binomial Coefficient)} \\
&= \frac{1}{k!} \lim_{x \rightarrow \infty} \frac{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} \right)!}{\left(\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} - k \right)!} \left(\frac{A_y \cdot a}{\phi(A_y) \cdot x} \right)^k && \text{(Constant coefficient limit law)} \\
&= \frac{1}{k!} \lim_{x \rightarrow \infty} \prod_{i=0}^{k-1} \left[\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} - i \right] \left(\frac{A_y \cdot a}{\phi(A_y) \cdot x} \right)^k && \text{(See note [1])} \\
&= \frac{1}{k!} \lim_{x \rightarrow \infty} \prod_{i=0}^{k-1} \left[\left(\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} - i \right) \left(\frac{A_y \cdot a}{\phi(A_y) \cdot x} \right) \right] && \text{(See note [2])} \\
&= \frac{1}{k!} \lim_{x \rightarrow \infty} \prod_{i=0}^{k-1} \left[\frac{\phi(A_y) \cdot A_y \cdot x \cdot a}{\phi(A_y) \cdot A_y \cdot x \cdot s(a)} - \frac{i \cdot A_y \cdot a}{\phi(A_y) \cdot x} \right] \\
&= \frac{1}{k!} \prod_{i=0}^{k-1} \left[\lim_{x \rightarrow \infty} \left(\frac{a}{s(a)} - \frac{i \cdot A_y \cdot a}{\phi(A_y) \cdot x} \right) \right] && \text{(Distribution over product limit law)} \\
&= \frac{1}{k!} \prod_{i=0}^{k-1} \left[\lim_{x \rightarrow \infty} \left(\frac{a}{s(a)} \right) - \lim_{x \rightarrow \infty} \left(\frac{i \cdot A_y \cdot a}{\phi(A_y) \cdot x} \right) \right] && \text{(Distribution over difference limit law)} \\
&= \frac{1}{k!} \prod_{i=0}^{k-1} \left[\frac{a}{s(a)} \right] \\
&= \frac{a^k}{k! \cdot s(a)^k} && \left(\prod_{i=0}^{k-1} c = c^k \right)
\end{aligned}$$

Part B:

Note: simplification checked using Wolfram-Alpha.

$$\begin{aligned}
\lim_{x \rightarrow \infty} \mathbb{B} &= \lim_{x \rightarrow \infty} \left(1 - \frac{A_y \cdot a}{\phi(A_y) \cdot x} \right)^{\frac{\phi(A_y) \cdot x}{A_y \cdot s(a)} - k} \\
&= e^{-a/s(a)}
\end{aligned}$$

Note [1]

$$\begin{aligned}
\frac{x!}{(x-k)!} &= \frac{x(x-1)(x-2) \dots (x-(k-1))(x-k)!}{(x-k)!} && \text{(Def. factorial)} \\
&= x(x-1)(x-2) \dots (x-(k-1)) \\
&= \prod_{i=0}^{k-1} (x-i)
\end{aligned}$$

Note [2]

$$\begin{aligned}
\prod_{i=0}^{k-1} (x-i) \cdot y^k &= \overbrace{(x)(x-1)(x-2) \dots (x-k)}^{k \text{ terms}} \cdot y^k \\
&= y(x) \cdot y(x-1) \cdot y(x-2) \dots y(x-k) \\
&= \prod_{i=0}^{k-1} y(x-i)
\end{aligned}$$

References

- [1] W. Bosma and B. Kane, *The aliquot constant*, The Quarterly Journal of Mathematics **63** (2010), no. 2, 309–323.
- [2] E. Catalan, *Propositions et questions diverses*, Bulletin de la Société Mathématique de France **2** (1888), 128–129.
- [3] K. Chum, R. K. Guy, Jr. M. J. Jacobson, and A. S. Mosunov, *Numerical and statistical analysis of aliquot sequences*, Experimental Mathematics **29** (2018), no. 4, 414–425.
- [4] Matthew Conroy, *Examples of integer sequence density calculations*.
- [5] Dennis DeTurck, *Finding the sum of the divisors of n*, 2005, <https://www2.math.upenn.edu/~deturck/m170/wk3/lecture/sumdiv.html>.
- [6] Richard K. Guy and J. L. Selfridge, *What drives an aliquot sequence?*, Mathematics of Computation **29** (1975), no. 129, 101–101.
- [7] Florian Luca and Carl Pomerance, *The range of the sum-of-proper-divisors function*, Acta Arithmetica **168** (2015), 187–199.
- [8] Gregory Pakosz, *Packedarray*, Dec 30 2015, [Online; accessed 2022-4-6].

- [9] Paul Pollack and Carl Pomerance, *Some problems of erdős on the sum-of-divisors function*, Transactions of the American Mathematical Society, Series B **3** (2016), 1–26.
- [10] Carl Pomerance and Hee-Sung Yang, *Variant of a theorem of erdős on the sum-of-proper-divisors function.*, Mathematics of Computation **83** (2014), no. 288, 1903–1913.
- [11] Neil Sloane, *A005114*, 2022.
- [12] Paul Zimmermann, *Aliquot sequences 276, 552, 564, 660, 996, 1074 and 1134*, http://christophe.clavier.free.fr/Aliquot/site/zimmermann_table.html, Aug 2016.