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

Gavin Guinn
Supervisor: Michael J. Jacobson

December 2021

1 Introduction

1.1 Sum-of-Proper-Divisors Function

1.2 Aliquot Sequences

1.3 Non-Aliquots and k-Parent Numbers

2 Modeling the Density of k-Parent Numbers

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 [Sloane, 2022], if $n \in N_a$ then $\nexists m$ s.t. $s(m) = n$ or $\#s^{-1}(n) = 0$. 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, where

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

in other words determining the proportion of numbers that are non-aliquot. Using probabilistic methods [Pollack and Pomerance, 2016] precisely model the density of non-aliquot numbers.

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

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

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

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

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 [DeTurck, 2005]. 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 [Luca and Pomerance, 2015] is useful to select the values of n.

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

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 $n > m^2$.

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

$n = (2m + 1)^2$, or

$2|n$ and ($n \neq m^2$ or $n \neq 2m^2$).

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 $\{s^{-1}(x)|\ \text{even}\ x,\ 1 \leq x \leq m\}$ 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 [Pomerance and Yang, 2014] is a useful improvement. Algorithm 1 presents the Pomerance-Yang algorithm modified to determine $\#s^{-1}(n)$ for all integers less than or equal to the bound x.

Algorithm 1 Pomerance-Yang

```
procedure ENUMERATE_kPARENT(x)
    f[i] ← 1, ∀i ∈ [1, x]
    for all odd m ∈ [1, x] do
        if 2|σ(m) then
            t ← 3σ(m) − 2m
            while t ≤ x do
                f[t] ← f[t] + 1
                t ← 2t + σ(m)
            end while
        end if
        if σ(m) = m + 1 then                    ▷ if m is prime
            f[m + 1] ← f[m + 1] + 1
        end if
    end for
    for all odd composite m ∈ [1, x^(2/3)) do
        if s(m^2) ≤ x then
            f[s(m^2)] ← f[s(m^2)] + 1
        end if
    end for
    return f
end procedure
```

Algorithm 1 was computed on a large scale by [Chum et al., 2018] who used the result to compute the geometric mean of $s(n)/n$ weighted by $\#s^{-1}(n)$. Extending this work to compute the density of k-parent numbers I have re-implemented the algorithm and can report a substantial performance improvement.

References

[Chum et al., 2018] Chum, K., Guy, R. K., Jacobson, J. M. J., and Mosunov, A. S. (2018). Numerical and statistical analysis of aliquot sequences. *Experimental Mathematics*, 29(4):414–425.

[DeTurck, 2005] DeTurck, D. (2005). Finding the sum of the divisors of n. https://www2.math.upenn.edu/~deturck/m170/wk3/lecture/sumdiv.html.

[Luca and Pomerance, 2015] Luca, F. and Pomerance, C. (2015). The range of the sum-of-proper-divisors function. *Acta Arithmetica*, 168:187–199.

[Pollack and Pomerance, 2016] Pollack, P. and Pomerance, C. (2016). Some problems of erdős on the sum-of-divisors function. *Transactions of the American Mathematical Society, Series B*, 3:1–26.

[Pomerance and Yang, 2014] Pomerance, C. and Yang, H.-S. (2014). Variant of a theorem of erdös on the sum-of-proper-divisors function. *Mathematics of Computation*, 83(288):1903–1913.

[Sloane, 2022] Sloane, N. (2022). A005114.