A Gleeful Algorithm

Efficiently generating sums of consecutive primes

Eleanor Waiss

Department of Mathematical Sciences '25
Butler University

11 April 2025
Butler Undergraduate Research Conference

Background

Let p_i denote the i-th prime number.

Definition

A *gleeful number* is a number g that can be written as a sum of consecutive primes:

$$g = p_i + p_{i+1} + \cdots + p_{i+\ell} = \sum_{k=i}^{i+\ell} p_i.$$

$$17 = 2 + 3 + 5 + 7$$

$$2357 = 773 + 787 + 797 = 461 + 463 + 467 + 487$$

Each unique way to express a gleeful number is a *representation*. The *length* of a representation is the value ℓ.

Moser's *Notes*

Theorem (Moser, 1963)

Let f count the representations of a gleeful number g. Then

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} f(i)}{n} = \log 2 \approx 0.6931.$$

Moser posed four open problems that arise with this result:

1. Is $f(n) = 1$ infinitely often?
2. Is $f(n) = k$ solvable for every integer $k \geq 0$?
3. Does the set of numbers n such that $f(n) = k$ have positive density for every integer $k \geq 0$?
4. Is $\limsup_{n\to\infty} f(n) = \infty$?

Our Approach

Get empirical data to non-rigourously answer Moser's questions by way of frequency table:

- Fix some upper bound n;
- Explicitly construct every representation possible for $g \leq n$;
- Summarize frequencies—do they follow some statistical distribution?

A Naïve Approach

1. Construct an array of all prime numbers $\leq n$:

2	3	5	7	11	13	17	$\cdots$

2. Construct the prefix sums S_i of primes:

0	2	5	10	17	28	41	$\cdots$

3. Consider all differences of pairs $S_j - S_i$, $0 \leq i < j \leq \pi(n)$

$$2, \quad 5, 3, \quad 10, 8, 5, \quad 17, 15, 12, 7, \quad 28, \ldots$$

4. **Sort the output** and count frequency

Takes $O(n^2/\log^2 n)$ time (step 3) and $O(n)$ space (step 4).

Considerations

- The best sorting algorithms require $O(n \log n)$ time and $O(n)$ space

  $\Rightarrow$ Generate representations of g in-order and count as we go.

- The *length* ℓ of a representation g gives a good estimate for the size of primes needed: g/ℓ; but g's representations can come in any length:
  - Constructing all primes at the start is time-optimal but takes $O(n/\log n)$ space;
  - Constructing primes "on the fly" is space-optimal but slow.

  $\Rightarrow$ Use different behavior depending on what kind of representation we're counting.

A New Algorithm

Give each possible length an object instance — each object contains information about primes contained in the summand.

1. Initialize each object with the gleeful representation starting at $p_1 = 2$.
2. Insert all objects into a priority queue based on the value g.
3. Iteratively dequeue each object, increment the histogram for g, update the object value, then queue the object back into the priority queue.

$$\begin{aligned} 2 + 3 + 5 + 7 \quad &= 17 \\ 3 + 5 + 7 + 11 &= 26 \end{aligned}$$

Theorem (Sorenson-W.,'25)

The above algorithm takes $O(x \log x)$ arithmetic operations and $x^{3/5+o(1)}$ space to compute the histogram h up to $x > 0$.

Tabulating Gleefuls up to 25

Timestep	1	2	3	4	g	$f(g)$
1	2	5	10	17	2	1
2	3	5	10	17	3	1
3	5	5	10	17	5	2
4	7	8	10	17	7	1
5	11	8	10	17	8	1
6	11	12	10	17	10	1
7	11	12	15	17	11	1
8	13	12	15	17	12	1
9	13	18	15	17	13	1
10	17	18	15	17	15	1
11	17	18	23	17	17	2
12	19	18	23	26	18	1
13	19	24	23		19	1
14	23	24	23		23	2
15	29	24	31		24	1

A Fishy Histogram for $n = 10^{12}$

Count	Observed
0	502159842109
1	347327858123
2	118662285846
3	26721935372
4	4465680602
5	591227093
6	64644512
7	6004622
8	484875
9	34610
10	2108
11	124
12	4

Count	$X \sim \mathrm{Pois}(\lambda = \log 2)$
0	500000000000
1	346573590279
2	120113253479
3	27752054332
4	4809064553
5	666677907
6	77017651
7	7626366
8	660774
9	50890
10	3527
11	222
12	12

Back to Moser

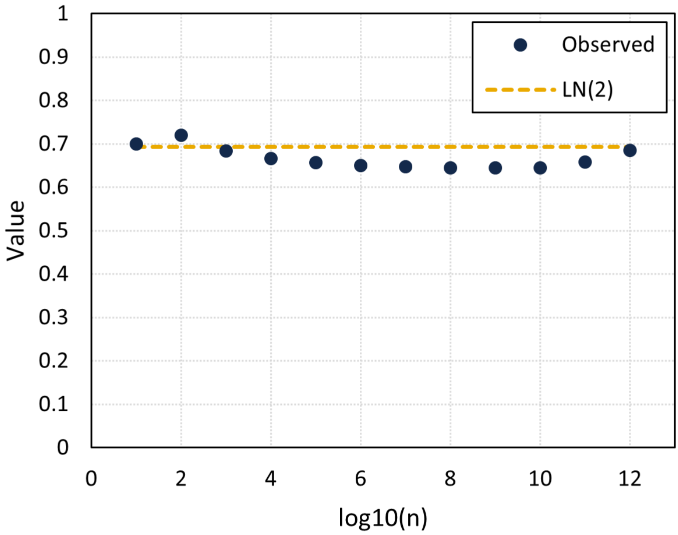
In the spirit of Cramér's model for the distribution for primes:

1. Is $f(n) = 1$ infinitely often?
   Yes,
2. Is $f(n) = k$ solvable for every integer $k \geq 0$?
   Yes,
3. Does the set of numbers n such that $f(n) = k$ have positive density for every integer $k \geq 0$?
   Yes, with density

$$\mathrm{Leb}_{\mathbb{N}}\left(f^{-1}(k)\right) = \frac{(\log 2)^k}{2 \cdot k!},$$

4. Is $\limsup_{n\to\infty} f(n) = \infty$?
   Yes

Average Representations



Future Work

$f(n)$	Min n
1	2
2	5
3	41
4	1151
5	311
6	34421
7	218918
8	3634531
9	48205429
10	1798467197
11	12941709050
12	166400805323
13	6123584726269

Table: OEIS A054859. $f(n) = 13$ from G. Resta (2020)

THANK YOU!

Eleanor Waiss

ewaiss@butler.edu

A Gleeful Algorithm

Butler University

This work is supported in part by a grant from the Holcomb Awards Committee.