

A Gleeful Algorithm

Algorithms to Write Integers as the Sum of Consecutive Primes

Eleanor Waiss¹, joint with Jon Sorenson²

¹Department of Mathematical Sciences
Butler University

²Department of Computer Science and Software Engineering
Butler University

INTEGERS 2025

Athens, GA USA

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:

$$g = p_i + p_{i+1} + \cdots + p_{i+(\ell-1)} = \sum_{k=0}^{\ell-1} p_{i+k}.$$

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

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

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:

$$g = p_i + p_{i+1} + \cdots + p_{i+(\ell-1)} = \sum_{k=0}^{\ell-1} p_{i+k}.$$

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

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

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

Theorem (Moser, 1963)

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

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

Theorem (Moser, 1963)

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

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

Since the average value of $f(n)$ is $\log 2$ it follows that $f(n) = 0$ infinitely often. The following problems, among others, suggest themselves:

1. Is $f(n) = 1$ infinitely often?
2. Is $f(n) = k$ solvable for every k ?
3. Do the numbers n for which $f(n) = k$ have a density for every k ?
4. Is $\overline{\lim} f(n) = \infty$?

Figure: Excerpt from Moser's 1963 paper; also found in Guy's *Unsolved Problems in Number Theory*.

Our Task

Get empirical data to investigate Moser's questions by way of frequency table:

- Fix some upper bound n ;
- Explicitly construct every representation possible for all $g \leq n$;
- Summarize $\#\{f^{-1}(k)\}$ — do they follow some known statistical distribution?

A Naïve Approach

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

2	3	5	7	11	13	17	...
---	---	---	---	----	----	----	-----

- ② Construct the prefix sums S_i of primes:

0	2	5	10	17	28	41	...
---	---	---	----	----	----	----	-----

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

2, 5, 3, 10, 8, 5, 17, 15, 12, 7, 28, ...

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

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

Considerations

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

→ Generate representations of g in-order, counting as we go.

- 2 The *length* ℓ of a representation g gives a good estimate for the size of primes needed: g/ℓ ;
 - Constructing all primes at the start is time-optimal but takes $O(n/\log n)$ word space;
 - Constructing primes “on the fly” is space-optimal but slow.

→ Differentiate behavior based on the value ℓ .

A New Algorithm

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

- 1 Initialize each object with the appropriate gleeful representation starting at $p_1 = 2$.
- 2 Enqueue all objects into a priority queue based on the value g .
- 3 Iteratively dequeue each object, increment the histogram for $f(g)$, update the object value g , and enqueue.

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

$$3 + 5 + 7 + 11 = 26$$

Theorem (Sorenson-W., '25)

The above algorithm takes $O(n \log n)$ arithmetic operations and $n^{3/5+o(1)}$ space to compute the histogram up to $n \in \mathbb{N}$.

Tabulating Gleefuls up to $n = 25$

Timestep	1	2	3	4			g	$f(g)$
1	②	5	10	17			2	1

Tabulating Gleefuls up to $n = 25$

Timestep	1	2	3	4	g	$f(g)$
1	②	5	10	17	2	1
2	③	5	10	17	3	1

Tabulating Gleefuls up to $n = 25$

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

Tabulating Gleefuls up to $n = 25$

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

Tabulating Gleefuls up to $n = 25$

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	⑳	24	㉓		23	2
15	29	㉔	31		24	1

A Fishy Histogram for $n = 10^{14}$

Count	Observed
0	52 255 406 573 294
1	33 983 734 548 972
2	10 980 796 355 393
3	2 351 331 657 326
4	375 496 312 243
5	47 717 060 499
6	5 027 735 200
7	451 927 961
8	35 376 934
9	2 452 073
10	151 480
11	8 546
12	430
13	14
14	1

A Fishy Histogram for $n = 10^{14}$

Count	Observed	Count	$X \sim \text{Pois}(\lambda = \log 2)$
0	52 255 406 573 294	0	50 000 000 000 000
1	33 983 734 548 972	1	34 657 359 027 997
2	10 980 796 355 393	2	12 011 325 347 955
3	2 351 331 657 326	3	2 775 205 433 241
4	375 496 312 243	4	480 906 455 381
5	47 717 060 499	5	66 667 790 732
6	5 027 735 200	6	7 701 765 197
7	451 927 961	7	762 636 690
8	35 376 934	8	66 077 434
9	2 452 073	9	5 089 043
10	151 480	10	352 746
11	8 546	11	22 228
12	430	12	1 284
13	14	13	68
14	1	14	3

(Heuristically) 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

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

- ④ Is $\limsup_{n \rightarrow \infty} f(n) = \infty$?

Yes

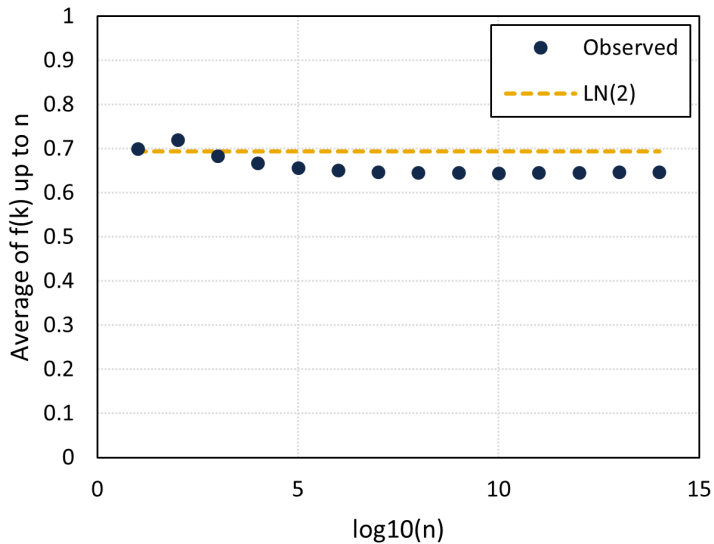


Figure: Empirical values for the average of f .

Computations

k	$\min\{f^{-1}(k)\}$
1	2
2	5
3	41
4	1 151
5	311
6	34 421
7	218 918
8	3 634 531
9	48 205 429
10	1 798 467 197
11	12 941 709 050
12	166 400 805 323
13	6 123 584 726 269
14	84 941 668 414 584

Table: OEIS A054859.

$f^{-1}(14)$ from Sorenson-W.
(2025). Primes are in **bold**.

Computations

k	$\min\{f^{-1}(k)\}$
1	2
2	5
3	41
4	1 151
5	311
6	34 421
7	218 918
8	3 634 531
9	48 205 429
10	1 798 467 197
11	12 941 709 050
12	166 400 805 323
13	6 123 584 726 269
14	84 941 668 414 584

Table: OEIS A054859.
 $f^{-1}(14)$ from Sorenson-W.
 (2025). Primes are in **bold**.

ℓ	p_{\min}	p_{\max}
2 117 074	21 797 833	58 785 359
361 092	231 753 581	238 710 779
288 268	291 853 531	297 473 801
199 390	424 030 259	427 989 799
112 544	753 590 641	755 886 067
73 026	1 162 407 049	1 163 930 791
68 854	1 232 927 929	1 234 369 457
296	286 965 092 209	286 965 099 727
294	288 917 235 553	288 917 243 497
206	412 338 193 609	412 338 198 731
146	581 792 247 697	581 792 251 207
86	987 693 817 667	987 693 819 859
26	3 266 987 246 389	3 266 987 247 019
2	42 470 834 207 273	42 470 834 207 311

Table: The 14 representations for
 $g = 84\,941\,668\,414\,584$.

- Ran computation (up to $n = 10^{12}$) on one thread of an Intel(R) Xeon(R) CPU E5-2650 v4 at 2.20GHz, taking ≈ 26 days.
- Some improvements possible on this single-node approach to lower storage requirements, could in-theory run for several years up to 10^{17} or 10^{18} .
- Comments on OEIS A054859 gave outline of a “window” algorithm that can be parallelized — this approach took ≈ 7 days in parallel to 10^{14} .

THANK YOU!

Eleanor Waiss

`ewaiss@butler.edu`

A Gleeeful Algorithm

Butler University

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

A very happy birthday to Mel and Carl!