Integer Complexity and Mixed Binary-Ternary Representation

Kazuyuki Amano
Gunma University, Kiryu, Japan

Abstract

The integer complexity of a natural number n, denoted by $\|n\|$, is the smallest number of 1's needed to express n using an arbitrary combination of addition and multiplication (and parentheses). For example, $\|6\| = 5$ since the expression $6 = (1+1) \cdot (1+1+1)$ contains five 1's and there are no such expressions containing at most four 1's. The investigation of this cute complexity measure was originated by Mahler and Popken in the 1950s. It is easy to see that $\|n\|/\log_3 n \in [3, 3\log_2 3]$ ($\sim [3, 4.755]$) for every n, but the distribution of $\|n\|$ is largely unknown.

In this work, we focus on the restricted expressions obtained by applying Horner's schema to a *mixed binary-ternary representation* of a given number in which we can arrange base-two and base-three digits in an arbitrary order. Let $f(n)$ denote the minimum number of 1's needed to express n in this way. Apparently, $f(n) \geq \|n\|$ for every n. We extensively investigate on $f(n)$ via the combination of computer experiments and theoretical analysis and obtain the following set of results: (i) Computer experiments supporting the hypothesis that $f(n)/\log_3 n < 3.483$ on average and $f(n)/\log_3 n < 4.212$ for all n, (ii) For almost all natural numbers n, $3.120 < f(n)/\log_3 n < 3.587$, and (iii) There are infinitely many n's such that $f(n)/\log_3 n > 3.934$. Several new bounds on the original integer complexity are also presented in the paper.

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics

Keywords and phrases Integer complexity, Lower bounds, Upper bounds, Horner's schema, Computer assisted proof

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2022.29

Supplementary Material *Dataset*: `https://gitlab.com/KazAmano/integer-complexity` archived at `swh:1:dir:e76eb9f674cdd675ba9a9aaf112f11934e7d23f6`

Funding Supported in part by JSPS Kakenhi No. JP21K19758, JP18K11152 and JP18H04090.

Acknowledgements The author would like to thank the anonymous reviewers of ISAAC '22 for their many helpful comments and suggestions.

1 Introduction

The *integer complexity* of a natural number n, denoted by $\|n\|$, is the smallest number of 1's needed to express n using an arbitrary combination of addition and multiplication (and parentheses). For instance, $\|6\| = 5$ since $6 = (1+1) \cdot (1+1+1)$ is a shortest expression (in terms of the number of 1's) of 6 that contains five 1's. Similarly, $\|11\| = 8$ by the witness

$$\begin{aligned} 11 &= (1+1+1) \cdot (1+1+1) + 1 + 1 \ (= 3 \cdot 3 + 2) \\ &= (1+1)(1+1+1+1+1) + 1 = (1+1)((1+1)(1+1)+1) + 1 \ (= 2 \cdot 5 + 1). \end{aligned}$$

The problem is to find a behavior of $\|n\|$ as well as an optimal expression of n. This cute problem was originated by Mahler and Popken [13] in the 1950s and was later popularized by Guy [9, 10].

Since the last operation is either addition or multiplication, the value of $\|n\|$ is given by a simple recursion. Namely, $\|1\| = 1$ and, for $n > 1$,

$$\|n\| = \min_{\substack{a,b<n\in\mathbb{N} \\ ab=n \text{ or } a+b=n}} \{\|a\| + \|b\|\}. \quad (1)$$

In spite of its simplicity, the analysis of $\|n\|$ seems extremely hard.

It is not hard to see that

$$3\log_3 n \le \|n\| \le 3\log_2 n \sim 4.755\log_3 n. \tag{2}$$

See e.g., [9]. The lower bound is attained by $n = 3^k$ whose shortest expression is

$$3^k = (1+1+1)\cdot(1+1+1)\cdot\cdots\cdot(1+1+1) \quad (k \text{ times}).$$

The optimality of this expression can easily be verified by noticing that $\log_3 m/m$ takes its maximum at $m = 3$ when m is a positive integer. The upper bound in Eq. (2) is obtained by applying Horner's method to the binary representation of n [9]. Namely, for a $(k+1)$-bit binary number $n = \sum_{i=0}^{k} d_i 2^i$, we can express n by

$$n = d_0 + (1+1)(d_1 + (1+1)(d_2 + (\cdots(1+1)(d_{k-1} + 1 + 1)\cdots))). \tag{3}$$

This shows that every $(k+1)$-bit binary number has integer complexity at most $2k$ plus one less than the number of 1's in the binary representation of n. This implies $\|n\| \le 2k + \sum_{i=0}^{k-1} d_i \le 3k$, which gives the upper bound in Eq. (2).

Despite a long line of research (e.g., [1, 2, 3, 4, 5, 6, 7, 11, 14, 18]), many fundamental problems remain open. For example, we cannot answer the question asked by Selfridge (see [9]) over three decades ago that

Is there a number a for which $\|2^a\| < 2a$?

For a notational simplicity, we write $\|n\|_{\log} := \|n\|/\log_3 n$ (following [11]). We know from Eq. (2) that $\|n\|_{\log} \in [3, 3\log_2 3]$ ($\sim [3, 4.755]$) for every n. However, the distribution of $\|n\|_{\log}$ is largely unknown. We do not know whether there are infinitely many n's such that $\|n\|_{\log} \ge (3+\epsilon)$ for some $\epsilon > 0$. Also, to the best of our knowledge, a general upper bound of the form $\|n\|_{\log} \le (3\log_2 3 - \epsilon)$ for some $\epsilon > 0$ has not been published yet.

Since the best known upper bound is obtained via the binary representation and the best known lower bound is obtained via the ternary representation, it would be interesting to investigate the integer complexity via the *mixed binary-ternary representation*. This is the main focus of this work.

The mixed binary-ternary representation is a number system in which we can arrange the base-two and base-three digits in an arbitrary order. Such a system has been considered before e.g., in [19] to pursue a new way of attack to the notorious Collatz problem.

When n is represented by this system using k digits where the i-th base is $b_i \in \{2, 3\}$ and the i-th digit is $d_i \in \{0, 1, \ldots, b_i - 1\}$ (here the least significant digit is counted as zero-th), then by applying Horner's schema, we have the expression

$$n = d_0 + b_0(d_1 + b_1(d_2 + (\cdots b_{k-3}(d_{k-2} + b_{k-2}(d_{k-1})\cdots))), \tag{4}$$

which contains $\sum_{i=0}^{k-2}(d_i + b_i)$ plus d_{k-1} 1's, where the last d_{k-1} can be discarded if $d_{k-1} = 1$.

Let $f(n)$ denote the minimum number of 1's needed to express n in this way. Obviously, $f(n) \ge \|n\|$ for every natural number n. In fact, a quick inspection shows that $f(n) = \|n\|$ for every $n \le 34$ and the first discrepancy occurs at $n = 35$ as $12 = f(35) \gneq \|35\| = 11$. See the end of Section 2 for some more experimental data on the comparison between $f(n)$ and $\|n\|$.

The value of $f(n)$ is given by the following recursive formula.[1]

$$f(n) := \begin{cases} \min\{f(\lfloor n/2 \rfloor) + 2 + (n \bmod 2),\ f(\lfloor n/3 \rfloor) + 3 + (n \bmod 3)\} & (n \geq 6), \\ n & (n \leq 5). \end{cases} \quad (5)$$

In what follows, we write $f(n)/\log_3 n$ as $f(n)_{\log}$. Apparently, $f(n)_{\log} \in [3, 4.755]$ and the lower limit is sharp since $f(3^k) = 3k$. At a first glance, we have a hope that it is not hard to evaluate $\limsup_{n\to\infty} f(n)_{\log}$ whose value is expected to be well below 4.755. However, we could not find a way to prove even a slightly better than a trivial bound such as $f(n)_{\log} < (3\log_2 3 - \epsilon)$ for some $\epsilon > 0$.

In this work, we extensively investigate on the behavior of $f(n)$ through the combination of computer experiments and theoretical analysis and obtain the following set of results.

1. Computer experiments support the hypothesis that $f(n)_{\log} < 3.483$ on average and $f(n)_{\log} < 4.212$ for all n.
2. For almost all natural numbers n, $3.120 < f(n)_{\log} < 3.587$.
3. There are infinitely many n's such that $f(n)_{\log} > 3.934$.

Note that the lower bound in #2 shows that the mixed binary-ternary representation is not strong enough for proving an upper bound of the form $\|n\|_{\log} \leq (3 + \epsilon)$.

In addition, we give the following "new records" on the original integer complexity by extending the computational efforts along the line suggested by the previous work [7, 15].

1. For a set of numbers of natural density one, $\|n\|_{\log} < 3.556$. This improves the bound of $\|n\|_{\log} < 3.620$ due to Cordwell et al. [5].
2. For a set of numbers of logarithmic density one, $\|n\|_{\log} < 3.472$. This improves the bound of $\|n\|_{\log} < 3.520$ due to David Bevan (described in [15]).

The rest of the paper is organized as follows. In Section 2, we formally introduce the mixed binary-ternary representation and discuss its connection to the integer complexity. In Section 3, we show the results of our computational experiments as well as some conjectures inspired by them. In Section 4, we show the theoretical results on the upper and lower bounds on $f(n)$. In Section 5, we describe an improvement to the upper bound on $\|n\|$. Finally, we make some discussions to close the paper in Section 6.

2 Mixed Binary-Ternary Representation

For natural numbers $n, m \in \mathbb{N}$, $[n]$ denotes the set $\{1, 2 \ldots, n\}$ and $[m, n]$ denotes the set $\{m, m+1, \ldots, n\}$.

Throughout this work, we consider a number system with mixed bases of two and three. The *mixed binary-ternary representation* of a natural number is consisting of two arrays of integers Digit and Base. For $i \geq 0$, $\mathsf{Base}[i] \in \{2, 3\}$ represents the base of the i-th digit (here the least significant digit is counted as zero-th) and $\mathsf{Digit}[i] \in \{0, 1, \ldots, \mathsf{Base}[i] - 1\}$ represents the i-th digit.

Given Digit and Base of length k, a number n is represented as

$$n = \sum_{i=0}^{k-1} \mathsf{Digit}[i] \prod_{j=0}^{i-1} \mathsf{Base}[j]. \quad (6)$$

[1] In Eq. (5), we apply the first case only when $n \geq 6$. Actually, the values of $f(n)$ will not be changed if we replace $n \geq 6$ by $n \geq 2$. The reason that we did not do so is as follows. The second term in the minimum should give a complexity corresponding to a representation such that the lowest digit is base-three. However, e.g., when $n = 5$, it wrongly gives $f(1) + 3 + 2 = 6$ because it ignores the fact that we can omit the leading 1 to express n.

For notational simplicity, we may write this as $n = (d_{k-1})_{b_{k-1}} \cdots (d_1)_{b_1}(d_0)_{b_0}$ for $d_i = \mathsf{Digit}[i]$ and $b_i = \mathsf{Base}[i]$. Note that the representation is not unique. For example, $25 = 1_2 1_2 0_2 0_2 1_2 = 1_2 1_3 0_2 1_3 = 2_3 0_2 0_3 1_2 = 2_3 2_3 1_3$ (and more).

Given a natural number n in the mixed binary-ternary representation, we can express n using $\{1, +, \cdot\}$ (and parentheses) as

$$n = d_0 + b_0(d_1 + b_1(d_2 + (\cdots b_{k-3}(d_{k-2} + b_{k-2}(d_{k-1})\cdots))),$$

where 2 is replaced by $(1+1)$ and 3 is replaced by $(1+1+1)$. This expression contains $\sum_{i=0}^{k-2}(d_i + b_i)$ plus d_{k-1} 1's, where the last d_{k-1} can be discarded if $d_{k-1} = 1$.

For a natural number $n \in \mathbb{N}$, let $f(n)$ denote the minimum number of 1's in such an expression, where the minimum is taken over all orderings of the base-two and base-three digits.

Obviously, $f(n) \geq \|n\|$ for every $n \in \mathbb{N}$.

As was described in Introduction, $f(n)$ is given by the following recursive formula:

$$f(n) := \begin{cases} \min\{f(\lfloor n/2 \rfloor) + 2 + (n \bmod 2),\ f(\lfloor n/3 \rfloor) + 3 + (n \bmod 3)\} & (n \geq 6), \\ n & (n \leq 5). \end{cases}$$

A quick computer calculation shows that $f(n) = \|n\|$ for every $n \leq 34$. The first discrepancy occurs at $n = 35$ as $12 = f(35) \gneq \|35\| = 11$. For $n \leq 100$, there are only four numbers $n \in \{35, 70, 71, 95\}$ satisfying $f(n) \gneq \|n\|$. As expected, our computer experiment suggests that the fraction of such numbers increases as n increases. For example, for $n \leq 10^6$, 652,037 numbers satisfy $f(n) \gneq \|n\|$.

3 Experimental Results

In this section, we describe our experimental results of $f(n)$ as well as some conjectures inspired by them. If we have enough memory to store all $f(n)$'s, then we can obtain the table of $f(n)$ in linear time by dynamic programming based on Eq. (5). If not, we can calculate $f(n)$ by expanding the recursion given in Eq. (5). This is time-consuming when n is large, but uses less memory. We can also combine these two methods in a hybrid manner.

In this work, we calculated $f(n)$ for $n \leq 2^{42}$ ($\sim 4.4 \cdot 10^{12}$) using a computer. All our experiments are conducted on an Ubuntu server with 64GB memory and Ryzen 3960x CPU (24 cores, 3.8GHz). The computation of $f(n)$ took about one week using a single core of the aforementioned machine.

3.1 Worst Case Bounds

For $k \geq 1$, let $e_f(k)$ denote the smallest integer n such that $f(n) \geq k$. Our computation is enough to determine the values of $e_f(k)$ for $k \leq 110$. Due to the space constraint, we only show the values $e_f(k)$ for $90 \leq k \leq 110$ in Table 1. See Appendix A for the full list.

It is interesting to see that $e_f(k) + 1$ (or $e_f(k) + 2$ for some cases) tends to be factored into the product of small primes, although there are some exceptions.

We also show a graph that plots $k/\log_3 e_f(k)$. See Fig. 1. The graph suggests that $k/\log_3 e_f(k)$ tends to around 4.2 as k tends to ∞. In the range $k \leq 110$, $k/\log_3 e_f(k)$ attains its maximum at $k = 81$ with $81/\log_3(1504935935) \sim 4.21103$. This suggests the following conjecture, which would give a considerably better bound than the known bound of $\|n\|_{\log} < 4.755$.

▶ **Conjecture 1.** *For every natural number n, $\|n\|_{\log} \leq f(n)_{\log} < 4.212$.*

Table 1 The value of $e_f(k)$ for $90 \le k \le 110$.

k	$e_f(k)$	factorization of $e_f(k)+1$
90	$18,059,231,231$	$2^{17} \cdot 3^9 \cdot 7$
91	$27,088,846,847$	$2^{16} \cdot 3^{10} \cdot 7$
92	$36,118,462,463$	$2^{18} \cdot 3^9 \cdot 7$
93	$54,177,693,691$	$2^2 \cdot 683 \cdot 19830781$
94	$54,177,693,695$	$2^{17} \cdot 3^{10} \cdot 7$
95	$72,236,924,927$	$2^{19} \cdot 3^9 \cdot 7$
96	$108,355,387,390$	prime $(= 2^{18} \cdot 3^{10} \cdot 7 - 1)$
97	$108,355,387,391$	$2^{18} \cdot 3^{10} \cdot 7$
98	$162,533,081,087$	$2^{17} \cdot 3^{11} \cdot 7$
99	$216,710,774,782$	prime $(= 2^{19} \cdot 3^{10} \cdot 7 - 1)$
100	$216,710,774,783$	$2^{19} \cdot 3^{10} \cdot 7$
101	$325,066,162,175$	$2^{18} \cdot 3^{11} \cdot 7$
102	$487,599,243,263$	$2^{17} \cdot 3^{12} \cdot 7$
103	$650,132,324,347$	$2^2 \cdot 13 \cdot 29 \cdot 15443 \cdot 27917$
104	$650,132,324,351$	$2^{19} \cdot 3^{11} \cdot 7$
105	$975,198,486,527$	$2^{18} \cdot 3^{12} \cdot 7$
106	$1,486,016,741,374$	$5^3 \cdot 163 \cdot 72933337$ $(= 2^{23} \cdot 3^{11} - 1)$
107	$1,486,016,741,375$	$2^{23} \cdot 3^{11}$
108	$1,950,396,973,055$	$2^{19} \cdot 3^{12} \cdot 7^1$
109	$3,715,041,853,439$	$2^{22} \cdot 3^{11} \cdot 5$
110	$3,900,793,946,111$	$2^{20} \cdot 3^{12} \cdot 7$

It should be noted that Iraids et al. [11] computed the integer complexity of n for $n \le 10^{12}$. Let $e(k)$ denote the smallest integer n such that $\|n\| \ge k$, i.e., a counterpart of $e_f(k)$. They determined $e(k)$ up to $k \le 89$ (see the sequence A005520 in OEIS). By inspecting the results, they conjectured that $\lim_{k\to\infty} \frac{k}{\log_3 e(k)} \sim 3.37$. This suggests that there is a substantial gap between two measures $f(n)$ and $\|n\|$ in the worst case.

3.2 Average Case

Recall that we write $f(n)_{\log} := f(n)/\log_3 n$. For $k \ge 1$, let E_k denote the average of $f(n)_{\log}$ over all k-bit integers, i.e., $E_k := \mathbf{E}_{n\in[2^{k-1},2^k-1]}[f(n)_{\log}]$. Our computation reveals the exact value of E_k's for $k \le 42$. See Table 2. Interestingly, the value of E_k hits the peak at $k = 36$ with $E_{36} \sim 3.48829$ and then starts to decrease.

In order to clarify this phenomenon, we tried to estimate the value of E_k's for larger values of k by random sampling. Namely, for each $43 \le k \le 100$, we compute $f(n)$ for randomly chosen 2^{25} k-bit integers to estimate E_k. Note that the computation of $f(n)$ itself is exact. See Fig. 2. The value of E_k seems to be steadily decreasing as k is increasing. The value of E_{100} is around 3.483.

We believe that $\lim_{k\to\infty} E_k$ exists. However, at this point, we do not have a reasonable guess on the value of this limit. In Section 4.3, we will show a non-trivial lower bound of $\lim_{k\to\infty} E_k > 3.120$ (if the limit exists).

▶ **Problem 2.** *Does $\lim_{k\to\infty} E_k$ exist? If it exists, find the value.*

For a comparison to the original integer complexity $\|n\|$, let $\tilde{E}_k$ denote a counterpart of E_k. A quick computation (based on Eq. (1)) suggest that $\tilde{E}_k$ seems to be decreasing as k is increasing. For example, $\tilde{E}_{20} \sim 3.3859$, $\tilde{E}_{25} \sim 3.3744$ and $\tilde{E}_{30} \sim 3.3644$ (see also [14] for an

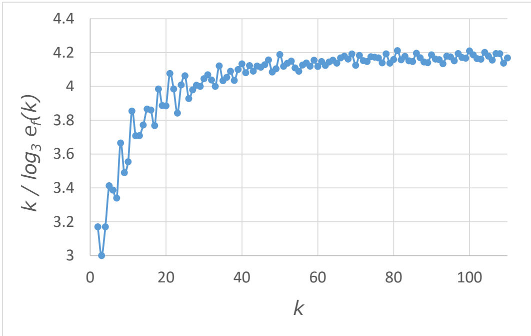

Figure 1 The plot of $k/\log_3 e_f(k)$.

Table 2 The average E_k of $f(n)_{\log}$ over all k-bit integers. The seventh digit from the decimal point is rounded.

k	E_k	k	E_k
20	3.482808	32	3.488126
21	3.483896	33	3.488197
22	3.484733	34	3.488248
23	3.485483	35	3.488275
24	3.486055	36	3.488290
25	3.486533	37	3.488284
26	3.486954	38	3.488270
27	3.487253	39	3.488243
28	3.487529	40	3.488208
29	3.487720	41	3.488164
30	3.487891	42	3.488114
31	3.488024		

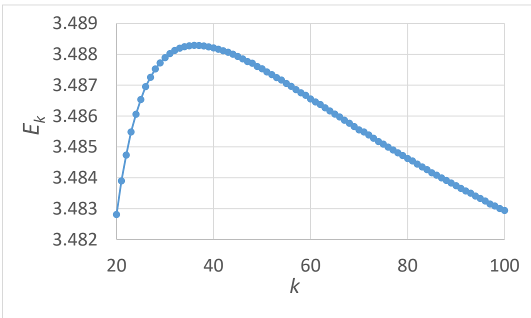

Figure 2 The plot of E_k. The value of E_k for $k \geq 43$ is an estimation by 2^{25} randomly chosen k-bit integers.

earlier experiment). We do not know whether $\lim_{k\to\infty} E_k$ is strictly larger than $\lim_{k\to\infty} \tilde{E}_k$, although the equality is unlikely to hold. Note that no lower bound on $\lim_{k\to\infty} \tilde{E}_k$ better than three is known.

4 Theoretical Results

In this section, we show the items #2 and #3 listed in Introduction. Note that some of the proofs are computer assisted.

4.1 Average Case Upper Bound

The first theoretical result is an upper bound on the average of $f(n)$. Note that the constant ~ 3.587 in the statement of the theorem is a bit worse than the observed average of ~ 3.483 described in the previous section.

▶ **Theorem 3.** *For almost all natural numbers n, $f(n)_{\log} \le 3.587$.*

Here and hereafter, we say that a statement holds for almost all natural numbers n if it holds on a subset of $\mathbb{N}$ of natural density one.

Proof. The proof relies on the argument developed in [7] together with additional computational efforts.

Let $a, b \ge 0$ be two integers (whose values will be chosen later) and put $c = a + b$. Let $S_{a,b}$ be the set of all c-tuples consisting of a 2's and b 3's. Note that $|S_{a,b}| = \binom{c}{a}$. Given an integer $m \in [0, 2^a 3^b - 1]$ and $t = (t_0, t_1, \dots, t_{c-1}) \in S_{a,b}$, let $d_t(m)$ denote the sum of the digits in the mixed binary-ternary representation in which the base of the i-th digit is t_i for $i = 0, 1, \dots, c-1$. That is, $d_t(m) = \sum_{i=0}^{c-1} d_i$ where $m = (d_{c-1})_{t_{c-1}} \cdots (d_1)_{t_1} (d_0)_{t_0}$.

Let $E_{a,b}$ be the expectation of the minimum of $d_t(m)$ where m is chosen uniformly from $[0, 2^a 3^b - 1]$ and the minimum is taken over all $t \in S_{a,b}$, i.e.,

$$E_{a,b} = \frac{1}{2^a 3^b} \sum_{m \in [0, 2^a 3^b - 1]} \min_{t \in S_{a,b}} d_t(m).$$

We first claim that, for almost all natural numbers n, it holds that

$$f(n)_{\log} = \frac{f(n)}{\log_3 n} \le \frac{2a + 3b + E_{a,b}}{\log_3(2^a 3^b)} + \epsilon,$$

for an arbitrarily small $\epsilon > 0$. Note that this claim was essentially shown in [7]. We include the proof here for the completeness.

The proof of the claim is as follows. Let k be a sufficiently large integer and let $n = r_k \cdots r_1 r_0$ in base $2^a 3^b$ such that $r_k \ne 0$. Notice that $n \ge (2^a 3^b)^k$.

By considering the expression based on Horner's expansion (Eq. (4)) to base $2^a 3^b$ representation of n, we see that

$$\frac{f(n)}{\log_3 n} \le \frac{f(r_k)}{k \log_3(2^a 3^b)} + \frac{k(2a + 3b) + \sum_{i=0}^{k-1} \min_{t \in S_{a,b}} d_t(r_i)}{k \log_3(2^a 3^b)}. \tag{7}$$

The first term in Eq. (7) tends to 0 as k tends to ∞. By applying Chernoff bounds, we see that for almost all natural numbers n, the difference

$$\frac{1}{k} \sum_{i=0}^{k-1} \min_{t \in S_{a,b}} d_t(r_i) - \frac{1}{2^a 3^b} \sum_{m \in [0, 2^a 3^b - 1]} \min_{t \in S_{a,b}} d_t(m)$$

is arbitrarily small. By noticing that the second term is equal to $E_{a,b}$, we complete the proof of the claim.

Given a and b, we can calculate $E_{a,b}$ using a computer if a and b are not so large. We calculated $E_{a,b}$ for a and b such that $a+b \leq 31$. The best bound is obtained when $a = 17$ and $b = 14$. We have $E_{17,14} = 7941127132545/(2^{17} \cdot 3^{14})$, which yields that

$$f(n)_{\log n} \leq \frac{76 + E_{17,14}}{\log_3(2^{17} \cdot 3^{14})} \sim 3.58601,$$

for almost all natural numbers n. This completes the proof of Theorem 3. ◀

The argument in the proof of Theorem 3 can be used for obtaining an upper bound on $\|n\|_{\log}$ using higher bases. We conducted the computation for several combinations of the exponents to the bases $\{2, 3, 5, 7\}$. The best bound on $\|n\|_{\log}$ that we have obtained so far is when we consider the base $2^{15}3^{14}5^{1}$. We state this as a separate theorem.

▶ **Theorem 4.** *For almost all natural numbers n, $\|n\|_{\log} \leq 3.556$.*

Proof. Let $a, b, c \geq 0$ be integers and define $E_{a,b,c}$ analogously to $E_{a,b}$, but the base is $2^a3^b5^c$ rather than 2^a3^b. By a similar reasoning to the proof of Theorem 3, we have

$$\|n\|_{\log n} \leq \frac{2a + 3b + 5c + E_{a,b,c}}{\log_3(2^a3^b5^c)} + \epsilon,$$

for an arbitrary small $\epsilon > 0$.

A computer calculation shows $E_{15,14,1} = 9111121836916/(2^{15} \cdot 3^{14} \cdot 5)$. This implies that

$$\|n\|_{\log} \leq \frac{77 + E_{15,14,1}}{\log_3(2^{15} \cdot 3^{14} \cdot 5)} \sim 3.55517,$$

for almost all natural numbers n. ◀

Theorem 4 improves the best-known bound of $\|n\|_{\log} \leq 3.6199$ due to Cordwell et al. [5], which was obtained by calculating $E_{11,9}$ in our terminology. (See also Arias de Reyna and van de Lune [7, Proposition 12] for an earlier bound of $\|n\|_{\log} < 3.635$ based on the calculation of $E_{9,8}$). In this work, we could calculate $E_{a,b}$ for much larger values of a and b than theirs. A key insight is that, for example, a 30-digit mixed binary-ternary representation in base $2^{15}3^{15}$ can be viewed as a concatenation of two 15-digit representations in base $2^{c_1}3^{15-c_1}$ and $2^{15-c_1}3^{c_1}$ for some $0 \leq c_1 \leq 15$. Then, a simple dynamic programming speeds up the calculation.

4.2 Average Case Lower Bound

The second theoretical result is a lower bound on the average of $f(n)$. The following theorem says that the expression based on the mixed binary-ternary representation is not rich enough to give an upper bound of $\|n\|_{\log} \leq (3 + \epsilon)$ for a small ϵ.

▶ **Theorem 5.** *For almost all natural numbers n, $f(n)_{\log} \geq 3.1201$.*

Proof. The proof is by counting argument. Suppose that k is a sufficiently large integer.

Let α be a real number such that $0 \leq \alpha \leq 1$ whose value will be chosen later. Consider a mixed binary-ternary representation of a k-bit integer in which αk digits are base two and $(1-\alpha)k \log_3 2$ digits are base three. Strictly speaking, these two numbers should be integers. This can be fulfilled by choosing α to be a multiple of $1/k$, and then set the number of base-three digits to $\lceil (1-\alpha)k \log_3 2 \rceil$. However, since the effect is negligible, we omit the ceiling functions for clarity.

We will count the number of k-bit integers such that it has a representation satisfying the condition that the sum of the digits is at most βk. The value of β will be chosen appropriately in the later part of the proof.

Given α, the number of orderings of the base-two and base-three digits is given by

$$\binom{(\alpha + (1-\alpha)\log_3 2)\,k}{\alpha k}. \tag{8}$$

Once the bases are fixed to $(b_0, \ldots, b_{(\alpha+(1-\alpha)\log_3 2)k-1})$, the number of integers such that the sum of the digits is βk is at most

$$\binom{(\alpha + 2(1-\alpha)\log_3 2)\,k}{\beta k}. \tag{9}$$

This is because such an integer is the sum of βk numbers chosen from the multiset of size $(\alpha + 2(1-\alpha)\log_3 2)k$ given by

$$\left\{\prod_{j=0}^{i-1} b_j\right\}_i \cup \left\{\prod_{j=0}^{i-1} b_j\right\}_{i:b_i=3},$$

where the element $\prod_{j=0}^{i-1} b_j$ has multiplicity two in the set when $b_i = 3$.

We will use the following standard bound on the binomial coefficients:

$$\binom{n}{k} \leq 2^{nH(n/k)},$$

where $H(\cdot)$ denotes the binary entropy function.

By multiplying Eqs. (8) and (9), we have that the number of k-bit integers that has a mixed binary-ternary representation with αk base-two digits such that the sum of the digits is at most βk is upper bounded by $poly(k) \cdot 2^{\gamma k}$, where

$$\begin{aligned}\gamma \quad = \quad & (\alpha + (1-\alpha)\log_3 2) H\left(\frac{\alpha}{\alpha + (1-\alpha)\log_3 2}\right) \\ & +(\alpha + 2(1-\alpha)\log_3 2) H\left(\frac{\beta}{\alpha + 2(1-\alpha)\log_3 2}\right).\end{aligned}$$

For $0 \leq \alpha \leq 1$, let β_α denote the maximum value of β in the above expression such that $\gamma \leq 0.99999$ (say, γ is strictly smaller than 1).

Notice that, given α, the number of 1's in the expression of n obtained by applying Horner's schema to its mixed binary-ternary representation is given by

$$k(2\alpha + 3(1-\alpha)\log_3 2 + \beta) - O(1),$$

where βk is the sum of the digits.

Note that since $\alpha = \alpha'/k$ for some $\alpha \in [0, k]$, the number of possible α's is $O(k)$. Thus, for every sufficiently large k and for almost all k-bit binary integers n, we have

$$f(n)_{\log} \geq \frac{\min_{0 \leq \alpha \leq 1}\{2\alpha + 3(1-\alpha)\log_3 2 + \beta_\alpha\}}{\log_3 2} - \epsilon$$

for an arbitrary small $\epsilon > 0$. Here we use the fact that $n < 2^k$.

By a simple numerical calculation, we can see that the right-hand side of the above is ~ 3.12017. The minimum is attained at $\alpha \sim 0.3536$ with $\beta_\alpha \sim 0.0379$. This completes the proof of Theorem 5. ◀

▶ Remark 6. The bound in Theorem 5 would slightly be improved if we refine the estimation in Eq. (9) by using trinomial coefficients instead of binary coefficients (by taking into account that the number $\prod_{j=0}^{i-1} b_j$ (for i with $b_i = 3$) appears twice in the ground set). However, the improvement seems to be tiny and hence we did not include a detailed calculation in this version.

4.3 Worst Case Lower Bound

The last result in this section is a lower bound on the worst case of $f(n)$.

▶ **Theorem 7.** *There are infinitely many numbers n such that $f(n)_{\log} \geq 3.934$.*

Proof. The proof is constructive. The key fact is that if $n \equiv -1 \pmod{2^a 3^b}$, then all of a lowest base-two digits as well as b lowest base-three digits have its maximal value, i.e., one or two depending on the base of the digit.

Let $n := \alpha 2^a 3^b - 1$ for a relatively small integer $\alpha > 0$. The values of a and b will be chosen later. For every ordering of the bases in a mixed binary-ternary representation of n, Horner's expression (shown in Eq. (4)) incurs cost three for each of the first a occurrences of the base-two digit, and similarly, incurs cost five for each of the first b occurrences of the base-three digit.

By recalling a general lower bound of $||n|| \geq 3\log_3 n$, we can see that $f(n)$ is lower bounded by the minimum of $3a + 3\log(n/2^a)$ and $5b + 3\log_3(n/3^b)$.

Let $k := b/a$. If a and b are sufficiently large with respect to α, then

$$\begin{aligned} f(n)_{\log} = \frac{f(n)}{\log_3 n} &\geq \frac{\min\{3a + 3ak, 5ak + 3a\log_3 2\}}{a\log_3 2 + ak} - \epsilon \\ &= \frac{\min\{3 + 3k, 5k + 3\log_3 2\}}{\log_3 2 + k} - \epsilon \end{aligned}$$

for an arbitrary small constant $\epsilon > 0$.

A simple calculation shows that the last term attains a maximum of $(15 - 9\log_3 2)/(3 - \log_3 2) \sim 3.9347$ at $k = \frac{3}{2}(1 - \log_3 2) \sim 0.5536$. This completes the proof of the theorem. ◀

▶ Remark 8. In the proof of Theorem 7, we use a (pessimistic) lower bound of $\|n\|_{\log} \geq 3$ to estimate the length of an upper part of an expression. If we use an optimistic (but not proven) constant ~ 3.48 instead of 3, which is suggested by the experiments in Section 3.2, then a simple calculation shows that the bound in the theorem would become ~ 4.17. This seems reasonably consistent with the experiments in Section 3.1 (See Fig. 1). In addition, our proof suggests that the worst case occurs when $n = \alpha 2^a 3^b - 1$ for $b \sim 0.55a$ and small α. This is also consistent with the factorization data shown in Table 1.

5 Beyond Ternary

If our aim is to obtain an upper bound on $\|n\|_{\log}$ rather than $f(n)_{\log}$, there are no reasons to restrict the bases to two or three. We can consider a number system including higher bases.

In [15] (refining the work in [16]), Shriver formalized a method for obtaining an upper bound on $\|n\|_{\log}$ that is satisfied by a set of numbers $n \in \mathbb{N}$ of *logarithmic density one.* The logarithmic density of a set $S \subseteq \mathbb{N}$ of integers is defined by

$$\lim_{N\to\infty} \frac{1}{\log N} \sum_{n\in S, n\leq N} \frac{1}{n}$$

if the limit exists.

Below we describe an outline of their method using a greedy algorithm in base six as an illustrative example. See [15] for more details.

Consider the following algorithm that outputs an expression of a given number n.

1. If $n \leq 5$, output an optimal representation of n. Otherwise, move to step 2.
2. Choose a representation depending on $n \pmod 6$:
   - If $n \pmod 6 \equiv 0$, write as $3(n/3)$,
   - If $n \pmod 6 \equiv 1$, write as $3((n-1)/3)+1$,
   - If $n \pmod 6 \equiv 2$, write as $2(n/2)$,
   - If $n \pmod 6 \equiv 3$, write as $3(n/3)$,
   - If $n \pmod 6 \equiv 4$, write as $2(n/2)$,
   - If $n \pmod 6 \equiv 5$, write as $2((n-1)/2)+1$.

   Then apply step 2 to the result (the number in brackets) until one of the representations in step 1 can be applied.
3. Replace every 2 by (1+1) and every 3 by (1+1+1).

Following [15], we write this algorithm by $D := (3,3,2,3,2,2)$ which represents the divisors of each residue class in step 2.

Divide $\mathbb{N}$ into six residue classes $\{0,1,\dots,5\}$. For example, in step 2, $6m+1$ is going to $2m$, that we rewrite as $6m'$, $6m'+2$ or $6m'+4$. We interpret this as that class 1 is going to class 0, 2, or 4 with equal probability. Then the transition of numbers in each class can be expressed by a six-state Markov model whose transition matrix is

$$M = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{3} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Let π be the stationary distribution of M. An easy calculation shows that

$$\pi^T = \left(\frac{1}{13}, \frac{2}{13}, \frac{4}{13}, 0, \frac{3}{13}, \frac{3}{13}\right).$$

Let C be a 6-tuple whose i-th element represents the number of 1's used in a single step for the residue class i in step 2 of the algorithm. Namely, $C := (3,4,2,3,2,3)$.

Shriver [15] showed that, for an arbitrary small constant $\epsilon > 0$, the algorithm outputs an expression including at most $(\alpha+\epsilon)\log_3 n$ 1's for a set of numbers n with logarithmic density one, where

$$\alpha := \frac{\sum_i \pi(i)C[i]}{\sum_i \pi(i)\log_3 D[i]}.$$

An easy calculation shows that $\alpha \sim 3.6522$ for the above algorithm.

As expected, we can improve a bound by seeking an algorithm for a larger base or/and by optimizing the divisors of each residue class. The best-known bound obtained along this line is by Bevan (described in [15]) that $\|n\|_{\log} < 3.5197$ based on an algorithm in base 2310 $(=2\cdot 3\cdot 5\cdot 7\cdot 11)$.

In this work, we further extend their computational efforts. We found an algorithm in base $2^3\cdot 3^2\cdot 5\cdot 7\cdot 11 = 27,720$, which gives the following improved bound.

▶ **Theorem 9.** *The set of natural numbers n such that $\|n\|_{\log} \leq 3.4713$ has the logarithmic density one.*

The witness of Theorem 9 is the description of a $27,720$-tuple D, which is available at `https://gitlab.com/KazAmano/integer-complexity`[2]. Note that the problem for finding a good algorithm (or, equivalently, finding the divisors for each residue class in our case) is a non-convex optimization problem. We apply the gradient descent method in our computation. In each step, we need to calculate the stationary distribution of a 27,720-state Markov chain many times. This step is efficiently be implemented by using a power method.

6 Discussions

There remain many interesting problems to be explored. One of such is a question whether $f(2^k) = 2k$ for every $k \geq 1$, which has a similar spirit to the question whether $\|2^k\| = 2k$ asked by Selfridge (see [9]). This question has a close connection to a fundamental problem in number theory, i.e., the distribution of digits in the ternary representation of 2^n (discussed in e.g., [8, 12, 17]). Currently, no counterexamples are known. During this work, we verified that $f(2^k) = 2k$ for every $k \leq 200$ by computer calculations.

In [11], Iraids et al. have made an interesting discussion on the distribution of $\|n\|_{\log}$. It is known that $\|n\|_{\log} \in [3, 4755]$. Based on their computational results, they see that the interval $[3, 4755]$ shall be divided into three epochs depending on the density of $\|n\|_{\log}$. From 3 to some constant, say C_1, the values of $\|n\|_{\log}$ is "sparse". Then, from C_1 to some constant, say C_2, the set of values of $\|n\|_{\log}$ is "dense". Finally, from C_2 (to 4.755), the $\|n\|_{\log}$ is "absolutely sparse" meaning that there are only finitely many numbers n such that $\|n\|_{\log} > C_2$. By inspecting their experimental results, they conjectured that

$$3.1699 \sim \|2\|_{\log} \leq C_1 \leq C_2 \leq 3.37.$$

Based on the results of this work, we find that a similar analysis can be made for our complexity measure $f(n)_{\log}$. However, this time, there may be four epochs, namely, "sparse", "dense", "sparse", and "absolutely sparse" from 3 to the largest value of $f(n)_{\log}$. Here the "sparse" means that it contains infinitely many numbers but its density is zero.

Theorems 3 and 5 imply that the "dense" epoch is contained in the interval $[3.120, 3.587]$. Narrowing this interval is interesting future work. Also, Theorem 7 says that the point separating the third and fourth epochs is at least 3.934 (if it exists). Refining this bound as well as obtaining a general upper bound on $f(n)_{\log}$ are also interesting future work.

References

1 Harry J. Altman. Integer complexity: Representing numbers of bounded defect. *Theor. Comput. Sci.*, 652:64–85, 2016. doi:10.1016/j.tcs.2016.09.005.

2 Harry J. Altman. Integer complexity: Algorithms and computational results. *Integers*, 18:A45, 2018. URL: http://math.colgate.edu/%7Eintegers/s45/s45.Abstract.html.

3 Harry J. Altman and Joshua Zelinsky. Numbers with integer complexity close to the lower bound. *Integers*, 12A:A1, 2012. URL: http://math.colgate.edu/%7Eintegers/a1self/a1self.Abstract.html.

4 Juris Cernenoks, Janis Iraids, Martins Opmanis, Rihards Opmanis, and Karlis Podnieks. Integer complexity: Experimental and analytical results II. In Jeffrey O. Shallit and Alexander Okhotin, editors, *Descriptional Complexity of Formal Systems - 17th International Workshop, DCFS 2015, Waterloo, ON, Canada, June 25-27, 2015. Proceedings*, volume 9118 of *Lecture Notes in Computer Science*, pages 58–69. Springer, 2015. doi:10.1007/978-3-319-19225-3_5.

[2] Note that an additional condition that, for an underlying graph of M (which is uniquely determined by D), every vertex has a path to the vertex labeled 1, should be satisfied ([15, Proposition 4.5]). We checked that our D satisfies this condition computationally.

5 Katherine Cordwell, Alyssa Epstein, Anand Hemmandy, Steven J. Miller, Eyvindur A. Palsson, Aaditya Sharma, Stefan Steinerberger, and Yen Nhi Truong Vu. On algorithms to calculate integer complexity. *Integers*, 19:A12, 2019. URL: `http://math.colgate.edu/%7Eintegers/t12/t12.Abstract.html`.

6 J. Arias de Reyna. Complexity of natural numbers, 2021. (translation of Complejidad de los números naturales, Gac. R. Soc. Mat. Esp. 3, 230–250, 2000). `arXiv:2111.03345`.

7 J. Arias de Reyna and J. van de Lune. Algorithms for determining integer complexity, 2014. `arXiv:1404.2183`.

8 Taylor Dupuy and David Weirich. Bits of 3^n in binary, wieferich primes and a conjecture of Erdös. *J. Number Theory*, 158:268–280, 2016. `doi:10.1016/j.jnt.2015.05.022`.

9 Ricard K. Guy. What is the least number of ones needed to represent n using only $+$ and $\times$ (and parentheses)? *Amer. Math. Monthly*, 93:189–190, 1986.

10 Ricard K. Guy. *Unsolved Problems in Number Theory, Third Edition, Problem F26: Expressing numbers with just one*. Springer-Verlag, New York, 2004.

11 Jānis Iraids, Kaspars Balodis, Juris Čerņenoks, Mārtiņš Opmanis, Rihards Opmanis, and Kārlis Podnieks. Integer complexity: Experimental and analytical results. *Scientific Papers University of Latvia, Computer Science and Information Technologies*, 787:153–179, 2012. (also at `arxiv:1203.6462`).

12 Jeffrey C. Lagarias. Ternary expansions of powers of 2. *J. London Mathematical Society*, 79(3):562–588, 2009. `doi:doi.org/10.1112/jlms/jdn080`.

13 K. Mahler and J. Popken. On a maximum problem in arithmetic (dutch). *Nieuw Arch. Wiskunde*, 3(1):1–15, 1953.

14 Daniel A. Rawsthorne. How many 1's are needed? *Fib. Quart.*, 27:14–17, 1989.

15 Christopher E. Shriver. An application of markov chain analysis to integer complexity, 2015. `arXiv:1511.07842`.

16 Stefan Steinerberger. A short note on integer complexity. *Contributions to Discrete Mathematics*, 9:63–69, 2014.

17 Cameron L. Stewart. On the representation of an integer in two different bases. *Journal fur die reine und angewandte Mathematik*, 319:63–72, 1980.

18 Venecia Wang. A counterexample to the prime conjecture of expressing numbers using just ones. *J. Number Theory*, 133(2), 2013. `doi:10.1016/j.jnt.2012.08.003`.

19 Emre Yolcu, Scott Aaronson, and Marijn J. H. Heule. An automated approach to the collatz conjecture. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 468–484. Springer, 2021. `doi:10.1007/978-3-030-79876-5_27`.

A Appendix A

In the following, we show the full list of $e_f(k)$ for $k \leq 110$ in Tables 3, 4 and 5.

Table 3 The value of $e_f(k)$ for $k \leq 40$.

k	$e_f(k)$	factorization of $e_f(k)+1$
1	1	2
2	2	3
3	3	2^2
4	4	5
5	5	$2 \cdot 3$
6	7	2^3
7	10	11
8	11	$2^2 \cdot 3$
9	17	$2 \cdot 3^2$
10	22	23
11	23	$2^3 \cdot 3$
12	35	$2^2 \cdot 3^2$
13	47	$2^4 \cdot 3$
14	59	$2^2 \cdot 3 \cdot 5$
15	71	$2^3 \cdot 3^2$
16	95	$2^5 \cdot 3$
17	142	$11 \cdot 13\ (= 2^4 \cdot 3^2 - 1)$
18	143	$2^4 \cdot 3^2$
19	215	$2^3 \cdot 3^3$
20	286	$7 \cdot 41\ (= 2^5 \cdot 3^2 - 1)$
21	287	$2^5 \cdot 3^2$
22	431	$2^4 \cdot 3^3$
23	718	prime $(= 2^4 \cdot 3^2 \cdot 5 - 1)$
24	719	$2^4 \cdot 3^2 \cdot 5$
25	863	$7 \cdot 103\ (= 2^5 \cdot 3^3)$
26	$1,439$	$2^5 \cdot 3^2 \cdot 5$
27	$1,727$	$2^6 \cdot 3^3$
28	$2,159$	$2^4 \cdot 3^3 \cdot 5$
29	$2,879$	$2^6 \cdot 3^2 \cdot 5$
30	$3,455$	$2^7 \cdot 3^3$
31	$4,319$	$2^5 \cdot 3^3 \cdot 5$
32	$6,047$	$2^5 \cdot 3^3 \cdot 7$
33	$8,638$	$53 \cdot 163\ (= 2^6 \cdot 3^3 \cdot 5 - 1)$
34	$8,639$	$2^6 \cdot 3^3 \cdot 5$
35	$13,823$	$2^9 \cdot 3^3$
36	$17,279$	$2^7 \cdot 3^3 \cdot 5$
37	$20,735$	$2^8 \cdot 3^4$
38	$31,103$	$2^7 \cdot 3^5$
39	$34,559$	$2^8 \cdot 3^3 \cdot 5$
40	$41,471$	$2^9 \cdot 3^4$

Table 4 The value of $e_f(k)$ for $41 \leq k \leq 80$.

k	$e_f(k)$	factorization of $e_f(k)+1$
41	$62,207$	$2^8 \cdot 3^5$
42	$72,575$	$2^7 \cdot 3^4 \cdot 7$
43	$103,679$	$2^8 \cdot 3^4 \cdot 5$
44	$124,415$	$2^9 \cdot 3^5$
45	$165,887$	$2^{11} \cdot 3^4$
46	$207,359$	$2^9 \cdot 3^4 \cdot 5$
47	$248,831$	$2^{10} \cdot 3^5$
48	$404,351$	$2^7 \cdot 3^5 \cdot 13$
49	$497,662$	prime $(= 2^{11} \cdot 3^5 - 1)$
50	$497,663$	$2^{11} \cdot 3^5$
51	$808,703$	$2^8 \cdot 3^5 \cdot 13$
52	$995,327$	$2^{12} \cdot 3^5$
53	$1,244,159$	$2^{10} \cdot 3^5 \cdot 5$
54	$1,866,239$	$2^9 \cdot 3^6 \cdot 5$
55	$2,612,735$	$2^9 \cdot 3^6 \cdot 7$
56	$2,985,983$	$2^{12} \cdot 3^6$
57	$3,732,479$	$2^{10} \cdot 3^6 \cdot 5$
58	$5,225,471$	$2^{10} \cdot 3^6 \cdot 7$
59	$5,971,967$	$2^{13} \cdot 3^6$
60	$8,957,951$	$2^{12} \cdot 3^7$
61	$10,450,943$	$2^{11} \cdot 3^6 \cdot 7$
62	$14,929,919$	$2^{12} \cdot 3^6 \cdot 5$
63	$17,915,903$	$2^{13} \cdot 3^7$
64	$22,394,879$	$2^{11} \cdot 3^7 \cdot 5$
65	$31,352,831$	$2^{11} \cdot 3^7 \cdot 7$
66	$35,831,807$	$2^{14} \cdot 3^7$
67	$44,789,759$	$2^{12} \cdot 3^7 \cdot 5$
68	$62,705,663$	$2^{12} \cdot 3^7 \cdot 7$
69	$71,663,615$	$2^{15} \cdot 3^7$
70	$125,411,326$	prime $(= 2^{13} \cdot 3^7 \cdot 7 - 1)$
71	$125,411,327$	$2^{13} \cdot 3^7 \cdot 7$
72	$188,116,991$	$2^{12} \cdot 3^8 \cdot 7$
73	$250,822,655$	$2^{14} \cdot 3^7 \cdot 7$
74	$286,654,463$	$2^{17} \cdot 3^7$
75	$376,233,983$	$2^{13} \cdot 3^8 \cdot 7$
76	$501,645,311$	$2^{15} \cdot 3^7 \cdot 7$
77	$752,467,966$	$3257 \cdot 231031$ $(= 2^{14} \cdot 3^8 \cdot 7 - 1)$
78	$752,467,967$	$2^{14} \cdot 3^8 \cdot 7$
79	$1,289,945,087$	$2^{16} \cdot 3^9$
80	$1,504,935,934$	$5 \cdot 300987187$ $(= 2^{15} \cdot 3^8 \cdot 7 - 1)$

Table 5 The value of $e_f(k)$ for $81 \leq k \leq 110$.

k	$e_f(k)$	factorization of $e_f(k)+1$
81	$1,504,935,935$	$2^{15} \cdot 3^8 \cdot 7$
82	$2,579,890,175$	$2^{17} \cdot 3^9$
83	$3,009,871,871$	$2^{16} \cdot 3^8 \cdot 7$
84	$4,514,807,807$	$2^{15} \cdot 3^9 \cdot 7$
85	$6,019,743,742$	$13 \cdot 463057211$ $(= 2^{17} \cdot 3^8 \cdot 7 - 1)$
86	$6,019,743,743$	$2^{17} \cdot 3^8 \cdot 7$
87	$9,029,615,615$	$2^{16} \cdot 3^9 \cdot 7$
88	$13,544,423,423$	$2^{15} \cdot 3^{10} \cdot 7$
89	$18,059,231,227$	$2^2 \cdot 4514807807$
90	$18,059,231,231$	$2^{17} \cdot 3^9 \cdot 7$
91	$27,088,846,847$	$2^{16} \cdot 3^{10} \cdot 7$
92	$36,118,462,463$	$2^{18} \cdot 3^9 \cdot 7$
93	$54,177,693,691$	$2^2 \cdot 683 \cdot 19830781$
94	$54,177,693,695$	$2^{17} \cdot 3^{10} \cdot 7$
95	$72,236,924,927$	$2^{19} \cdot 3^9 \cdot 7$
96	$108,355,387,390$	prime $(= 2^{18} \cdot 3^{10} \cdot 7 - 1)$
97	$108,355,387,391$	$2^{18} \cdot 3^{10} \cdot 7$
98	$162,533,081,087$	$2^{17} \cdot 3^{11} \cdot 7$
99	$216,710,774,782$	prime $(= 2^{19} \cdot 3^{10} \cdot 7 - 1)$
100	$216,710,774,783$	$2^{19} \cdot 3^{10} \cdot 7$
101	$325,066,162,175$	$2^{18} \cdot 3^{11} \cdot 7$
102	$487,599,243,263$	$2^{17} \cdot 3^{12} \cdot 7$
103	$650,132,324,347$	$2^2 \cdot 13 \cdot 29 \cdot 15443 \cdot 27917$
104	$650,132,324,351$	$2^{19} \cdot 3^{11} \cdot 7$
105	$975,198,486,527$	$2^{18} \cdot 3^{12} \cdot 7$
106	$1,486,016,741,374$	$5^3 \cdot 163 \cdot 72933337$ $(= 2^{23} \cdot 3^{11} - 1)$
107	$1,486,016,741,375$	$2^{23} \cdot 3^{11}$
108	$1,950,396,973,055$	$2^{19} \cdot 3^{12} \cdot 7^1$
109	$3,715,041,853,439$	$2^{22} \cdot 3^{11} \cdot 5$
110	$3,900,793,946,111$	$2^{20} \cdot 3^{12} \cdot 7$