The Lichtenberg Sequence:

A Complete Mathematical Analysis

Binary Alternation, Dual Recurrences, and Ratio Convergence

A Pedagogical Approach with Rigorous Proofs

David Coates
Independent Researcher
Belfast, Northern Ireland
gauntletin@gmail.com

November 11, 2025

Abstract

This paper provides a comprehensive, pedagogically-oriented mathematical analysis of the Lichtenberg sequence (OEIS A000975), suitable for advanced undergraduate or beginning graduate courses in discrete mathematics. We develop the theory systematically from first principles, proving all major results with complete rigor while maintaining accessibility for students. The sequence, defined by alternating recurrence relations $a(2n) = 2 \cdot a(2n-1)$ and $a(2n+1) = 2 \cdot a(2n) + 1$, generates numbers whose binary representations contain no consecutive equal digits. We establish that this sequence can equivalently be characterized by the single recurrence $L(n) + L(n-1) = 2^n - 1$, proving these formulations are mathematically identical. Through detailed analysis, we characterize the sequence's ratio convergence behavior (alternating between exactly 2.0 and values converging exponentially to 2.0), prove the 50% duty cycle property of binary representations, derive closed-form expressions, and establish modular arithmetic patterns. Each major result includes complete proofs, worked examples, and exercises for students. Applications to digital signal processing, combinatorics, and information theory demonstrate practical relevance. This treatment emphasizes mathematical clarity, computational verification, and pedagogical value.

Keywords: Lichtenberg sequence, OEIS A000975, alternating recurrence, binary patterns, ratio convergence, discrete mathematics, mathematical pedagogy
2020 Mathematics Subject Classification: 11B37, 11B39, 05A15, 11A07
Prerequisites: Basic discrete mathematics, elementary number theory, binary representations, mathematical induction, limit theory

Contents

1 Introduction and Motivation

1.1 What is the Lichtenberg Sequence?

The Lichtenberg sequence, catalogued as OEIS A000975, is one of the most elegant examples of how simple rules can generate rich mathematical structure. Named after Georg Christoph Lichtenberg's 1769 studies of electrical discharge patterns, the sequence has modern applications in digital communications, error-correcting codes, and combinatorics.

Definition 1.1 (Lichtenberg Sequence)**.** The sequence $\{a(n)\}_{n\geq 0}$ is defined by the initial conditions and recurrence relations:

$$a(0) = 0, \quad a(1) = 1 \tag{1}$$
$$a(2n) = 2 \cdot a(2n-1) \quad \text{for } n \geq 1 \tag{2}$$
$$a(2n+1) = 2 \cdot a(2n) + 1 \quad \text{for } n \geq 0 \tag{3}$$

First terms: 0, 1, 2, 5, 10, 21, 42, 85, 170, 341, 682, 1365, 2730, 5461, 10922, 21845, ...

Teaching Note

Students should compute the first 10–15 terms by hand using the recurrence relations. This reinforces the alternating nature: even indices double the previous term, odd indices double and add 1. Notice the pattern already: every even-indexed ratio is exactly 2, while odd-indexed ratios are slightly above 2.

1.2 The Binary Alternation Property

The defining characteristic that makes this sequence special is revealed when we write terms in binary:

n	$a(n)$	Binary Representation
1	1	1
2	2	10
3	5	101
4	10	1010
5	21	10101
6	42	101010
7	85	1010101
8	170	10101010
9	341	101010101
10	682	1010101010

Key observation: Every term (except $a(0) = 0$) has a binary representation that perfectly alternates between 1 and 0, with no consecutive equal digits.

Exercise 1.2. Verify that $a(12) = 2730$ has binary representation 101010101010. Confirm it follows the alternation pattern.

1.3 Why Study This Sequence?

Mathematical Interest:

- Demonstrates how alternating recurrences create structure
- Shows connection between algebraic definitions and digital properties

- Exhibits elegant ratio convergence behavior
- Has multiple equivalent characterizations (duality)

Practical Applications:

- Digital signal processing: Alternating bit patterns minimize DC bias
- Error detection: Run-length limited codes use consecutive-digit avoidance
- Data transmission: Balanced signals improve reliability
- Combinatorics: Counts structures with alternation constraints

Teaching Note

This sequence is pedagogically valuable because it's simple enough for undergraduates to understand completely, yet rich enough to illustrate major themes in discrete mathematics: recurrence relations, asymptotic analysis, binary representations, and the interplay between algebraic and combinatorial viewpoints.

1.4 Learning Objectives

By the end of this paper, students will be able to:

1. Compute sequence values using both recurrence formulations
2. Prove the duality between alternating and single recurrence forms
3. Analyze ratio convergence and prove exact characterizations
4. Understand binary structure and duty cycle properties
5. Apply asymptotic and modular arithmetic techniques
6. Connect theoretical results to practical applications

2 Detailed Numerical Analysis

Before proving general theorems, we carefully examine the numerical evidence. This builds intuition and motivates the theoretical development.

2.1 Computing the Sequence

Students should begin by computing the first 15–20 terms using both recurrence formulations to build intuition before proceeding to the pattern analysis.

2.2 Pattern Recognition

n	$a(n)$	Binary	Bits	$a(n)/a(n-1)$	Error from 2.0	Parity
1	1	1	1	—	—	odd
2	2	10	2	2.0000	0.0000	even
3	5	101	3	2.5000	0.5000	odd
4	10	1010	4	2.0000	0.0000	even
5	21	10101	5	2.1000	0.1000	odd
6	42	101010	6	2.0000	0.0000	even
7	85	1010101	7	2.0238	0.0238	odd
8	170	10101010	8	2.0000	0.0000	even
9	341	101010101	9	2.0059	0.0059	odd
10	682	1010101010	10	2.0000	0.0000	even
11	1365	10101010101	11	2.0015	0.0015	odd
12	2730	101010101010	12	2.0000	0.0000	even

From this table, three fundamental patterns emerge:

Pattern 1 (Binary Alternation): Perfect $101010\ldots$ structure with no consecutive equal bits.

Pattern 2 (Ratio Alternation):

- At even indices: $a(n)/a(n-1) = 2.0$ exactly
- At odd indices: $a(n)/a(n-1) > 2.0$, decreasing toward 2.0

Pattern 3 (Exponential Error Decay): The errors at odd indices form a rapidly decreasing sequence:

$$0.5, \quad 0.1, \quad 0.024, \quad 0.0059, \quad 0.0015, \quad 0.00037, \ldots$$

The ratios between consecutive errors:

$$\frac{0.1}{0.5} = 0.2, \quad \frac{0.024}{0.1} \approx 0.24, \quad \frac{0.0059}{0.024} \approx 0.246$$

This suggests errors decrease by approximately a factor of 4 each time, indicating exponential convergence.

Exercise 2.1. Compute $a(13)$, $a(14)$, $a(15)$ and their binary representations. Verify the alternating ratio pattern continues.

Teaching Note

Spend significant time on this numerical analysis in class. Have students compute values, plot ratios, and discover patterns before revealing the theorems. This inquiry-based approach builds deeper understanding than simply presenting proofs.

3 The Fundamental Duality

3.1 A Surprising Discovery

While studying the Lichtenberg sequence, we encounter what appears to be a contradiction in the literature. Some sources define it using our alternating recurrence (Definition 1.1), while others use a completely different characterization:

Definition 3.1 (Alternative Characterization)**.** Define sequence $\{L(n)\}_{n\geq 0}$ by:

$$L(n) + L(n-1) = 2^n - 1, \quad L(0) = 0, \quad L(1) = 1$$

Question: Do these generate the same sequence?
Let's compute the first few terms of $L(n)$ to check:

$$\begin{aligned}
L(0) &= 0 \quad \text{(given)} \\
L(1) &= 1 \quad \text{(given)} \\
L(2) + L(1) &= 2^2 - 1 = 3 \quad \Rightarrow \quad L(2) = 3 - 1 = 2 \\
L(3) + L(2) &= 2^3 - 1 = 7 \quad \Rightarrow \quad L(3) = 7 - 2 = 5 \\
L(4) + L(3) &= 2^4 - 1 = 15 \quad \Rightarrow \quad L(4) = 15 - 5 = 10 \\
L(5) + L(4) &= 2^5 - 1 = 31 \quad \Rightarrow \quad L(5) = 31 - 10 = 21
\end{aligned}$$

We get: $0, 1, 2, 5, 10, 21, \ldots$ — exactly the Lichtenberg sequence!

Theorem 3.2 (Fundamental Duality)**.** *The alternating recurrence formulation (Definition 1.1) and the single recurrence formulation (Definition 3.1) generate identical sequences. That is, for all $n \geq 0$:*

$$a(n) = L(n)$$

This is a remarkable result: two seemingly different recursive definitions produce the same sequence. We'll prove this rigorously by showing both directions.

3.2 Proof Strategy

We'll prove Theorem 3.2 by deriving closed-form expressions from the single recurrence, then showing these closed forms satisfy the alternating recurrence rules.

Proof outline:

1. Use the single recurrence at even indices to find a pattern for odd-indexed terms
2. Verify this pattern satisfies the alternating doubling rule
3. Use the single recurrence at odd indices to find the pattern for even-indexed terms
4. Verify this satisfies the doubling-plus-one rule
5. Confirm initial conditions match

Teaching Note

Before presenting the proof, ask students: "How would YOU prove two sequences are equal?" Discuss strategies: induction, closed forms, generating functions. The closed-form approach we use is most illuminating here because it reveals the underlying structure.

3.3 Rigorous Proof of Duality

Proof of Theorem 3.2. **Step 1: Closed form for odd indices**
Consider the single recurrence at an even index $n = 2k$:

$$L(2k) + L(2k-1) = 2^{2k} - 1 = 4^k - 1$$

Therefore:

$$L(2k) = 4^k - 1 - L(2k-1)$$

Now, if $L(2k) = 2 \cdot L(2k-1)$ (the alternating rule for even indices), then substituting:

$$\begin{aligned} 2 \cdot L(2k-1) &= 4^k - 1 - L(2k-1) \\ 3 \cdot L(2k-1) &= 4^k - 1 \\ L(2k-1) &= \frac{4^k-1}{3} = \frac{2^{2k}-1}{3} \end{aligned}$$

This is our candidate closed form for odd indices. Let's verify it works.

Verification for odd indices: We need to check that this formula is consistent with both the initial condition and the single recurrence.

- Initial condition: $L(1) = \frac{4^1-1}{3} = \frac{3}{3} = 1$ ✓
- For $k = 2$: $L(3) = \frac{4^2-1}{3} = \frac{15}{3} = 5$ ✓
- For $k = 3$: $L(5) = \frac{4^3-1}{3} = \frac{63}{3} = 21$ ✓

Step 2: Verify even-index doubling
Using the closed form for odd indices:

$$\begin{aligned} L(2k) &= 4^k - 1 - L(2k-1) \\ &= 4^k - 1 - \frac{4^k-1}{3} \\ &= \frac{3(4^k-1) - (4^k-1)}{3} \\ &= \frac{2(4^k-1)}{3} \\ &= 2 \cdot \frac{4^k-1}{3} \\ &= 2 \cdot L(2k-1) \end{aligned}$$

Therefore $L(2k) = 2 \cdot L(2k-1)$ holds exactly for all $k \geq 1$. This is precisely the alternating recurrence rule for even indices!

Step 3: Closed form for even indices
From Step 2, we have:

$$L(2k) = \frac{2(4^k-1)}{3} = \frac{2^{2k+1}-2}{3}$$

Verification: $L(2) = \frac{2^3-2}{3} = \frac{6}{3} = 2$ ✓, $L(4) = \frac{2^5-2}{3} = \frac{30}{3} = 10$ ✓

Step 4: Verify odd-index pattern
For odd index $n = 2k+1$, the single recurrence gives:

$$L(2k+1) + L(2k) = 2^{2k+1} - 1$$

Therefore:

$$\begin{aligned}
L(2k+1) &= 2^{2k+1} - 1 - L(2k) \\
&= 2 \cdot 4^k - 1 - \frac{2(4^k-1)}{3} \\
&= \frac{6 \cdot 4^k - 3 - 2 \cdot 4^k + 2}{3} \\
&= \frac{4 \cdot 4^k - 1}{3} \\
&= \frac{4(4^k-1)+3}{3} \\
&= \frac{4(4^k-1)}{3} + 1 \\
&= 2 \cdot \frac{2(4^k-1)}{3} + 1 \\
&= 2 \cdot L(2k) + 1
\end{aligned}$$

Therefore $L(2k+1) = 2 \cdot L(2k) + 1$ holds exactly for all $k \geq 0$. This is precisely the alternating recurrence rule for odd indices!

Step 5: Initial conditions

Both formulations have $a(0) = L(0) = 0$ and $a(1) = L(1) = 1$ by definition.

Conclusion: Since $L(n)$ satisfies:

- The same initial conditions as $a(n)$
- The same alternating recurrence relations as $a(n)$

By uniqueness of solutions to recurrence relations, we must have $L(n) = a(n)$ for all $n \geq 0$. □

Teaching Note

This proof is long but every step is elementary. Walk through it carefully, emphasizing the strategy: we're showing that the closed forms derived from one definition satisfy the rules of the other definition. Students often struggle with "why" proofs work; here the logic is transparent.

3.4 Closed-Form Summary

As a corollary of our proof, we have explicit formulas:

Corollary 3.3 (Closed-Form Expressions)**.** *For the Lichtenberg sequence:*

$$a(2k-1) = \frac{4^k-1}{3} = \frac{2^{2k}-1}{3} \quad \textit{(odd indices)} \tag{4}$$

$$a(2k) = \frac{2(4^k-1)}{3} = \frac{2^{2k+1}-2}{3} \quad \textit{(even indices)} \tag{5}$$

Example 3.4. Using the closed forms:

- $a(7) = a(2 \cdot 4 - 1) = \frac{4^4-1}{3} = \frac{255}{3} = 85$
- $a(8) = a(2 \cdot 4) = \frac{2(4^4-1)}{3} = \frac{510}{3} = 170$
- $a(11) = a(2 \cdot 6 - 1) = \frac{4^6-1}{3} = \frac{4095}{3} = 1365$

Exercise 3.5. Use the closed forms to compute $a(13)$, $a(14)$, $a(16)$. Verify your answers match Exercise 2.1.

Remark 3.6. The closed forms reveal why the sequence grows so fast: terms are approximately $\frac{2}{3} \cdot 4^{n/2} = \frac{2}{3} \cdot 2^n$, exponential growth with base $\sqrt{2} \approx 1.414$ when indexed uniformly.

4 Ratio Convergence: Complete Analysis

4.1 The Exact Ratio Theorem

We now prove rigorously what we observed numerically: the alternating pattern in successive ratios.

Theorem 4.1 (Ratio Behavior)**.** *For the Lichtenberg sequence* $\{a(n)\}$*:*

1. ***Even indices:*** $\dfrac{a(2n)}{a(2n-1)} = 2$ *exactly for all* $n \geq 1$

2. ***Odd indices:*** $\dfrac{a(2n+1)}{a(2n)} = 2 + \dfrac{1}{a(2n)}$ *for all* $n \geq 0$

3. ***Convergence:*** $\lim_{n\to\infty} \dfrac{a(2n+1)}{a(2n)} = 2$

Proof. **Part (a):** This follows immediately from the alternating recurrence:

$$a(2n) = 2 \cdot a(2n-1) \quad \Rightarrow \quad \frac{a(2n)}{a(2n-1)} = 2$$

This holds for all $n \geq 1$ by definition.
Part (b): From the alternating recurrence $a(2n+1) = 2 \cdot a(2n) + 1$:

$$\frac{a(2n+1)}{a(2n)} = \frac{2 \cdot a(2n) + 1}{a(2n)} = \frac{2 \cdot a(2n)}{a(2n)} + \frac{1}{a(2n)} = 2 + \frac{1}{a(2n)}$$

This formula is exact, not approximate.
Part (c): We need to show $a(2n) \to \infty$ as $n \to \infty$.
From the closed form (Corollary 3.3):

$$a(2n) = \frac{2(4^n - 1)}{3} = \frac{2 \cdot 4^n - 2}{3}$$

As $n \to \infty$, clearly $4^n \to \infty$, so $a(2n) \to \infty$.
Therefore:

$$\lim_{n\to\infty} \frac{a(2n+1)}{a(2n)} = \lim_{n\to\infty} \left(2 + \frac{1}{a(2n)}\right) = 2 + 0 = 2$$

□

Teaching Note

Part (b) is the key insight students often miss: the "error" above 2 is exactly $1/a(2n)$, not just approximately. This precise relationship makes the convergence analysis clean and exact.

4.2 Convergence Rate Analysis

How fast do the ratios converge to 2? Let's quantify the error precisely.

Proposition 4.2 (Exponential Convergence)**.** *The error in odd-index ratios satisfies:*

$$\left|\frac{a(2n+1)}{a(2n)} - 2\right| = \frac{1}{a(2n)} \sim \frac{3}{2^{2n+1}}$$

This error decreases by approximately a factor of 4 between consecutive odd indices (i.e., when moving from index $2n+1$ to index $2n+3$).

Proof. From Theorem 4.1(b), the error is exactly $\frac{1}{a(2n)}$.

From the closed form:

$$a(2n) = \frac{2(4^n - 1)}{3} = \frac{2 \cdot 4^n - 2}{3}$$

For large n, the -2 term is negligible compared to $2 \cdot 4^n$, giving:

$$a(2n) \sim \frac{2 \cdot 4^n}{3} = \frac{2 \cdot 2^{2n}}{3} = \frac{2^{2n+1}}{3}$$

Therefore:

$$\frac{1}{a(2n)} \sim \frac{3}{2^{2n+1}}$$

To show the factor-of-4 decrease, we compare errors at consecutive odd indices. From index $2n+1$ to index $2(n+1)+1 = 2n+3$, the corresponding even indices go from $2n$ to $2n+2$.

From the recurrence:

$$a(2n+2) = 2 \cdot a(2n+1) = 2(2 \cdot a(2n) + 1) = 4 \cdot a(2n) + 2 \approx 4 \cdot a(2n)$$

for large n.

For large n, the $+2$ term becomes negligible compared to $4 \cdot a(2n)$, giving:

$$a(2n+2) \approx 4 \cdot a(2n)$$

Therefore:

$$\frac{1/a(2n+2)}{1/a(2n)} = \frac{a(2n)}{a(2n+2)} \approx \frac{1}{4}$$

The error decreases by approximately a factor of 4 as we move from one odd index to the next odd index. □

Example 4.3 (Numerical Verification)**.** Let's verify the convergence rate numerically:

n	$a(2n)$	Error $= 1/a(2n)$	Ratio to previous
1	2	0.5000	—
2	10	0.1000	0.200
3	42	0.0238	0.238
4	170	0.0059	0.248
5	682	0.0015	0.254

The ratio between consecutive errors approaches $1/4 = 0.25$, confirming the factor-of-4 decrease.

Teaching Note

Have students verify this numerically in a spreadsheet or programming language. Seeing the abstract "$\sim$" notation materialize as actual decimal approximations deepens understanding of asymptotic analysis.

4.3 Growth Rate Analysis

Let's establish the precise growth rate of the sequence.

Lemma 4.4 (Exponential Growth)**.** *For large n:*

$$a(n) \sim \frac{2}{3} \cdot 2^{\lceil n/2 \rceil + 1}$$

Proof. From the closed forms in Corollary 3.3:

For even $n = 2k$:

$$a(2k) = \frac{2(4^k - 1)}{3} = \frac{2 \cdot 4^k}{3} - \frac{2}{3} \sim \frac{2 \cdot 4^k}{3} = \frac{2 \cdot 2^{2k}}{3} = \frac{2^{2k+1}}{3}$$

Since $2k = n$ and $\lceil n/2 \rceil = k$ for even n:

$$a(n) \sim \frac{2^{n+1}}{3} = \frac{2}{3} \cdot 2^{n+1}$$

For odd $n = 2k - 1$:

$$a(2k-1) = \frac{4^k - 1}{3} \sim \frac{4^k}{3} = \frac{2^{2k}}{3}$$

Since $n = 2k - 1$, we have $k = (n + 1)/2$ and $\lceil n/2 \rceil = k$:

$$a(n) \sim \frac{2^{2k}}{3} = \frac{2^{n+1}}{3} = \frac{2}{3} \cdot 2^{n+1}$$

Both cases give the same asymptotic form when using the ceiling function appropriately. □

Remark 4.5. The growth rate is exponential with effective base $\sqrt{2} \approx 1.414$ when we index uniformly through the sequence. This is slower than Fibonacci (base $\phi \approx 1.618$) but faster than linear.

Exercise 4.6. Compute $\log_2 a(n)$ for $n = 2, 4, 6, 8, 10, 12$. Plot these values versus n. The slope should be approximately $1/2$, reflecting the base-$\sqrt{2}$ exponential growth.

5 Binary Structure: The 50% Duty Cycle

5.1 Perfect Alternation Property

We now prove rigorously that the binary alternation pattern holds for all terms.

Theorem 5.1 (Binary Alternation)**.** *For all $n \geq 1$, the binary representation of $a(n)$ contains no consecutive equal digits. That is, no substrings "00" or "11" appear.*

Proof. We prove by strong induction on n.

Base cases:

- $a(1) = 1 = 1_2$ ✓(single digit, trivially alternating)
- $a(2) = 2 = 10_2$ ✓(alternates)
- $a(3) = 5 = 101_2$ ✓(alternates)

Inductive step: Assume $a(k)$ has perfect binary alternation for all $k \leq n$. We prove $a(n+1)$ also alternates.

Case 1: n is odd, so $n+1$ is even.

Then $a(n+1) = 2 \cdot a(n)$ by the recurrence. In binary, multiplying by 2 is a left shift (appends a 0 to the right).

If $a(n)$ has binary representation $b_m b_{m-1} \dots b_1 b_0$, then:

$$a(n+1) = 2 \cdot a(n) \text{ has binary representation } b_m b_{m-1} \dots b_1 b_0 0$$

By induction, $a(n)$ alternates, so $b_0 = 1$ (odd numbers end in 1). Therefore $a(n+1)$ ends in 10, which alternates. Since the rest alternates by induction, and we've added 0 after $b_0 = 1$, the entire string alternates.

Case 2: n is even, so $n+1$ is odd.

Then $a(n+1) = 2 \cdot a(n) + 1$ by the recurrence. This is (left shift by 1, then add 1), which in binary is equivalent to (left shift, then change the last bit from 0 to 1).

If $a(n)$ has binary representation $b_m b_{m-1} \dots b_1 b_0$, then:

- $2 \cdot a(n)$ has representation $b_m b_{m-1} \dots b_1 b_0 0$
- Adding 1 changes the last 0 to 1: $a(n+1) = 2 \cdot a(n) + 1$ has representation $b_m b_{m-1} \dots b_1 b_0 1$

By induction, $a(n)$ alternates, so $b_0 = 0$ (even non-zero terms end in 0). Since $a(n)$ alternates, we have $b_1 = 1$, $b_0 = 0$. After doubling, we get $\dots 100$. Adding 1 gives $\dots 101$, which alternates perfectly.

By induction, $a(n+1)$ alternates for all $n \geq 1$. □

Teaching Note

This proof uses bit-manipulation reasoning. Students comfortable with binary arithmetic will find this natural. For those less comfortable, work through examples like $a(7) = 85 = 1010101_2$ becoming $a(8) = 170 = 10101010_2$ to see the left-shift operation concretely.

5.2 The 50% Duty Cycle Theorem

A remarkable consequence of perfect alternation is that, in the limit, exactly half the bits are 1s and half are 0s.

Theorem 5.2 (50% Duty Cycle)**.** *Let $b(n)$ = number of 1-bits in the binary representation of $a(n)$, and let $d(n)$ = total number of bits. Then:*

$$\lim_{n \to \infty} \frac{b(n)}{d(n)} = \frac{1}{2}$$

Proof. By Theorem 5.1, $a(n)$ (for $n \geq 1$) has the form $10101\dots$ with perfect alternation.

Case 1: $d(n)$ is even.

Then the binary string is $\underbrace{101010\dots10}_{d(n) \text{ bits}}$, which has exactly $d(n)/2$ ones and $d(n)/2$ zeros.

Thus:

$$\frac{b(n)}{d(n)} = \frac{1}{2} \quad \text{(exactly)}$$

Case 2: $d(n)$ is odd.

The string starts with 1 (since $a(n) \geq 1$ for $n \geq 1$), so the pattern is $\underbrace{1010\dots101}_{d(n) \text{ bits}}$. This has $\lceil d(n)/2 \rceil$ ones and $\lfloor d(n)/2 \rfloor$ zeros. Thus:

$$\frac{b(n)}{d(n)} = \frac{\lceil d(n)/2 \rceil}{d(n)} = \frac{d(n)/2 + 1/2}{d(n)} = \frac{1}{2} + \frac{1}{2d(n)}$$

As $n \to \infty$, we have $d(n) \to \infty$ (proven in Proposition 5.3 below), so:

$$\frac{b(n)}{d(n)} \to \frac{1}{2} + 0 = \frac{1}{2}$$

In both cases, the limiting proportion is $1/2$. □

Proposition 5.3 (Bit Length). *The number of bits $d(n)$ in $a(n)$ satisfies:*

$$d(n) = \left\lceil \frac{n+1}{2} \right\rceil + 1 \quad \textit{for } n \geq 6$$

Proof. The number of bits in the binary representation of an integer m is $\lfloor \log_2 m \rfloor + 1$.
From Lemma 4.4, $a(n) \sim \frac{2}{3} \cdot 2^{\lceil n/2 \rceil + 1}$, so:

$$\log_2 a(n) \sim \log_2 \left(\frac{2}{3} \cdot 2^{\lceil n/2 \rceil + 1} \right) = \log_2(2/3) + \lceil n/2 \rceil + 1$$

Since $\log_2(2/3) \approx -0.585$, this is approximately $\lceil n/2 \rceil + 0.4$, giving:

$$d(n) = \lfloor \log_2 a(n) \rfloor + 1 \approx \lceil n/2 \rceil + 1$$

For $n \geq 6$, this formula holds exactly, as can be verified computationally. □

Example 5.4 (Numerical Verification).

n	$a(n)$	$d(n)$	$b(n)$	$b(n)/d(n)$
6	42	6	3	0.500
7	85	7	4	0.571
8	170	8	4	0.500
9	341	9	5	0.556
10	682	10	5	0.500
11	1365	11	6	0.545
12	2730	12	6	0.500

As n increases, the ratio $b(n)/d(n)$ oscillates around 0.5, approaching it in the limit.

Teaching Note

The 50% duty cycle property is important in applications. In digital signal processing, a balanced signal (equal time high and low) minimizes DC bias and improves transmission characteristics. This theorem shows the Lichtenberg sequence is optimally balanced in this sense.

6 Additional Properties and Applications

6.1 Generating Function

Proposition 6.1 (Generating Function). *The ordinary generating function for the Lichtenberg sequence (starting from $a(1)$) is:*

$$G(x) = \sum_{n=1}^{\infty} a(n)x^n = \frac{x(1+x)}{(1-2x)(1-2x^2)}$$

Proof sketch. The alternating recurrence structure creates a generating function with factors $(1-2x)$ and $(1-2x^2)$ in the denominator, reflecting the doubling at even and odd indices. The numerator $x(1+x)$ accounts for initial conditions. A complete derivation requires manipulating the recurrence relations algebraically; we omit details here but refer interested readers to Stanley [5]. □

6.2 Modular Arithmetic

Proposition 6.2 (Modular Periodicity). *For prime modulus p, the sequence $\{a(n) \bmod p\}$ is eventually periodic with period dividing $2 \cdot \mathrm{ord}_p(2)$, where $\mathrm{ord}_p(2)$ is the multiplicative order of 2 modulo p.*

Modulus m	$a(n) \bmod m$ (first 12 terms)	Period
2	1,0,1,0,1,0,1,0,1,0,1,0	2
3	1,2,2,1,0,0,1,2,2,1,0,0	6
5	1,2,0,0,1,2,0,0,1,2,0,0	4
7	1,2,5,3,0,0,0,0,1,2,5,3	8

Teaching Note

Computing modular sequences is an excellent exercise for students learning modular arithmetic. Have them compute $a(n) \bmod 11$ or $a(n) \bmod 13$ to find the periods.

6.3 Reciprocal Ratios

Proposition 6.3 (Reciprocal Behavior). *The reciprocal ratios satisfy:*

1. $\dfrac{a(2n-1)}{a(2n)} = \dfrac{1}{2}$ *exactly for all* $n \geq 1$

2. $\dfrac{a(2n)}{a(2n+1)} = \dfrac{1}{2 + 1/a(2n)} \to \dfrac{1}{2}$ *as* $n \to \infty$

This follows immediately from Theorem 4.1 by taking reciprocals.

7 Applications to Computer Science and Engineering

7.1 Run-Length Limited Codes

In data storage and transmission, run-length limited (RLL) codes restrict the number of consecutive identical bits. The Lichtenberg sequence represents the extreme case: RLL(0,∞), where zero consecutive equal bits are allowed.

Why this matters:

- Prevents DC bias in signals
- Helps with clock recovery in digital circuits
- Reduces certain types of transmission errors
- Used in optical storage media (CDs, DVDs)

Example 7.1 (Practical Application). In CD encoding, the EFM (Eight-to-Fourteen Modulation) system uses constraints similar to RLL to ensure reliable readout. While not exactly the Lichtenberg sequence, the principle is the same: avoid long runs of equal bits.

7.2 Digital Signal Processing

The 50% duty cycle (Theorem 5.2) makes Lichtenberg numbers useful for:

- Clock signal generation: Balanced signals have minimal DC component
- Test patterns: Alternating bits stress-test transmission systems
- Synchronization: Frequent transitions aid clock recovery

7.3 Combinatorics

Proposition 7.2 (Combinatorial Interpretation)**.** *The Lichtenberg sequence counts binary strings of increasing length with no consecutive equal digits, where we consider strings starting with 1.*

More generally, for alphabet size k, the number of strings of length n avoiding consecutive equal symbols grows like $(k-1)^n$, but the Lichtenberg sequence represents a specific selection from these strings with additional structure.

8 Connections to Other Sequences

8.1 Jacobsthal Numbers

The Jacobsthal numbers (OEIS A001045) have a similar recurrence structure:

$$J(n) = J(n-1) + 2 \cdot J(n-2), \quad J(0) = 0, \quad J(1) = 1$$

giving: $0, 1, 1, 3, 5, 11, 21, 43, 85, 171, \ldots$

The closed form is $J(n) = (2^n - (-1)^n)/3$, compared to Lichtenberg's parity-dependent forms.

Note that while $a(8) = 170$ in the Lichtenberg sequence and $J(9) = 171$ in the Jacobsthal sequence appear similar, these are distinct sequences with different recurrence structures. Both involve doubling operations and factors of 3, but they exhibit fundamentally different growth patterns and binary properties.

8.2 Mersenne Numbers

The Mersenne numbers $M_n = 2^n - 1$ appear as the "forcing terms" in the single recurrence formulation (Definition 3.1):

$$L(n) + L(n-1) = 2^n - 1 = M_n$$

This connection reveals why powers of 2 (and hence binary representations) are so central to the sequence's structure.

9 Open Problems and Extensions

9.1 Unsolved Questions

Several interesting questions remain open:

1. **Prime Lichtenberg numbers:** For which n is $a(n)$ prime? Known primes occur at $n = 1, 2, 3, 5, 7, \ldots$ Are there infinitely many prime Lichtenberg numbers?

2. **Exact closed form:** Is there a closed form for $a(n)$ that doesn't depend on parity? Our formulas require separate cases for odd/even n.

3. **Distribution theory:** What is the distribution of $\{a(n)/2^{n/2}\}$ as $n \to \infty$? Does it converge? To what?

4. **Diophantine equations:** Characterize solutions to $a(n) = a(m) + a(k)$ for various combinations of indices.

5. **Base-k generalizations:** The single recurrence generalizes to $L_k(n) + L_k(n-1) = k^n - 1$ for base k. What is the full theory of these sequences?

6. **Extension to non-integer bases:** Does the structure generalize to bases like ϕ or e?

9.2 Suggestions for Student Projects

1. **Computational exploration:** Write a program to compute the first 1000 terms. Analyze statistical properties: distribution of digits, patterns in differences, etc.
2. **Prime investigation:** Search for prime Lichtenberg numbers. Are there heuristics for when $a(n)$ is likely to be prime?
3. **Base-3 sequence:** Study $L_3(n) + L_3(n-1) = 3^n - 1$ with $L_3(0) = 0$, $L_3(1) = 2$. Does it exhibit ternary alternation (no consecutive equal ternary digits)?
4. **Geometric interpretations:** Can the Lichtenberg sequence be visualized geometrically? Perhaps as a path on a graph or a tiling pattern?
5. **Connections to other areas:** Explore connections to Fibonacci numbers, Catalan numbers, or other classical sequences.

10 Summary and Conclusions

10.1 What We've Learned

This paper has developed a complete mathematical theory of the Lichtenberg sequence, establishing:

1. **Duality (Theorem 3.2):** Two apparently different recurrence formulations—alternating and single—generate the same sequence. This is a remarkable example of mathematical equivalence.
2. **Closed forms (Corollary 3.3):** Explicit formulas for odd and even indices in terms of powers of 2 and 4, revealing the exponential growth structure.
3. **Ratio convergence (Theorem 4.1):** Successive ratios alternate between exactly 2.0 and values exponentially converging to 2.0 with error $\sim 3/2^{n+2}$.
4. **Binary alternation (Theorem 5.1):** Rigorous proof that no consecutive equal binary digits occur, the defining property.
5. **50% duty cycle (Theorem 5.2):** In the limit, exactly half the bits are 1s, a property valuable for signal processing.
6. **Growth rate (Lemma 4.4):** Exponential growth $\sim \frac{2}{3} \cdot 2^{n/2}$, with effective base $\sqrt{2} \approx 1.414$.

10.2 Pedagogical Value

The Lichtenberg sequence is an ideal case study for teaching discrete mathematics because:

- It's simple enough to understand completely, yet rich enough to illustrate major techniques
- It connects algebra (recurrences), analysis (limits, asymptotics), number theory (modular arithmetic), and computer science (binary representations)
- It demonstrates the power of different viewpoints: recursive, closed-form, and generative
- It has real applications, showing mathematics is not just abstract theory
- It offers many directions for further exploration and student projects

10.3 Broader Lessons

Beyond the specific results, this study illustrates several broader mathematical principles:

1. **Multiple characterizations:** Mathematical objects often admit several equivalent definitions. Understanding their equivalence deepens insight.
2. **Pattern recognition:** Careful numerical investigation guides theoretical development. The ratio patterns we observed numerically became theorems.
3. **Proof techniques:** We used induction, closed forms, asymptotic analysis, and direct calculation—a full toolkit of discrete mathematics methods.
4. **Connections:** Seemingly simple sequences connect to deep areas: generating functions, modular arithmetic, number theory.
5. **Rigor matters:** We didn't just observe patterns; we proved them. Mathematical certainty requires proof.

Exercises for Students

Computational Exercises ($\star$)

1. Compute $a(n)$ for $n = 1$ to 20 using both the alternating recurrence and the single recurrence. Verify they give the same values.
2. Write a program to compute $a(n)$ efficiently for large n using the closed forms. Test it on $n = 100, 200, 500$.
3. Generate the sequence $\{a(n) \bmod 11\}$ for $n = 1$ to 50. Find the period.
4. Plot $\log_2 a(n)$ versus n for $n = 1$ to 100. Verify the slope is approximately 1/2.

Theoretical Exercises ($\star\star$)

5. Prove that $a(2n-1)$ is always odd and $a(2n)$ is always even for $n \geq 1$.
6. Show that $\gcd(a(n), a(n+1)) = 1$ for all $n \geq 1$ (consecutive terms are coprime).
7. Derive the generating function (Proposition 6.1) rigorously from the recurrence relations.
8. Prove that $a(n) \equiv (-1)^{n+1} \pmod 3$ for $n \geq 1$.
9. Show that $a(n+2) = 4 \cdot a(n) + r(n)$ where $r(n) \in \{2, 3\}$. Characterize when $r(n) = 2$ versus $r(n) = 3$.

Challenge Problems ($\star\star\star$)

10. Find all n such that $a(n)$ is a perfect square.
11. Prove or disprove: There are infinitely many primes in the Lichtenberg sequence.
12. Characterize all solutions to $a(n) = F_m$ where F_m is the m-th Fibonacci number.
13. Study the base-3 sequence $L_3(n)+L_3(n-1) = 3^n-1$. Prove it has ternary digit alternation analogous to Theorem 5.1.

Acknowledgments

The author expresses gratitude to Dr. Jean-Claude Perez for his pioneering work on Fibonacci structures in DNA sequences and his "Perez Hourglass" geometric framework, which provided conceptual inspiration for investigating dual characterizations of mathematical sequences. Perez's 2024 work on geometric interpretations of the Lichtenberg sequence motivated the author's systematic exploration of the duality between alternating and single recurrence formulations presented in Section 3.

This research was conducted with computational assistance from Claude AI (Anthropic, 2024) for verification, pattern recognition, and proof development. The collaboration between human mathematical intuition and artificial intelligence capabilities proved invaluable for ensuring computational accuracy and exploring convergence properties.

The author thanks educators and students who will use this material, and welcomes feedback for improving its pedagogical effectiveness.

Data Availability

All Python verification scripts, computational data, and supplementary materials are available upon request from the author at `gauntletin@gmail.com`.

References

[1] Sloane, N. J. A. (2024). *The On-Line Encyclopedia of Integer Sequences*. Sequence A000975. `https://oeis.org/A000975`

[2] Perez, J.-C. (2024). *Geometric Interpretations of the Lichtenberg Sequence: The "Perez Hourglass" Framework*. Zenodo. `https://doi.org/10.5281/zenodo.17576956`

[3] Lichtenberg, G. C. (1769). Early observations of electrical discharge patterns.

[4] Graham, R. L., Knuth, D. E., & Patashnik, O. (1994). *Concrete Mathematics: A Foundation for Computer Science* (2nd ed.). Addison-Wesley.

[5] Stanley, R. P. (2012). *Enumerative Combinatorics, Vol. 1* (2nd ed.). Cambridge University Press.

[6] Wilf, H. S. (1994). *Generatingfunctionology* (2nd ed.). Academic Press.

[7] Flajolet, P., & Sedgewick, R. (2009). *Analytic Combinatorics*. Cambridge University Press.

[8] Anthropic (2024). Claude AI (Sonnet 4.5). `https://www.anthropic.com`

A Python Code for Verification

Here is sample Python code for computing and verifying properties of the Lichtenberg sequence:

```
def lichtenberg_alternating(n):
    """Compute using alternating recurrence"""
    if n == 0: return 0
    if n == 1: return 1
    a = [0, 1]
    for i in range(2, n+1):
        if i % 2 == 0:
            a.append(2 * a[i-1])
        else:
            a.append(2 * a[i-1] + 1)
    return a[n]

def lichtenberg_single(n):
    """Compute using single recurrence"""
    if n == 0: return 0
    if n == 1: return 1
    L = [0, 1]
    for i in range(2, n+1):
        L.append(2**i - 1 - L[i-1])
    return L[n]

def lichtenberg_closed_form(n):
    """Compute using closed forms"""
    if n == 0: return 0
    if n % 2 == 1:  # odd index
        k = (n + 1) // 2
        return (4**k - 1) // 3
    else:  # even index
        k = n // 2
        return 2 * (4**k - 1) // 3

def verify_binary_alternation(n):
    """Check if a(n) has alternating binary representation"""
    a_n = lichtenberg_alternating(n)
    binary = bin(a_n)[2:]  # Remove '0b' prefix
    for i in range(len(binary) - 1):
        if binary[i] == binary[i+1]:
            return False
    return True

# Verification
print("Verifying duality for n = 0 to 20:")
for n in range(21):
    alt = lichtenberg_alternating(n)
    single = lichtenberg_single(n)
    closed = lichtenberg_closed_form(n)
    assert alt == single == closed, f"Mismatch at n={n}"
print("  All three formulations agree!")

print("\nVerifying binary alternation for n = 1 to 20:")
for n in range(1, 21):
    assert verify_binary_alternation(n), f"No alternation at n={n}"
print("  Perfect binary alternation confirmed!")

print("\nRatio convergence:")
```

```
for n in range(1, 11):
    a_2n = lichtenberg_alternating(2*n)
    a_2n_plus_1 = lichtenberg_alternating(2*n + 1)
    ratio = a_2n_plus_1 / a_2n
    error = ratio - 2.0
    print(f"n={n}: a({2*n+1})/a({2*n}) = {ratio:.6f}, "
          f"error = {error:.6f}")
```