On the binary digits of n and n^2

Karam Aloui, Damien Jamet, Hajime Kaneko, Steffen Kopecki, Pierre Popoli, Thomas Stoll

ON THE BINARY DIGITS OF n AND n^2

KARAM ALOUI, DAMIEN JAMET, HAJIME KANEKO, STEFFEN KOPECKI, PIERRE POPOLI, AND THOMAS STOLL

Abstract. Let $s(n)$ denote the sum of digits in the binary expansion of the integer n. Hare, Laishram and Stoll (2011) studied the number of odd integers such that $s(n) = s(n^2) = k$, for a given integer $k \geq 1$. The remaining cases that could not be treated by theses authors were $k \in \{9, 10, 11, 14, 15\}$. In this paper we show that there is only a finite number of solutions for $k \in \{9, 10, 11\}$ and comment on the difficulties to settle the two remaining cases $k \in \{14, 15\}$. A related problem is to study the solutions of $s(n^2) = 4$ for odd integers. Bennett, Bugeaud and Mignotte (2012) proved that there are only finitely many solutions and conjectured that $n = 13, 15, 47, 111$ are the only solutions. In this paper, we give an algorithm to find all solutions with fixed sum of digits value, supporting this conjecture, as well as show related results for $s(n^2) = 5$.

1. Introduction

Let $s(n)$ be the sum of digits in the binary expansion (i.e. the Hamming weight) of $n \in \mathbb{N}$. In the present paper we investigate the question of whether or not the equation

$$s(n) = s(n^2) = k \tag{1}$$

has infinitely many odd solutions in n for a given $k \in \mathbb{N}$.[1] Hare, Laishram and Stoll [9] settled all cases with the exception of $k \in \{9, 10, 11, 14, 15\}$. Our contribution here is to show, via a combinatorial and algorithmic approach, that the equation only has finitely many solutions for $k \in \{9, 10, 11\}$. We will address the computational issues that we encounter for the last remaining open cases, namely, $k = 14, 15$.

The main motivation to consider (1) comes from work of Madritsch and Stoll [13] who showed that $(s(n^2)/s(n))_{n\geq 1}$ is dense in $\mathbb{R}^+$. This elaborates on an old result of Stolarsky [18] (see also [8, 11, 15, 14, 17]) who showed that $\liminf_{n\to\infty} s(n^2)/s(n) = 0$. Since the average size of $s(n^2)$ is twice as large as that of $s(n)$ (see [1, 16]) the equation (1) concerns an exceptional set of integers. In particular, it is intriguing that for certain values of k the equation allows for infinitely many odd solutions n and for other values of k there is just a finite number. One of the results of Hare, Laishram and Stoll [9] states that there are infinite parametric families of solutions for $k = 12, 13$ and $k \geq 16$. They showed that

$$s(n) = s(n^2) = 12, \text{ for all } n = 111 \cdot 2^t + 111, \text{ with } t \geq 15, \tag{2}$$

$$s(n) = s(n^2) = 13, \text{ for all } n = 23 \cdot 2^t + 1471, \text{ with } t \geq 21, \tag{3}$$

$$s(n) = s(n^2) = 16, \text{ for all } n = 111 \cdot 2^t + 1919, \text{ with } t \geq 21. \tag{4}$$

On the other side of the spectrum, there are only finitely many solutions for $k \leq 8$. For example, for

$$s(n) = s(n^2) = 8,$$

there are only 64 solutions in odd integers and the largest solution is $n = 266335$ (see [9, Table 2]). These results are based on an algorithm that handles all the possible orderings of the exponents

Key words and phrases. Digital expansions; numeration system; sum of digits function; sequences and sets.

[1]Note that for all integers n we have $s(2n) = s(n)$. This means that the restriction to n odd is necessary to make this question meaningful.

in n^2 when n is written as a sum of a small number of powers of 2. Since the algorithm treats (in an exhaustive way) all cases, the method of [9] allowed to explicitly determine all the solutions for $k \leq 8$. The running time of the algorithm, however, explodes for larger values of k. Some heuristic arguments are given in [9, Section 5] to support the conjecture that there are only finitely many solutions for $k \in \{9, 10\}$. The main purpose of this paper is to combine a new combinatorial factorization lemma and two algorithms with recent results by Kaneko and Stoll [10] to reduce the investigation to a finite case analysis. We then carry out this case analysis for $k \in \{9, 10, 11\}$ in a unified manner.

2. Main results

In the present article we show the following theorem:

Theorem 1. *Let $k \in \{9, 10, 11\}$. Then the number of odd integers n with*

$$s(n) = s(n^2) = k$$

is finite.

We have made a global search for $s(n^2) = s(n) = k$, $11 \leq k \leq 15$, up to $n < 2^{80}$ (see Section 7). No infinite family occurs clearly in the cases $k = 14$ and $k = 15$ compared to $k \in \{12, 13, 16\}$, see (2)–(4). We therefore formulate the following conjecture:

Conjecture 1. Let $k \in \{14, 15\}$. Then the number of odd integers n with $s(n) = s(n^2) = k$ is finite.

For Theorem 1 it is crucial to have efficient algorithms at our disposal that calculate certain sets that appear in intermediate steps in the proof.

For fixed $\ell_1 \geq 1$, $\ell_2 \geq 1$ and $m \geq 1$, set

$$\Delta_{\ell_1,\ell_2,m} := \{n \in \mathbb{N} : \quad s(n) = \ell_1, \quad s(n^2) \leq \ell_2, \quad n < 2^m, \quad n \text{ odd}\}. \tag{5}$$

Our first algorithm, called `next`, has the purpose to calculate $\Delta_{\ell_1,\ell_2,m}$ for small values of ℓ_1, ℓ_2, m.

For fixed $k \in \mathbb{N}$ and $\lambda \geq 1$, let

$$E_{k,\lambda} := \{n \in \mathbb{N} : \quad s(n^2) = k, \quad s(n) = \lambda, \quad n \text{ odd}\}, \tag{6}$$

and set

$$E_k := \bigcup_{\lambda \geq 1} E_{k,\lambda}. \tag{7}$$

The aim of the second algorithm, called `max-integer`, is to calculate efficiently $E_{k,\lambda}$ for small k and λ. Several results about the sets E_k are already known in the literature. To begin with, it is an elementary calculation to show that $E_2 = \{3\}$. Szalay [19] showed that

$$E_3 = \{2^t + 1 : \ t \geq 2\} \cup \{7, 23\}, \tag{8}$$

see also [12] for a generalization. Szalay's proof relies on a result of Beukers on the Ramanujan–Nagell equation. Typically such sets are composed of a union of a set of infinite parametrized integers and a set of small sporadic solutions. Bennett/Bugeaud/Mignotte [4], Bennett/Bugeaud [3], Hajdu/Pink [7] and Bérczes/Hajdu/Miyazaki/Pink [5] generalized Szalay's results to other bases and more general powers, see also Bennett [2]. Finally, we mention also the recent work of Szalay [20] who considered algorithms to find the solution set of the Diophantine equation $2^n + \alpha \cdot 2^m + \alpha^2 = x^2$, where α is a fixed positive integer.

As for E_4, it is known that it is a finite set (so, no infinite families occur), see Bennett, Bugeaud and Mignotte [4], and Corvaja and Zannier [6]. Bennett, Bugeaud and Mignotte conjectured:

Conjecture 2.

$$E_4 = \{13,\ 15,\ 47,\ 111\}. \tag{9}$$

This conjecture remains still open.

We here consider a refined version of Conjecture 2, namely, we restrict our attention to those integers n that have a fixed sum of binary digits. This is particularly valuable in the study of (1), where we need to know explicitly $E_{4,\lambda}$ and $E_{5,\lambda}$ for small values of λ. Indeed, the infinite family given in (3) is built from the two integers 23 and 1471. Since $s(23^2) = 3$, $s(1471^2) = 5$ and $s(23 \cdot 1471) = 5$, we have the correct amount of bits in the square of $n = 23 \cdot 2^t + 1471$ for sufficiently large t.

We apply `max-integer` to show the following result.

Theorem 2. *We have*

$$\bigcup_{\lambda \le 17} E_{4,\lambda} = \{13,\ 15,\ 47,\ 111\} \tag{10}$$

and

$$\bigcup_{4 \le \lambda \le 15} E_{5,\lambda} = \{29,\ 31,\ 51,\ 79,\ 91,\ 95,\ 157,\ 223,\ 279,\ 479,\ 727,\ 1471,\ 5793\}. \tag{11}$$

Moreover,

$$E_5 = \{1 + 2 + 2^\ell :\ \ell \ge 3\} \cup \{1 + 2^\ell + 2^{\ell+1} :\ \ell \ge 3\} \cup \{1 + 2^\ell + 2^{2\ell-1} :\ \ell \ge 3\} \cup E', \tag{12}$$

where E' is a finite set.

This theorem gives more evidence on (9): all $E_{4,\lambda}$ are empty sets for $5 \le \lambda \le 17$. The result might also point towards a possible computational proof if we could establish a universal bound for λ. For (12) we conjecture:

Conjecture 3.

$$E' = \{29,\ 31,\ 51,\ 79,\ 91,\ 95,\ 157,\ 223,\ 279,\ 479,\ 727,\ 1471,\ 5793\}.$$

This is supported by the fact that the largest weight of an integer in E' occurs for $n = 1471$, namely $s(1471) = 9$, so that the sets $E_{5,\lambda}$ are all empty for $10 \le \lambda \le 15$.

The paper is structured as follows. In Section 3, we state, collect and prove some related auxiliary results; in particular, we give a new combinatorial factorization lemma that is at the core of our method. Section 4 is devoted to the proof of Theorem 1, where we make use of the algorithm `next`. In Section 5, we give the proof of Theorem 2 where we rely on the algorithm `max-integer`. We postpone, for an easier readability, the detailed description of the two algorithms and their implementation to Section 6. We finally conclude in Section 7 with some remarks on the remaining cases of (1), i.e. the cases $k = 14$ and $k = 15$, that remain unsettled.

3. Preliminaries

Let $n = \sum_{i=0}^{\ell} \varepsilon_i 2^i$ with $\varepsilon_i \in \{0, 1\}$ and $\varepsilon_\ell = 1$. We write $(n)_2$ to refer to the binary expansion of the integer n. Recall that $s(n) = \sum_{i=0}^{\ell} \varepsilon_i$ is the sum of digits of n. We use the letter $\mathtt{x}$ (with or without indices) to denote binary blocks that always end in $\mathbf{1}$, so that they correspond to the binary expansions of odd integers. In the language of combinatorics on words, $\mathtt{x}$ will be a non-empty word over the alphabet $\{\mathbf{0}, \mathbf{1}\}$ (the "bits") whose rightmost symbol is $\mathbf{1}$. We always use boldface to talk about bits. To keep notation as simple and readable as possible, we will use x (note the change

in the font) to denote the associated odd integer. For example, for $\mathsf{x} = \mathbf{11}$ we have $x = 3$; also, we write 1 for the integer "one", and **1** for the one-bit. Note that $\ell = \ell((n)_2)$ is the length of the binary expansion of the integer n. Again, for simplicity reasons, we also use $\ell(n)$ for $\ell((n)_2)$. As usual, in combinatorics on words, we will write xy for the concatenation of the binary words x and y, and x^ℓ for the ℓ-fold concatenation of the word x. Moreover, we write $|\mathsf{x}|$ for the length of the word x. On the other hand, xy, or $x \cdot y$, will systematically denote the multiplication of the integers x and y.

For an odd integer n such that $s(n) = k$, we consider a decomposition of its binary expansion in the form

$$(n)_2 = \mathsf{x}_m \mathbf{0}^{\ell_m} \mathsf{x}_{m-1} \cdots \mathsf{x}_1 \mathbf{0}^{\ell_1} \mathsf{x}_0 \tag{13}$$

into $m < k$ blocks of **0**-bits of length ℓ_i, for $1 \le i \le m$, which separate x_i, for $0 \le i \le m$. Recall that all x_i end in **1**-bits, so that they correspond to the binary expansion of odd integers. Note that the decomposition (13) is not unique since we can merge or split inner blocks to obtain other factorizations. Let us denote $y_{i,j} = x_i \cdot x_j$ for $0 \le i, j \le m$, and $\mathsf{y}_{i,j}$ the associated binary blocks that will compose n^2. Note that $\mathsf{y}_{i,j}$ again ends in a **1**-bit. Let

$$\hat{\ell_j} = \sum_{i=1}^{j} (\ell_i + \ell(x_{i-1}))$$

for $0 \le j \le m$, which represents the length of $\mathbf{0}^{\ell_j} \mathsf{x}_{j-1} \cdots \mathsf{x}_1 \mathbf{0}^{\ell_1} \mathsf{x}_0$. Furthermore let $\ell_{i,i} = 2\hat{\ell_i}$ for $0 \le i \le m$ and $\ell_{i,j} = \hat{\ell_i} + \hat{\ell_j} + 1$ for $0 \le i, j \le m$ and $i \ne j$. Then the square n^2 is the sum of the integers

$$u_{i,i} = 2^{\ell i,i} y_{i,i} \quad \text{for } 0 \le i \le m \text{ and}$$
$$u_{i,j} = 2^{\ell i,j} y_{i,j} \quad \text{for } 0 \le i < j \le m.$$

We say that the summand $y_{i,j}$ *interferes* with the summand $y_{i',j'}$ if, in the addition of the two terms written in binary, a carry propagation caused by $y_{i,j}$ reaches a binary bit of $y_{i',j'}$, or vice-versa (we take liberty to say also, that $\mathsf{y}_{i,j}$ interferes with $\mathsf{y}_{i',j'}$). We will frequently discuss the situation on how many **1**-bits remain in the addition of interfering terms. We will reject possibilities when the additions lead to numbers with too many **1**-bits. If blocks are *non-interfering* then the number of **1**-bits of the their sum is the sum of the **1**-bits of the summands. Let us explain the procedure with an example. If we add **111** or its shifts to **11100001**, we observe that it is impossible to find shifts in a way that the sum of the two summands gives a single **1**-bit:

$$\begin{array}{rc} & \mathbf{11100001} \\ + & \longleftarrow \mathbf{111} \longrightarrow \end{array}$$

This procedure can be easily implemented: it is sufficient to test via one `for`-loop. If more than two terms are added together, then more `for`-loops will do the job.

The next lemma gives a sufficient condition for non-interference between two summands.

Lemma 1. *Let $y_{i,j}$, $y_{i',j'}$ be two summands defined as before. If $\ell_{i,j} \ge \ell_{i',j'} + \ell(y_{i',j'}) + k^2$, then $y_{i,j}$ does not interfere with $y_{i',j'}$.*

Proof. If these two summands interfered, then there would be at least k^2 **1**-bits involved in the carry propagation from $y_{i',j'}$ to $y_{i,j}$. But the number of **1**-bits in all summands is at most $k + \frac{k(k-1)}{2} < k^2$. □

Let $\ell_{\min} = \min_{1 \le i \le m}(\ell_i)$ and $\ell_{\max} = \max_{0 \le i \le m} \ell(x_i)$. By Lemma 1 we can deduce that if $\ell_{\min} > 2\ell_{\max} + k^2$, then two summands $y_{i,j}$, $y_{i',j'}$ do not interfere if $i > i'$ and $j \ge j'$.

Lemma 2 (Factorization lemma)**.** *For $k \geq 1$ there is a bound N_k such that the binary expansion of every odd $n \geq N_k$ that satisfies $s(n) = s(n^2) = k$ can be factorized as*

$$(n)_2 = \mathsf{x}_m \mathbf{0}^{\ell_m} \mathsf{x}_{m-1} \cdots \mathsf{x}_1 \mathbf{0}^{\ell_1} \mathsf{x}_0 \tag{14}$$

where $1 \leq m < k$, $\mathsf{x}_0, \ldots, \mathsf{x}_m$ are the binary words corresponding to the binary expansions of odd integers and $\ell_1, \ldots, \ell_m \in \mathbb{N}$ such that $\ell_{\min} > 2x_{\max} + k^2$ where $\ell_{\min} = \min_{1 \leq i \leq m}(\ell_i)$ and $x_{\max} = \max_{0 \leq i \leq m} |\mathsf{x}_i|$.

Proof. Let $f(i) = 4i + k^2$, let $N_k = 2^{f^k(1)}$ where $f^k(1)$ is the k-fold composition of f evaluated at 1. Consider an odd integer $n \geq N_k$ such that $s(n) = k$. If the binary expansion of n contains $k-1$ blocks of **0**-bits and if each of these **0**-blocks is longer than k^2+2, then each **1**-bit in the expansion forms one of the x_i with $m = k-1$ and we are done. Otherwise, we combine all **0**-blocks which have a length at most k^2+2 with its bordering **1**-bits and make this one of the factors x_i. If all remaining **0**-blocks are longer than $2(k^2+4)+k^2$, we find a suitable factorization of $(n)_2$ with $m = k-2$. Otherwise, we continue inductively and obtain $n > 2^{f^k(1)}$ has a desired factorization. □

We will also need some elementary results on multiples of 3 with few non-zero digits.

Lemma 3. *Let n be an odd integer with $s(3n) = 2$. Then $(n)_2 \in \{(\mathbf{10})^\ell \mathbf{11} : \ \ell \geq 0\} \cup \{\mathbf{1}\}$.*

Proof. For $n \geq 5$ we set $(n)_2 = \mathbf{1}\varepsilon_d\varepsilon_{d-1}\cdots\varepsilon_0\mathbf{1}$ ($\varepsilon_i \in \{\mathbf{0}, \mathbf{1}\}, d \geq 0$) and observe that the usual addition $2n + n$ translates into

$$\begin{array}{cccccccc} & \mathbf{1} & \varepsilon_d & \varepsilon_{d-1} & \cdots & \varepsilon_1 & \varepsilon_0 & \mathbf{1} \\ + & & \mathbf{1} & \varepsilon_d & \varepsilon_{d-1} & \cdots & \varepsilon_1 & \varepsilon_0 \ \mathbf{1}. \end{array}$$

Since the last **1**-bit will stay after the addition, the addition of the penultimate **1** to ε_0 must give rise to a carry that propagates up to the highest significant digits. The only way to make this happen without creating additional **1**-bits in the sum is $\varepsilon_0 = \varepsilon_2 = \varepsilon_4 = \cdots = \mathbf{1}$ and $\varepsilon_1 = \varepsilon_3 = \varepsilon_5 = \cdots = \mathbf{0}$. □

Lemma 4. *Let n be an odd integer with $s(3n) = 4$. Then $(n)_2 = \mathsf{x}_1\mathbf{0}^s\mathsf{x}_0$ for some $s \geq 2$ with*

$$\mathsf{x}_1, \mathsf{x}_0 \in \{(\mathbf{10})^\ell \mathbf{11} : \ \ell \geq 1\} \cup \{\mathbf{1}\},$$

or $n \leq 2^{2s(n)-1}$.

Proof. If there is a block of **0**'s of length ≥ 2 inside $(n)_2$, then in the addition of $2n + n = 3n$ there are non-interfering terms and the additions have to amount for 2 bits in the sum of the corresponding portions. Lemma 3 shows that the only possibilities are the blocks $(\mathbf{10})^\ell\mathbf{11}$ for some $\ell \geq 0$, and the block consisting of a single **1**. This gives the first part in the statement. If there is no block of consecutive **0**'s of length ≥ 2 then x is evidently bounded by $2^{2s(n)} - 1$. □

Lemma 5. *Let n be an odd integer with $s(3n) = 3$. Then*

$$(n)_2 \in \{\mathbf{1}(\mathbf{01})^{\ell_1}(\mathbf{10})^{\ell_2}\mathbf{11} : \ \ell_1, \ell_2 \geq 0\}$$

Proof. Recall the reasoning of the proof of Lemma 3 first. A possible carry has to propagate all the way up to the highest significant digits in order to generate only **0**-bits except the lowest significant bit and the highest significant bit. In the former proof, this implied an alternation of **0**-bits and **1**-bits in the middle part. In the statement of the present lemma, since we want three **1**-bits in the resulting sum, we need to break this alternation at least once. We therefore have the following addition:

$$\begin{array}{rcccccccccccc} & \mathbf{1} & \varepsilon_d & \cdots & \varepsilon_k & \mathbf{1} & \underline{\mathbf{1}} & \mathbf{0} & \cdots & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ + & & \mathbf{1} & \varepsilon_d & \cdots & \varepsilon_k & \underline{\mathbf{1}} & \mathbf{1} & \mathbf{0} & \cdots & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1}. \end{array}$$

The lowest significant **1**-bit will stay after the summation of the two numbers (it does not interact with the other bits). In the overlapping **1**-bits at the breaking point (underlined in the above addition scheme) there will remain one **1**-bit in the sum. The addition of these bits generates a carry that has to generate only **0**-bits up to the highest significant **1**-bit. The only way to achieve this is again to alternate the **0**-bits and **1**-bits. This directly translates into the given form. □

Lemma 6. *If* $(n)_2 \in \{(\mathbf{10})^\ell\mathbf{11} : \ \ell \geq 2\}$ *then* $s(n^2) \geq 7$.

Proof. An elementary calculation shows that, if $(n)_2 = (\mathbf{10})^\ell\mathbf{11}$, $\ell \geq 1$, then

$$(n^2)_2 = \begin{cases} (\mathbf{111000})^{j-1}\mathbf{111001}(\mathbf{000111})^j\mathbf{001} & \text{if } \ell = 3j \text{ for some } j \geq 1, \\ (\mathbf{111000})^j\mathbf{1111}(\mathbf{000111})^j\mathbf{001} & \text{if } \ell = 3j+1 \text{ for some } j \geq 0, \\ (\mathbf{111000})^j\mathbf{11100111}(\mathbf{000111})^j\mathbf{001} & \text{if } \ell = 3j+2 \text{ for some } j \geq 0. \end{cases}$$

□

In our application for the infinite family in Lemma 4, we will fix the value of $s(n)$, which means that ℓ is small. Lemma 6 then guarantees that the squares of such integers have (too) many **1**-bits, which will lead to a contradiction. We will make use of this procedure at several places in our investigation, in particular to check that there are no solutions in odd integers n for the system $s(3n) = 4$, $s(n) = 9$ and $s(n^2) = 5$. The sporadic solutions that are bounded in Lemma 4 can be checked directly by an exhaustive computer search.

We next recall two recent results by Kaneko and Stoll [10] that deal with products of integers with few binary digits.

Lemma 7. *Let* $\ell, m \geq 2$*, and let* a *and* b *be two odd integers such that* $s(a) = \ell$ *and* $s(b) = m$*. If* $s(ab) = 2$*, then we have*

$$ab < 2^{2\ell m - 4}.$$

Lemma 8. *Let* $\ell, m \geq 2$*, and* a *and* b *be two odd integers such that* $s(a) = \ell, s(b) = m$ *and* $m\ell \geq 5$*. If* $s(ab) = 3$*, then we have*

$$ab < 2^{4\ell m - 13}.$$

We will use these results when we look for solutions in the form $\mathsf{x}_1\mathbf{0}\cdots\mathbf{0}\mathsf{x}_0$ for a large inner block of **0**-bits. For such a structure, we have three separated contributions to the binary decomposition in the square: x_1^2, x_0^2 and the double product $x_1 \cdot x_0$ which do not interfere since they are well-separated. When $s(x_1x_0) = 2$ or $s(x_1x_0) = 3$, we can apply these two lemmas to bound x_1 and x_0, and an exhaustive search will then be sufficient to conclude. We mention that the direct analogue to Lemma 7 and Lemma 8 for $s(ab) = 4$ does not hold true (see [10]).

4. Proof of Theorem 1

According to Lemma 2, if there exist infinitely many odd solutions of (1) for some k, then almost all (i.e. all with a finite number of exceptions) of the binary expansions of these solutions can be factorized. Consider a factorization of $(n)_2$ as stated in Lemma 2 and note that none of the $2m+1$ summands in the set $\{y_{m,i}\}_{i=0}^m \cup \{y_{i,0}\}_{i=0}^m$ interfere with each other. Some of these summands may interfere with other summands, yet, even in that cases, each contributes with at least one **1**-bit to the binary expansion of n^2. Thus, if $m \geq k/2$, then $s(n^2) \geq 2m+1 > k$ and n cannot be a solution of (1). We therefore can safely suppose that $1 \leq m < k/2$.

We have the corresponding graphs that show the various possibilities of interference. Herein, vertices are the summands and the edges correspond to possible instances of interference between summands.

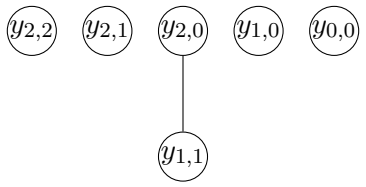

FIGURE 1. Interference graph for $m = 2$.

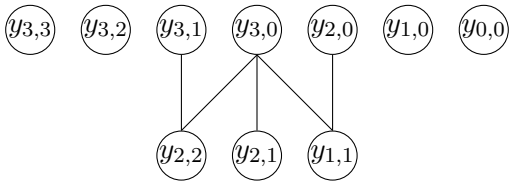

FIGURE 2. Interference graph for $m = 3$.

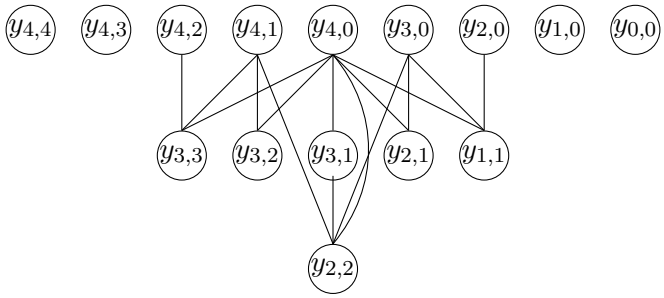

FIGURE 3. Interference graph for $m = 4$.

As a consequence of Lemma 2, the following result implies Theorem 1.

Lemma 9. *Let $k \in \{9, 10, 11\}$ and $n \geq N_k$ be a sufficiently large odd integer with*

$$s(n) = s(n^2) = k.$$

Then there exists no factorization of $(n)_2$ in the form (14).

For $\boxed{k = 9}$ and $\boxed{k = 10}$ we have $1 \leq m \leq 4$, and for $\boxed{k = 11}$ we have $1 \leq m \leq 5$. The proof of Lemma 9 is based on elementary considerations but we need to combine several ingredients to conclude: Szalay's result on E_3, Kaneko and Stoll's results on products of integers with 2 or 3 digits, non-existence of certain squares modulo powers of 2, analysis of possible interference for multiple blocks, the tables of Hare, Laishram and Stoll [9] etc. The investigation results in a finite case analysis where the details depend on the parameters. For the convenience of the reader, we have arranged the proof for fixed m since the reasoning is very similar for $k \in \{9, 10, 11\}$ when the interference graph stays the same. We use freely our algorithms `next` and `max-integer`, whose descriptions are postponed to Section 6.

4.1. **The case $m = 1$.** We have $(n)_2 = \mathtt{x}_1\mathbf{0}\cdots\mathbf{0}\mathtt{x}_0$ with a (long) contiguous inner block of **0**-bits. First, we observe that if $x_0 = 1$ (recall that this is the same as saying that $\mathtt{x}_0 = \mathbf{1}$), then n cannot be a solution of (1) since then we would have $s(n^2) = s(x_1^2) + s(x_1) + 1 = s(x_1^2) + (k-1) + 1 > k$. Similarly, we have $x_1 \neq 1$. By symmetry we can suppose that $s(x_1) \geq s(x_0)$. We therefore have to solve the following system:

$$\text{(15)} \qquad \begin{cases} s(x_1) + s(x_0) = k, \\ s(x_1^2) + s(x_1 x_0) + s(x_0^2) = k, \\ s(x_1),\ s(x_0),\ s(x_1^2),\ s(x_0^2),\ s(x_1 \cdot x_0) \geq 2, \\ s(x_1) \geq s(x_0). \end{cases}$$

We distinguish the cases according to the value of $s(x_1 \cdot x_0) \in \{2, \ldots, k-4\}$.

(1) $\underline{s(x_1 \cdot x_0) = 2.}$ Here, Lemma 7 provides an upper bound for x_1 and x_0. Recall the definition of $\Delta_{\ell_1,\ell_2,m}$ given in (5). The following table lists all sets to check for possible solutions (x_1, x_0) of (15):

Sets Δ for $k = 9$	Sets Δ for $k = 10$	Sets Δ for $k = 11$
$\Delta_{5,5,36} \times \Delta_{4,5,36}$	$\Delta_{5,6,46} \times \Delta_{5,6,46}$	$\Delta_{6,7,56} \times \Delta_{5,7,56}$
$\Delta_{6,5,36} \times \Delta_{3,5,36}$	$\Delta_{6,6,46} \times \Delta_{4,6,46}$	$\Delta_{7,7,56} \times \Delta_{4,7,56}$
$\Delta_{7,5,36} \times \Delta_{2,5,36}$	$\Delta_{7,6,46} \times \Delta_{3,6,46}$	$\Delta_{8,7,56} \times \Delta_{3,7,56}$
-	$\Delta_{8,6,46} \times \Delta_{2,6,46}$	$\Delta_{9,7,56} \times \Delta_{2,7,56}$

We construct these sets with our algorithm `next` in an efficient manner and check whether it gives a solution to the system (15); there is no solution. We could also use the tables of [9] for the cases $\boxed{k = 9}$ and $\boxed{k = 10}$, however, for the case $\boxed{k = 11}$, the tables of [9] do not allow to conclude and we need a new method to construct the related sets.

(2) $\underline{s(x_1 \cdot x_0) = 3.}$ With the help of Lemma 8 we can, similarly to before, restrict our attention to a finite number of sets. These sets are given in the following table:

Sets Δ for $k=9$	Sets Δ for $k=10$	Sets Δ for $k=11$
$\Delta_{5,4,67} \times \Delta_{4,4,67}$	$\Delta_{5,5,87} \times \Delta_{5,5,87}$	$\Delta_{6,6,107} \times \Delta_{5,6,107}$
$\Delta_{6,4,67} \times \Delta_{3,4,67}$	$\Delta_{6,5,87} \times \Delta_{4,5,87}$	$\Delta_{7,6,99} \times \Delta_{4,6,99}$
$\Delta_{7,4,67} \times \Delta_{2,4,67}$	$\Delta_{7,5,87} \times \Delta_{3,5,87}$	$\Delta_{8,6,83} \times \Delta_{3,6,83}$
-	$\Delta_{8,5,87} \times \Delta_{2,5,87}$	$\Delta_{9,6,59} \times \Delta_{2,6,59}$

In the last column we have already reduced the number of cases to consider in order to speed up the calculations. Again, as before, there is no solution.

(3) $\underline{s(x_1 \cdot x_0) = 4.}$ We here investigate the weight of the square parts since there is no universal bound on $x_1 \cdot x_0$.

- Let $\boxed{k=9}$. Then we have $\max(s(x_1^2), s(x_0^2)) = 3$ and $\min(s(x_1^2), s(x_0^2)) = 2$. Recall the definitions of $E_{k,\lambda}$ and E_k in (6) and (7), and Szalay's result (8). We have $x_1 \in E_3$ and $x_0 = 3$, or $x_0 \in E_3$ and $x_1 = 3$. Then $s(x_1) + s(x_0) \le 6 < 9$, which is a contradiction.
- Let $\boxed{k=10}$.
  (a) If $s(x_1^2) = 4$ then we have $s(x_0^2) = 2$ and $x_0 = 3$. Thus $s(x_1) = 8$ and there is no such x_1, see Table 3 in [9].
  (b) If $s(x_1^2) = 3$ then $x_0, x_1 \in E_3$ and we have $s(x_1) + s(x_0) \le 8 < 10$.
  (c) If $s(x_1^2) = 2$ then $s(x_1) = 2$ and $s(x_0) = 8$. We conclude as in case (a).
- Let $\boxed{k=11}$.
  (a) If $s(x_1^2) = 5$ then we have $s(x_0^2) = 2$ and $x_0 = 3$. Thus x_1 satisfies

$$s(3x_1) = 4, \qquad s(x_1) = 9. \tag{16}$$

  A machine calculation shows that there is no solution of (16) with $x_1 \le 2^{17}$ that also satisfies $s(x_1^2) = 5$. Lemma 4 now states that if there were a solution of (16) with $x_1 > 2^{17}$ then

$$\mathsf{x}_1 \in \{\mathbf{10}^\alpha(\mathbf{10})^6\mathbf{11} : \ \alpha \ge 1\} \ \cup \ \{(\mathbf{10})^6\mathbf{110}^\alpha\mathbf{1} : \ \alpha \ge 1\}$$
$$\cup \ \{(\mathbf{10})^{\ell_1}\mathbf{110}^\alpha(\mathbf{10})^{\ell_2}\mathbf{11} : \ \ell_1 + \ell_2 = 4, \ \ell_1, \ell_2 \ge 0, \ \alpha \ge 1\}.$$

  A direct computer search shows that none of the above forms satisfies $s(x_1^2) = 5$ (note that for sufficiently large α the sum of digits of the above forms stabilizes since the blocks in the square do not interfere anymore, so this is a finite verification.) Alternatively, we could also check the solutions of (16) via the algorithms in the Section 5.
  (b) If $s(x_1^2) = 4$. We have $s(x_0^2) = 3$ and $x_0 \in E_3$. Thus $s(x_1) \in \{7, 8, 9\}$. Algorithm `max-integer` shows that $E_{4,\lambda}$ is empty for $\lambda \in \{7, 8, 9\}$ which allows to conclude.

(4) $\underline{s(x_1 \cdot x_0) = 5.}$

- If $\boxed{k=9}$, then we have $s(x_1^2) = s(x_0^2) = 2$ and $x_1 = x_0 = 3$. Hence $s(x_1) + s(x_0) = 4 < 9$.
- If $\boxed{k=10}$, then we have $\max(s(x_1^2), s(x_0^2)) = 3$ and $\min(s(x_1^2), s(x_0^2)) = 2$ and $s(x_1) + s(x_0) \le 6 < 10$.
- Let $\boxed{k=11}$.
  (a) If $s(x_1^2) = 4$, then we have $s(x_0^2) = 2$ and $x_0 = 3$. Thus $s(x_1) = 9$ and we conclude since $E_{4,9}$ is empty.
  (b) If $s(x_1^2) = 3$, then we have $x_0, x_1 \in E_3$ and we have $s(x_1) + s(x_0) \le 8 < 11$.

(5) $\underline{s(x_1 \cdot x_0) = 6.}$

- If $\boxed{k=10}$, then we have $s(x_1^2) = s(x_0^2) = 2$ and $x_1 = x_0 = 3$. Then $s(x_1) + s(x_0) = 4 < 10$.
- If $\boxed{k=11}$, then we have $\max(s(x_1^2), s(x_0^2)) = 3$ and $\min(s(x_1^2), s(x_0^2)) = 2$ and again $s(x_1) + s(x_0) \leq 6 < 11$.

(6) $\underline{s(x_1 \cdot x_0) = 7.}$ Here, necessarily $\boxed{k=11}$, thus we have $s(x_1^2) = s(x_0^2) = 2$ and $x_1 = x_0 = 3$. Then $s(x_1) + s(x_0) = 4 < 11$.

The proof of Lemma 9 is therefore complete for the case $m = 1$.

4.2. **The case $m = 2$.** Here, we change our strategy and use the interference graph given in Figure 1. There are five independents summands $y_{2,2}, y_{2,1}, y_{2,0}, y_{1,0}, y_{0,0}$ each contributing to $(n)_2$ with at least one **1**-bit and only $y_{2,0}$ may interfere with $y_{1,1}$. None of these five summands can contribute with more than $(k-4)$ **1**-bits since $s(n^2) = k$.

We distinguish the following cases.

(1) $\underline{x_2 = 1.}$
- If $\boxed{k=9}$ then $s(y_{2,1}) = s(x_1)$ and x_1 has at most five **1**-bits. Hence $x_0 \neq 1$ and $s(y_{1,0}) \geq 2$. Therefore,
$$1 + s(x_1) + s(x_0) = 9 \geq 1 + s(x_1) + 1 + 2 + s(x_0^2).$$
Thus $s(x_0^2) \leq s(x_0) - 3$. This condition has no solution when $2 \leq s(x_0) \leq 7$, see [9].
- If $\boxed{k=10}$ then $s(y_{2,1}) = s(x_1)$ and x_1 has at most six **1**-bits. Hence $x_0 \neq 1$ and $s(y_{1,0}) \geq 2$. As before, we get $s(x_0^2) \leq s(x_0) - 3$. The only solution to this is $\mathsf{x}_0 = \mathbf{111011111}$ and $x_1 = 1$, see [9], Table 3. But now $y_{1,0} = x_0$ has $8 > 6$ **1**-bits, which is too many.
- If $\boxed{k=11}$ then $s(y_{2,1}) = s(x_1)$ and x_1 has at most seven **1**-bits, thus $x_0 \neq 1$ and $s(y_{1,0}) \geq 2$. Again, $2 \leq s(x_0^2) \leq s(x_0) - 3$ and the only solution for $s(x_0) \leq 8$ is $\mathsf{x}_0 = \mathbf{111011111}$. We get $\mathsf{x}_1 = \mathbf{10}^\alpha\mathbf{1}$ for some $\alpha \geq 0$. But this implies $s(y_{1,0}) \geq 7$ and this contribution is too large to $s(n^2) = 11$ to hold since $s(y_{0,0}) = 5$. In fact, $s(y_{1,0})$ is constant for $\alpha \geq 9$ so we can check the values for $0 \leq \alpha \leq 9$ directly. If there were any such solution for $s(x_0) = 9$, then $x_1 = 1$ and so $s(y_{1,0}) = 9 > 11 - 4$. Again, this contradicts with $s(n^2) = 11$.

(2) $\underline{x_0 = 1}$. This is symmetric to the first case.

(3) $\underline{x_2 \neq 1 \text{ and } x_0 \neq 1.}$ The four summands $y_{2,2}, y_{2,1}, y_{1,0}, y_{0,0}$ contain more than one **1**-bit. Note also that $x_1 \neq 1$. In fact, if $x_1 = 1$, then $s(n) = s(x_2)+1+s(x_0) \leq s(y_{2,1})+s(y_{2,0})+s(y_{1,0}) < s(n^2)$.
- If $\boxed{k=9}$, then all these summands contain two **1**-bits and therefore $x_2 = x_0 = 3$. The fact that $s(x_1) = 5$ and $s(x_2x_1) = 2$, together with Lemma 3, implies that $\mathsf{x}_1 = \mathbf{10101011}$. Then the summand $\mathsf{y}_{1,1} = \mathbf{111001000111001}$ contributes with more than one **1**-bit to $(n^2)_2$. Indeed if we shift $\mathsf{y}_{1,1} = \mathbf{111001000111001}$ against $\mathsf{y}_{2,0} = \mathbf{1001}$ and add the terms, the result always has more than one **1**-bit (this can be checked by a simple `for`-loop, see the discussion in Section 3).
- If $\boxed{k=10}$, then at least three of the summands contain exactly two **1**-bits. By symmetry, we may assume that $y_{2,2}$ and $y_{2,1}$ contain exactly two **1**-bits. This implies $x_2 = 3$ and $\mathsf{x}_1 = (\mathbf{10})^\alpha\mathbf{11}$ for some $\alpha \geq 0$, depending on x_0.
  (a) If $s(y_{0,0}) = 2$ then $x_0 = 3$ and $\mathsf{x}_1 = \mathbf{1010101011}$. Thus we have $\mathsf{y}_{1,1} = \mathbf{1110001111000111001}$ which may interfere with $\mathsf{y}_{2,0} = \mathbf{1001}$. This interference contributes with more than two **1**-bits.

(b) If $s(y_{0,0}) = 3$ then the possible choices of x_0 are **10111**, **111** and $\mathbf{10}^\beta\mathbf{1}$ for some $\beta \geq 1$, see Szalay's result (8). Therefore, $4 \leq s(x_1) \leq 6$ and we can easily check that in all cases $y_{1,0}$ has more than two **1**-bits.

- If $\boxed{k = 11}$, the four summands $y_{2,2}, y_{2,1}, y_{1,0}, y_{0,0}$ contain more than one **1**-bit, thus at least two of them contain exactly two **1**-bits. We now distinguish the cases regarding the number of summands with two **1**-bits.

  First, suppose that there are at least three terms among $y_{2,2}, y_{2,1}, y_{1,0}, y_{0,0}$ that contain exactly two **1**-bits. By symmetry we can assume that $y_{2,2}$ and $y_{2,1}$ contain exactly two **1**-bits. This implies $\mathsf{x}_2 = \mathbf{11}$ and $\mathsf{x}_1 = (\mathbf{10})^\alpha\mathbf{11}$ for some $\alpha \geq 0$.

  (a) If $s(y_{0,0}) = 2$ then $\mathsf{x}_0 = \mathbf{11}$, $\mathsf{x}_1 = (\mathbf{10})^5\mathbf{11}$ and

  $$\mathsf{y}_{1,1} = \mathbf{1110001110011100011 1001}$$

  which may interfere with $\mathsf{y}_{2,0} = \mathbf{1001}$. This interference has more than three **1**-bits.

  (b) If $s(y_{0,0}) = 3$ then the possible choices for x_0 are **10111**, **111**, and $\mathbf{10}^\beta\mathbf{1}$ for some $\beta \geq 1$. Therefore, $s(x_1) \geq 5$ and in all cases $y_{1,0}$ has more than two **1**-bits.

  (c) If $s(y_{0,0}) = 4$ then, since $s(x_0) \leq 7$, the possible choices for x_0 are **101111**, **1101111**, **1111**, and **1101** (see Table 1 and Table 3 in [9]). Therefore, $s(x_1) \geq 3$ and in all cases $y_{1,0}$ has more than two **1**-bits.

  Suppose now that there are exactly two among $y_{2,2}, y_{2,1}, y_{1,0}, y_{0,0}$ that contain two **1**-bits. It is then sufficient to discuss the following cases:

  (a) $y_{2,2}$ and $y_{1,0}$ (or analogously $y_{2,1}$ and $y_{0,0}$) contain exactly two **1**-bits. Then $\mathsf{x}_2 = \mathbf{11}$ and $s(y_{0,0}) = 3$ so that $x_0 \in E_3$. Hence, $5 \leq s(x_1) \leq 7$. We now consider $y_{2,1} = 3x_1$ and $s(3x_1) = 3$.
  By Lemma 5, the only solutions for x_1 are $\mathbf{1}(\mathbf{01})^{\ell_1}(\mathbf{10})^{\ell_2}\mathbf{11}$ for some $\ell_1, \ell_2 \geq 0$ such that $2 \leq \ell_1 + \ell_2 \leq 4$.
  Therefore, all the possible values for $\mathsf{y}_{2,0}$ and $\mathsf{y}_{1,1}$ are:

$\mathsf{y}_{2,0}$	$\mathsf{y}_{1,1}$
10101	**1110110010001**
1000101	**10000001011001**
$\mathbf{110}^{t-2}\mathbf{11}$, $t \geq 2$	**10110010111001**
	11100101110010001
	11101011001011001
	100000001010111001
	100000001010111001
	11101010110101010111001
	11101010110101010111001
	11101010110101010111001
	1000000010000000111001
	1000000010000000111001

  Therefore, in each of the above possibilities the terms $y_{1,1}$ and $y_{2,0}$ will contribute with more than one **1**-bit.

  (b) $y_{2,2}$ and $y_{0,0}$ contain exactly two **1**-bits. Then $\mathsf{x}_2 = \mathsf{x}_0 = \mathbf{11}$ so $s(x_1) = 7$. Thus the contribution of interference of $y_{2,0}$ with $y_{1,1}$ is only one **1**-bit if and only if $(x_1^2)_2$ is of the form $\mathbf{1}\cdots\mathbf{10111}$. This implies that $x_1^2 \equiv 7 \pmod 8$, which is not possible for any odd integer x_1.

(c) $y_{2,1}$ and $y_{1,0}$ contain exactly two **1**-bits. Then $y_{2,2}$ and $y_{0,0}$ contain exactly three **1**-bits thus $x_0, x_2 \in E_3$. The summand $y_{2,0}$ which might interfere with $y_{1,1}$ has to be one **1**-bit.
The following forms for $y_{2,0}$ contradict the fact that $y_{1,1} \equiv 1 \pmod 8$:

$\mathsf{y}_{2,0}$	(x_2, x_0)	Requested form for $\mathsf{y}_{1,1}$
110001	(7,7)	$(*)$**11**
1000010001	(23,23)	$(*)$**11**
10100001	(7,23)	$(*)$**11**
10001	$(7, 2^2+1)$	$(*)$**11**
1110011	$(23, 2^2+1)$	$(*)$**101**
-	$(2^{t_2}+1, 2^{t_0}+1), t_2, t_0 \geq 2$	$(*)$**11**

The remaining forms are more complicated because the same argument does not work.

$\mathsf{y}_{2,0}$	(x_2, x_0)	Requested form for $\mathsf{y}_{1,1}$
11001111	$(23, 2^3+1)$	$\mathbf{1}\cdots$**100110001**
110000111	$(23, 2^4+1)$	$\mathbf{1}\cdots$**1001111001**
$\mathbf{1110}^{t-3}\mathbf{111}$	$(7, 2^t+1), t \geq 3$	$\mathbf{1}\cdots\mathbf{10001}^{t-3}\mathbf{001}$
$\mathbf{101110}^{t-5}\mathbf{10111}$	$(23, 2^t+1), t \geq 5$	$\mathbf{1}\cdots\mathbf{1010001}^{t-5}\mathbf{01001}$

The first case implies that $x_1^2 = 2^8(2^\lambda - 1) + 49$ for some $\lambda \geq 1$, since $s(x_1) = 5$ gives that $x_1^2 = 49$ is not possible. However, λ is bounded since $s(x_1^2) \leq s(x_1)(s(x_1)+1)/2$, so $\lambda \leq 12$. Thus it is sufficient to check if the integers $2^8(2^\lambda - 1) + 49$, for $0 \leq \lambda \leq 12$ are perfect square, and it is not the case.
The second case is similar since we have $x_1^2 = 2^8(2^\lambda - 1) + 111$ for some $\lambda \geq 1$ and $s(x_1) = 4$. We conclude in the same way as before.
As for the third case, we have $x_1^2 = 2^{t+4}(2^\lambda - 1) + 2^3(2^{t-3} - 1) + 1$ for some $\lambda \geq 1$ and $s(x_1) = 6$. Again, t is bounded and we find the only solution $x_1 = 31$ for $t = 3$. However, $s(y_{2,1}) = s(7 \times 31) = 5$ contradicts our first hypothesis in this case. The last case works in a same manner and we are done.

Therefore, the proof of Lemma 9 is complete for $m = 2$.

4.3. **The case** $m = 3$**.** There are seven independent summands $y_{3,3}, \ldots, y_{3,0}, \ldots, y_{0,0}$ each contributing to $(n^2)_2$ with at least one **1**-bit, see Figure 2. The considerations for $k = 9, 10, 11$ are slightly different here.

If $\boxed{k = 9}$ then at least five of the seven independents summands have to contain only one **1**-bit.

(1) $\underline{x_3 = x_0 = 1.}$ We have $y_{3,2} = x_2$ and $y_{1,0} = x_1$. Therefore , $s(n^2) > s(y_{3,3}) + s(y_{3,2}) + s(y_{1,0}) + s(y_{0,0}) = s(x_3) + s(x_2) + s(x_1) + s(x_0) = 9$, a contradiction.

(2) $\underline{x_3 \neq 1.}$ The summands $y_{3,3}$ and $y_{3,2}$ contain two **1**-bits while all the other summands only contribute with one **1**-bit. Thus, $x_3 = 3$, $x_1 = x_0 = 1$, and $s(x_2) = 5$. Lemma 3 shows that the only solution to $s(y_{3,2}) = 2$ is $\mathsf{x}_2 =$ **10101011**. However, the summand $\mathsf{y}_{2,2} =$ **111001000111001** can only interfere with either $\mathsf{y}_{3,1} =$ **11** or $\mathsf{y}_{3,0} =$ **11** and in both cases its contribution to n^2 is larger than one **1**-bit.

(3) $\underline{x_0 \neq 1.}$ This is symmetric to the previous case.

If $\boxed{k = 10}$ then, among the seven independent summands $y_{3,3}, \ldots, y_{3,0}, \ldots, y_{0,0}$, at least four of them must contain exactly one **1**-bit and none of the summands can contribute with more than

three **1**-bits. In fact, if there were a summand with more than three **1**-bits then all other six summands have to contribute with a single **1**-bit which is impossible. We now distinguish several cases.

(1) $\underline{x_3 = x_0 = 1}$. We have $y_{3,2} = x_2$ and $y_{1,0} = x_1$, therefore, $s(n^2) > s(y_{3,3}) + s(y_{3,2}) + s(y_{1,0}) + \underline{s(y_{0,0}) = 10}$, a contradiction.

(2) $\underline{s(y_{3,3}) = 2}$. This implies $\mathsf{x}_3 = \mathbf{11}$: the summand $y_{3,2}$ contains at least two **1**-bits, thus, one of $\mathsf{y}_{1,0}$ and $\mathsf{y}_{0,0}$ is **1** which implies that $x_0 = 1$.

(a) $\underline{s(y_{3,2}) = 2 \text{ and } s(y_{1,0}) = 2}$. As $y_{1,0} = x_1$ and $s(x_1) = 2$, we obtain $\mathsf{x}_2 = (\mathbf{10})^3\mathbf{11}$ and $\mathsf{y}_{2,2} = \mathbf{111001000111001}$. This summand may interfere with one of

$$\mathsf{y}_{3,1} \in \{\mathbf{110}^\gamma\mathbf{11} : \gamma \geq 0\} \cup \{\mathbf{1001}\}$$

and $\mathsf{y}_{3,0} = \mathbf{11}$, in both cases its contribution will be more than one **1**-bit. Note that for $y_{3,1}$ we can use a `for`-loop over γ to conclude.

(b) $\underline{s(y_{3,2}) = 2 \text{ and } s(y_{1,0}) = 1}$. We have $x_1 = 1$, $\mathsf{x}_2 = (\mathbf{10})^4\mathbf{11}$ and

$$\mathsf{y}_{2,2} = \mathbf{1110001111000111001}.$$

This summand may interfere with one of $y_{3,1} = x_3 = 3$ and $y_{3,0} = x_3 = 3$, but its contribution will be more than two **1**-bits.

(c) $\underline{s(y_{3,2}) = 3}$. Here again $x_1 = 1$ and the summand $y_{2,1} = x_2$ can only interfere with $\mathsf{y}_{3,0} = \mathbf{11}$ and they must add up to a power of 2. According to Lemma 5, the system $s(3x_2) = 3$, $s(x_2) = 6$ implies that

$$\mathsf{x}_2 \in \{\mathbf{101010111}, \mathbf{101101011}, \mathbf{101011011}, \mathbf{110101011}\}.$$

In any case, the interference between $y_{2,1}$ and $y_{3,0} = 3$ would then again contribute with too many **1**-bits.

(3) $\underline{s(y_{3,3}) = 3}$. We have $x_3 \in E_3$ and $s(y_{3,2}) \geq 2$. Furthermore, we have $x_1 = x_0 = 1$ because $y_{1,0} = 1$. In any case $4 \leq s(x_2) \leq 6$ the summand $y_{2,0} = x_2$ has to interfere with $y_{1,1} = 1$ and add up to a power of 2. Thus, $\mathsf{x}_2 = \mathbf{1}^j$ for $4 \leq j \leq 6$ and $\mathsf{y}_{2,2} = \mathbf{1}^{j-1}\mathbf{0}^j\mathbf{1}$. This summand can only interfere with one of $y_{3,1} = x_3$ and $y_{3,0} = x_3$ and we see that the contribution of this summand is then again more than one **1**-bit in all cases.

(4) $\underline{s(y_{0,0}) \geq 2}$. These cases are symmetric to the previous two cases.

If $\boxed{k = 11}$ then at least three of the independents summands have to contain one **1**-bit only. We distinguish between the cases:

(1) $\underline{x_3 = x_0 = 1}$. We have $y_{3,2} = x_2$ and $y_{1,0} = x_1$, therefore, $s(n^2) > s(y_{3,3}) + s(y_{3,2}) + s(y_{1,0}) + \underline{s(y_{0,0}) = 11}$, again a contradiction.

(2) $\underline{s(y_{3,3}) = 4}$. Since $s(x_3) \leq 8$ the possible values for x_3 are **101111**, **1101111**, **1111** and **1101** (see [9]). Also, since in this case $s(y_{3,2}) \geq 2$, we have $x_1 = x_0 = 1$. This implies $3 \leq s(x_2) \leq 6$. The summand $y_{2,0} = x_2$ has to interfere with $y_{1,1} = 1$ and the contribution has exactly one **1**-bit. Hence, $\mathsf{x}_2 = \mathbf{1}^i$ for $3 \leq i \leq 6$ and $\mathsf{y}_{2,2} = \mathbf{1}^{i-1}\mathbf{0}^i\mathbf{1}$. This summand can only interfere with one of $y_{3,1} = x_3$ and $y_{3,0} = x_3$ and we can see that the contribution of this summand is more than one **1**-bit in each case.

(3) $\underline{s(y_{3,3}) = 3}$. Possible forms of x_3 are **10111**, **111** and $\mathbf{10}^i\mathbf{1}$ for some $i \geq 1$. Furthermore, $\underline{x_0 = 1 \text{ since}}$ one of $y_{0,0}$ and $y_{1,0}$ is 1.

(a) If $s(x_1) = 2$ then in any case $4 \leq s(x_2) \leq 6$. The summand $y_{2,0} = x_2$ has to interfere with $y_{1,1}$ and contributes with one **1**-bit. Since $\mathsf{x}_1 = \mathbf{10}^j\mathbf{1}$ for some $j \geq 0$, we have $\mathsf{y}_{1,1}$ equals to **1001** or $\mathbf{10}^{j-1}\mathbf{10}^{j+1}\mathbf{1}$ for some $j \geq 1$. A computation regarding the possible interference of the summands shows that we have one of the following cases:

$s(x_2)$	x_2
4	$\{\mathbf{10111}, \mathbf{100111}\}$
5	$\{\mathbf{110111}, \mathbf{1100111}, \mathbf{101111}\}$
6	$\{\mathbf{1110111}, \mathbf{11100111}, \mathbf{10101111}\}$

By computing every value of $y_{2,2}$ we finally have to check the possible interference of $y_{2,2}$ with $y_{3,1}$ or $y_{3,0} = x_3$. In each case the result has more than one **1**-bit and this leads to a contradiction.

(b) If $x_1 = 1$ then $5 \leq s(x_2) \leq 7$. First, suppose that the summand $y_{2,0} = x_2$ interferes with $y_{1,1} = 1$ and adds up to a power of 2. Then $\mathsf{x}_2 = \mathbf{1}^j$ for $5 \leq j \leq 7$ and $s(y_{3,2}) \geq 3$. The term $\mathsf{y}_{2,2} = \mathbf{1}^{j-1}\mathbf{0}^j\mathbf{1}$ can only interfere with one of $y_{3,1} = x_3$ and $y_{3,0} = x_3$ and the contribution is again more than one **1**-bit. If $y_{2,0} = x_2$ interferes with $y_{1,1} = 1$ and gives a contribution of the form $\mathbf{10}^j\mathbf{1}$, then x_2 is of the form $\mathbf{11110}^{j-4}\mathbf{1}$, $\mathbf{111110}^{j-5}\mathbf{1}$ or $\mathbf{1111110}^{j-6}\mathbf{1}$. We compute $y_{2,2}$ in all theses cases and conclude that this summand contributes with more than one **1**-bit when it interacts with x_3.

(4) $s(y_{3,3}) = 2$. We have $\mathsf{x}_3 = \mathbf{11}$.

First, we assume that $x_0 = 1$. We have the following cases:

$s(y_{3,2})$	$s(y_{1,0})$	x_2	$\mathsf{y}_{2,2}$
2	3	$(\mathbf{10})^3\mathbf{11}$	$\mathbf{111001000111001}$
2	2	$(\mathbf{10})^4\mathbf{11}$	$\mathbf{1110001111000111001}$
2	1	$(\mathbf{10})^5\mathbf{11}$	$\mathbf{11100011100111000111001}$
3	2	$(\mathbf{10})^4\mathbf{11}$	$\mathbf{1110001111000111001}$
3	1	$(\mathbf{10})^5\mathbf{11}$	$\mathbf{11100011100111000111001}$
4	1	$(\mathbf{10})^5\mathbf{11}$	$\mathbf{11100011100111000111001}$

In each case, $y_{2,2}$ may interfere with one of $y_{3,1} = x_3 \cdot x_1$ and $\mathsf{y}_{3,0} = \mathbf{11}$. In both cases; this interference is always more than one **1**-bit due to the submultiplicativity property of the sum of digits function. Indeed, this fact ensure that $s(x_3 \cdot x_1) \leq s(x_3) \cdot s(x_1) \leq 6$ and it is not sufficient to cancel all the inner **1**-bits in $y_{2,2}$.

Secondly, assume that $x_0 \neq 1$. Then, $s(y_{0,0}) = s(y_{1,0}) = s(y_{3,2}) = 2$ and so $\mathsf{x}_0 = \mathbf{11}$, $\mathsf{x}_1 = (\mathbf{10})^i\mathbf{11}$ and $\mathsf{x}_2 = (\mathbf{10})^j\mathbf{11}$ for some positive integers i and j such that $i + j = 3$. (The case $x_1 = 1$ and/or $x_2 = 1$ is easier and can be treated in a similar fashion.) By symmetry we may suppose $i < j$.

(a) If $j = 3$, then $\mathsf{x}_2 = \mathbf{10101011}$ and $\mathsf{x}_1 = \mathbf{11}$, and we have $\mathsf{y}_{2,2} = \mathbf{111001000111001}$. This summand may interfere with one of $\mathsf{y}_{3,1} = \mathbf{1001}$ and $\mathsf{y}_{3,0} = \mathbf{1001}$, but this contribution is always more than one **1**-bit.

(b) If $j = 2$, then $\mathsf{x}_2 = \mathbf{101011}$ and $\mathsf{x}_1 = \mathbf{1011}$. Then we have $\mathsf{y}_{2,2} = \mathbf{11100111001}$. This summand may interfere with one of $\mathsf{y}_{3,1} = \mathbf{100001}$ and $\mathsf{y}_{3,0} = \mathbf{1001}$, but this contribution is always more than one **1**-bit.

Therefore, the proof of Lemma 9 is complete for $m = 3$.

4.4. **The case** $m = 4$**.** There are nine independent summands $y_{4,4}, \ldots, y_{4,0}, \ldots, y_{0,0}$ each contributing to n^2 with at least one **1**-bit and at most $(k-8)$ **1**-bits, see Figure 3. While there will be more restrictions compared to the previous cases due to the fact that there are more independent terms, the downside is that we have now three levels of interaction in the interference graph.

Let $\boxed{k = 9}$. This is the easiest case since it implies that all of these summands contribute to n^2 with exactly one **1**-bit. We have $y_{4,4} = y_{4,3} = y_{1,0} = y_{0,0} = 1$ it implies that $x_4 = x_3 = x_1 = x_0 = 1$

and $s(x_2) = 5$. The summand $y_{4,2} = x_2$ can only interfere with $y_{3,3} = 1$ and the result of adding these two summands has to be a power of 2; this is only possible if $\mathsf{x}_2 = \mathbf{11111}$. Now, the summand $\mathsf{y}_{2,2} = \mathbf{111100001}$ can interfere with two of the summands $y_{3,1}, y_{4,1}, y_{3,0}$ and $y_{4,0}$. As each of these four summands is 1, this contribution to n^2 has more than one **1**-bit. This concludes this case.

For $\boxed{k = 10}$, among the nine independant summands, only one can contribute with two **1**-bits. We immediately obtain that $x_0 = x_4 = 1$ and one of x_1 and x_3 is 1.

(1) $\underline{x_3 \neq 1}$ (or symmetrically $x_1 \neq 1$). We see that $s(x_3) = s(y_{4,3}) = 2$ and $s(x_2) = 5$. The factor $y_{2,0} = x_2$, which can only interfere with $y_{1,1} = 1$, has to contribute with one **1**-bit only, therefore, $\mathsf{x}_2 = \mathbf{11111}$. Now, the summand $\mathsf{y}_{2,2} = \mathbf{1111000001}$ can interfere with two of the summands $y_{3,1} = x_3, y_{4,1} = 1, y_{3,0} = x_3$, and $y_{4,0} = 1$. This contribution to n^2 has more than one **1**-bit.

(2) $\underline{x_3 = x_1 = 1}$. We have $s(x_2) = 6$ and the factor $y_{2,0} = x_2$, which can only interfere with $y_{1,1} = 1$, has to contribute with one **1**-bit only. Indeed if it contributes with more than **1**-bit, by symmetry $y_{4,2}$ will interfere with $y_{3,3} = 1$ with more than one **1**-bit and we have a contradiction. In fact, if $\mathsf{x}_2 \neq \mathbf{111111}$, then the factor $y_{2,0} = x_2$ (resp. $y_{4,2} = x_2$), which can only interfere with $y_{1,1} = 1$ (resp. $y_{3,3} = 1$) contributes contribute with more than one **1**-bit. Therefore, $\mathsf{x}_2 = \mathbf{111111}$. The summand $\mathsf{y}_{2,2} = \mathbf{111110000001}$ can interfere with two of the summands $y_{3,1} = 1$, $y_{4,1} = 1$, $y_{3,0} = 1$, and $y_{4,0} = 1$ and its contribution to n^2 has to be at most two **1**-bits. There is a solution for the given conditions as the summands may be shifted against each other. However, as we will see next, the contradiction arises when we try to find a solution for $\hat{\ell_1}, \ldots, \hat{\ell_4}$, where

$$\hat{\ell_j} = \sum_{i=1}^{j} \ell_i + |\mathsf{x}_{i-1}|$$

(we refer to Section 3 for the notation). The following pairs of summands have to interfere with each other such that their contribution has one **1**-bit only:

$$(y_{2,0}, y_{1,1}),\ (y_{3,0}, y_{2,1}),\ (y_{4,1}, y_{3,2}),\ (y_{4,2}, y_{3,3}).$$

More precisely, their least significant bits have to align:

$$\begin{cases} \hat{\ell_2} + 1 = 2\hat{\ell_1} \implies \hat{\ell_2} = 2\hat{\ell_1} - 1. \\ \hat{\ell_3} + 1 = \hat{\ell_2} + \hat{\ell_1} + 1 \implies \hat{\ell_3} = 3\hat{\ell_1} - 1. \\ \hat{\ell_4} + \hat{\ell_1} + 1 = \hat{\ell_3} + \hat{\ell_2} + 1 \implies \hat{\ell_4} = 4\hat{\ell_1} - 2. \\ \hat{\ell_4} + \hat{\ell_2} + 1 = 2\hat{\ell_3}. \end{cases}$$

Recall that $\hat{\ell_0} = 0$. Thus, the summands $y_{2,2}, y_{3,1}$, and $y_{4,0}$ align as

$$\begin{array}{r} \mathbf{111110000001} \\ \mathbf{1} \\ \mathbf{1} \end{array}$$

and their contribution to n^2 is **111110000111** which is too much.

For $\boxed{k = 11}$ we distinguish the following cases:

(1) $\underline{x_4 \neq 1}$. The summands $y_{4,4}$ and $y_{4,3}$ have to contain two **1**-bits and all other summands only contribute with one **1**-bit. Thus $x_4 = 3$ and $x_1 = x_0 = 1$. As $s(y_{4,3}) = 2$, we have $\mathsf{x}_3 = (\mathbf{10})^i\mathbf{11}$ for some $0 \leq i \leq 4$. Furthermore $y_{2,0} = x_2$ can only interfere with $y_{1,1}$ and this summand has to contribute with only one **1**-bit so that $\mathsf{x}_2 = \mathbf{1}^k$ for some $1 \leq k \leq 5$. Yet

the summand $y_{4,2}$ can only interfere with $y_{3,3}$. The statement $\sum_{0\le i\le 4} s(x_i) = 11$ implies $i + k = 5$. Thus the possible values for the couple $(y_{4,2}, y_{3,3})$ are:

$\mathsf{y}_{4,2}$	$\mathsf{y}_{3,3}$
1011101	**1001**
101101	**1111001**
10101	**11100111001**
1001	**111001000111001**
11	**1110001111000111001**

This gives a contribution to n^2 that has more than one **1**-bit.

(2) $\underline{x_0 \neq 1}$. This is symmetric to the previous case.

(3) $\underline{x_4 = x_0 = 1}$. By symmetry we may assume that $s(x_3) \geq s(x_1)$. As each of the nine independent summands has to contribute to n^2 with at least one **1**-bit, by inspection of $y_{4,3} = x_3$ we have $s(x_3) \leq 3$. By an inspection of $y_{1,0} = x_1$, we have in the same way $s(x_1) \leq 3$. If $s(x_3) = 3$, then $s(x_1) = 1$. We finally have to discuss the following four cases:

$s(x_3)$	$s(x_1)$	$s(x_2)$
3	1	5
2	2	5
2	1	6
1	1	7

For the $\underline{\text{first line}}$ in this table, except for $y_{4,3} = x_3$, all contributions have to be exactly one **1**-bit. Since $y_{2,0} = x_2$ can only interfere with $y_{1,1} = 1$, we have $\mathsf{x}_2 = \mathbf{11111}$. Furthermore, $y_{4,2} = x_2$ can only interfere with $y_{3,3} = x_3^2$ and this contribution has to be a single **1**-bit. This happens if and only if $(x_3^2)_2 = \mathbf{1}\cdots\mathbf{100001}$, i.e. x_3^2 correspond to the integer $1 + 2^5(2^\lambda - 1)$ for some $\lambda \geq 1$. We can solve this such as in the case $m = 2, k = 11$ (c), or more directly with the following calculation: Write $x_3 = 1 + 2^a + 2^b$ for some $0 < a < b$. Then

$$\begin{aligned} x_3^2 &= 1 + 2^{a+1} + 2^{b+1} + 2^{2a} + 2^{a+b+1} + 2^{2b}, \\ &= 1 + 2^{a+1}(1 + 2^{b-a} + 2^{a-1} + 2^b + 2^{2b-a-1}). \end{aligned}$$

Thus we have $a = 4$ and all other power of 2 have to be consecutive. This implies in particular $b - a = 2$ and $b = 3$. This is not possible since $a < b$.

For the $\underline{\text{second line}}$, $\mathsf{x}_1 = \mathbf{10}^i\mathbf{1}$ for some $i \geq 0$. Thus $\mathsf{y}_{1,1} = \mathbf{1001}$ for $i = 0$ or $\mathbf{10}^{i-1}\mathbf{10}^{i+1}\mathbf{1}$ for $i \geq 1$. Again, the summand $y_{2,0} = x_2$ can only interfere with $y_{1,1}$ and this contribution has more than one **1**-bit except for the cases $(\mathsf{x}_1, \mathsf{x}_2) = (\mathbf{11}, \mathbf{110111})$ and $(\mathbf{101}, \mathbf{1100111})$. For $i \geq 2$, having a contribution of one **1**-bit implies that $s(x_2) \geq 6$, and the blocks of **0**-bits of $y_{1,1}$ are too large to be covered. Yet $y_{4,2} = x_2$ can only interfere with $y_{3,3}$ with only one **1**-bit and we have the same result as before. The value of x_2 sets the values of x_1 and x_3, and we have to study the two following cases:

(a) If $\mathsf{x}_2 = \mathbf{110111}$, then we have $\mathsf{x}_1 = \mathsf{x}_3 = \mathbf{11}$. This implies $\mathsf{y}_{3,2} = \mathbf{10100101}$ and $y_{3,2}$ can interfere with $y_{4,1} = x_1$ and $y_{4,0} = x_0$. In all cases the contribution is more than one **1**-bit.

(b) If $\mathsf{x}_2 = \mathbf{1100111}$, then we have $\mathsf{x}_1 = \mathsf{x}_3 = \mathbf{101}$. This implies $\mathsf{y}_{3,2} = \mathbf{1000000011}$ and $y_{3,2}$ can interfere with $y_{4,1} = x_1$ and $y_{4,0} = x_0$. In all cases the contribution is more than one **1**-bit.

For the third line, we have that $y_{2,0} = x_2$ can only interfere with $y_{1,1} = x_1^2 = 1$ with at most two **1**-bits. We write $\mathsf{x}_3 = \mathbf{10}^i\mathbf{1}$ for some $i \geq 0$. We distinguish two cases according to this contribution.

(a) If this contribution has only one **1**-bit then we have $\mathsf{x}_2 = \mathbf{111111}$. Yet $y_{4,2} = x_2$ can only interfere with $y_{3,3} = x_3^2$ and the contribution has at most two **1**-bits. This implies $\mathsf{x}_3 \in \{\mathbf{11}, \mathbf{1001}, \mathbf{100001}, \mathbf{10000001}\}$, i.e $i \in \{0, 2, 4, 6\}$. To see this, if $i \geq 8$, the **0**-blocks are too large in x_3^2 and we can check all other possible values of i directly. In all theses cases the contribution is exactly of two **1**-bits. The summand $y_{3,2} = x_3x_2$ can only interfere with $y_{4,1} = 1$ and $y_{4,0} = 1$. Since $\mathsf{x}_{3,2} \in \{\mathbf{10111101}, \mathbf{1000110111}, \mathbf{100000011111}, \mathbf{1111110111111}\}$, all of theses contributions are more than one **1**-bit.

(b) Suppose now that the contribution between $y_{2,0}$ and $y_{1,1}$ is exactly two **1**-bits. We then have $\mathsf{x}_2 = \mathbf{1011111}$ or $\mathsf{x}_2 = \mathbf{1111101}$. Moreover, $y_{4,2} = x_2$ can only interfere with $y_{3,3} = x_3^2$ and this contribution has to be exactly one **1**-bit. However, in both cases, this contribution exceeds one **1**-bit, by a similar argument as before for the different values of i.

For the fourth line, the summand $y_{2,0} = x_2$ can only interfere with $y_{1,1} = x_1$ with at most three **1**-bits. As in the precedent case, we have a contribution of one **1**-bit if and only if $\mathsf{x}_2 = \mathbf{1111111}$, resp., a contribution of two **1**-bits if and only if

$$\mathsf{x}_2 \in \{\mathbf{10}^a\mathbf{111111} : \ a \geq 1\} \ \cup \ \{\mathbf{1111110}^a\mathbf{1} : \ a \geq 1\} := B_2,$$

resp., a contribution of three **1**-bits if and only if

$$\mathsf{x}_2 \in \{\mathbf{10}^{a'}\mathbf{10}^a\mathbf{11111} : \ a, a' \geq 1\} \ \cup \ \{\mathbf{10}^{a'}\mathbf{111110}^a\mathbf{1} : \ a, a' \geq 1\}$$
$$\cup \ \{\mathbf{111110}^{a'}\mathbf{10}^a\mathbf{1} : \ a, a' \geq 1\}.$$

Denote this last union by B_3.

(a) If $\mathsf{x}_2 \in B_3$ then $y_{4,2}$ can only interfere with $y_{3,3} = 1$ resulting in one **1**-bit. This is not possible since $y_{4,2} = x_2$.

(b) If $\mathsf{x}_2 \in B_2$ then $y_{4,2} = x_2$ can only interfere with $y_{3,3} = 1$ with at most two **1**-bits. If this contribution does not exceed two bits, it has to be exactly two **1**-bits by the form of the binary expansion of x_2. This implies that the summand $y_{3,2} = x_2$ can interfere with $y_{4,1} = 1$ and $y_{4,0} = 1$ with at most one **1**-bit. This is not possible.

(c) If $\mathsf{x}_2 = \mathbf{11111111}$ then we have $\mathsf{y}_{2,2} = \mathbf{11111100000001}$ can interfere with $y_{4,1}$, $y_{4,0}$, $y_{3,1}$ and $y_{3,0}$ and all contributions should be equal to **1** in the end. All theses contributions have more than one **1**-bit since the **0**-block is too large.

Therefore, the proof of Lemma 9 is complete for $m = 4$.

4.5. **The case** $m = 5$**.** There are eleven independent summands $y_{5,5}, \ldots, y_{5,0}, \ldots, y_{0,0}$ each contributing to n^2 with exactly one **1**-bit. We have not drawn the interference graph since it gets too large and it is not needed to follow the argument. Recall that we here necessarily have $\boxed{k = 11}$. Since $y_{5,5} = y_{5,4} = y_{1,0} = y_{0,0} = 1$ then $x_5 = x_4 = x_1 = x_0 = 1$. The summand $y_{5,3} = x_3$ can only interfere with $y_{4,4} = 1$ and the result of these two summands has to be a power of two. The same remark is also true for $y_{2,0}$ which could only interfere with $y_{1,1} = 1$. We therefore have $\mathsf{x}_3 = \mathbf{1}^{n_3}$ and $\mathsf{x}_2 = \mathbf{1}^{n_2}$ for some $n_3 \geq 1$ and $n_2 \geq 1$ such that $n_3 + n_2 = 7$.

(1) If $\underline{n_3 = 6}$, i.e. $\mathsf{x}_3 = \mathbf{111111}$, then $\mathsf{y}_{3,3} = \mathbf{111110000001}$ can interfere with three of the summands $y_{5,2}, y_{5,1}, y_{5,0}, y_{4,2}, y_{4,1}$, and $y_{4,0}$. As each of these six summands is 1, this contribution to n^2 has more than **1**-bit since the **0**-block of $y_{3,3}$ containing 6 consecutive **0**-bits is too large.

(2) If $\underline{n_3 = 5}$, i.e. $\mathsf{x}_3 = \mathbf{11111}$ and $\mathsf{x}_2 = \mathbf{11}$. Then $\mathsf{y}_{3,3} = \mathbf{1111000001}$ can interfere with three of the summands $\mathsf{y}_{5,2} = \mathsf{y}_{4,2} = \mathbf{11}$, $y_{5,1} = y_{5,0} = y_{4,1} = y_{4,0} = 1$. We conclude as done previously.

(3) If $\underline{n_3 = 4}$, i.e $\mathsf{x}_3 = \mathbf{1111}$ and $\mathsf{x}_2 = \mathbf{111}$ then $\mathsf{y}_{3,3} = \mathbf{11100001}$ can interfere with three of the summands $\mathsf{y}_{5,2} = \mathsf{y}_{4,2} = \mathbf{111}$, $y_{5,1} = y_{5,0} = y_{4,1} = y_{4,0} = 1$. There exists a solution to this which is graphically presented as follows:

$$\mathbf{1111100001}$$
$$\mathbf{111}$$
$$\mathbf{1}$$
$$\mathbf{1}$$

Here we have to study other situations of interference in order to deduce a contradiction. The following pairs of summands have to interfere and to contribute with one **1**-bit: $(y_{2,0}, y_{1,1})$, $(y_{3,0}, y_{2,1})$, $(y_{5,2}, y_{4,3})$, $(y_{5,3}, y_{4,4})$. More precisely, their last significant bits have to align:

$$\begin{cases} \hat{\ell_2} + 1 = 2\hat{\ell_1}. \\ \hat{\ell_3} + 1 = \hat{\ell_2} + \hat{\ell_1} + 1. \\ \hat{\ell_5} + \hat{\ell_2} + 1 = \hat{\ell_4} + \hat{\ell_3} + 1. \\ \hat{\ell_5} + \hat{\ell_3} + 1 = 2\hat{\ell_4}. \end{cases}$$

This implies $\hat{\ell_2} = 2\hat{\ell_1} - 1$, $\hat{\ell_3} = 3\hat{\ell_1} - 1$, $\hat{\ell_4} = 4\hat{\ell_1}$ and $\hat{\ell_5} = 5\hat{\ell_1}$. Thus the summand $y_{3,3}$ does not align in a correct way with all the other summands. In fact, since we are looking for one **1**-bit of contribution, we need an alignment of the first bit. Here this is not the case since all the following values are different

$$\begin{cases} 2\hat{\ell_3} = 6\hat{\ell_1} - 2 \\ \hat{\ell_5} + \hat{\ell_2} + 1 = 7\hat{\ell_1}. \\ \hat{\ell_4} + \hat{\ell_2} + 1 = 6\hat{\ell_1}. \\ \hat{\ell_5} + \hat{\ell_1} + 1 = 6\hat{\ell_1} + 1. \\ \hat{\ell_5} + 1 = 5\hat{\ell_1} + 1. \\ \hat{\ell_4} + \hat{\ell_1} + 1 = 5\hat{\ell_1} + 1. \\ \hat{\ell_4} + 1 = 4\hat{\ell_1} + 1. \end{cases}$$

This provides us with the wanted contradiction.

(4) If $1 \leq n_3 \leq 3$ then it turns out that $\mathsf{x}_2 = \mathbf{1}^{n_2}$ for some $4 \leq n_2 \leq 6$ which is symmetric to one of the previous cases.

Therefore, the proof of Lemma 9 is complete for $m = 5$, and since all cases $1 \leq m \leq 5$ are treated, this finishes the proof of Lemma 9 and therefore of Theorem 1.

5. On the equation $s(n^2) \in \{4, 5\}$

The aim in this section is to study the equations $s(n^2) \in \{4, 5\}$ in odd integers. Compared to the previous sections, the point of view is different here as there is no precondition on the weight of n. Since there is neither an *a priori* bound on the length of n, a simple direct computation is not sufficient to determine finiteness of solutions. Our aim is to solve these equations for all n composed by as many **1**-bits as possible. The heuristic is that the larger the weight of n the more

unlikely such an n can be solution of $s(n^2) \leq 5$, since carry propagations have to cancel out more and more bits. In view of Conjecture 2, under this heuristic, this shows that it is more and more improbable to find new solutions other than those given by that conjecture.

5.1. **The case $s(n^2) = 4$.** Let n be an odd integer with $s(n) = k \geq 9$ such that $s(n^2) = 4$. To start with, we can suppose that $k \geq 9$ since all the other cases are done in [9], but our algorithms could also handle smaller k. Write $n = 1 + 2^\ell m$ with $\ell \geq 1$ and m an odd integer which satisfies $s(m) = k - 1$. Thus we have $n^2 = 1 + 2^{\ell+1}m + 2^{2\ell}m^2$ and it implies that $s(2^{\ell+1}m + 2^{2\ell}m^2) = 3$. Otherwise said, we have

$$s(m + 2^{\ell-1}m^2) = 3. \tag{17}$$

At this point there are two possible ways to attack the problem. The first one would be to use Lemma 8, more precisely, a specific case in its proof where the upper bound can be improved, see [10]. This allows to get

$$m(1 + 2^{\ell-1}m) \leq 2^{k(k-1)-13}, \tag{18}$$

and therefore we would have $m < 2^{k(k-1)-(\ell-1)/2-2}$ which in turn implies a bound on ℓ and what would remain is to use the algorithm `next` for each such ℓ to find the set of the solutions. However, this method is not sufficient for the case $s(n^2) = 5$ and this is the main reason that we have created the algorithm `max-integer` that we describe shortly in the sequel (we give a more detailed description in Section 6).

According to (17) we have to allocate three **1**-bits in the sum $S = m + 2^{\ell-1}m^2$. In order to do so, we use the basic fact that if two integers a, b satisfy $a \equiv b \pmod{2^\lambda}$ then $a^2 \equiv b^2 \pmod{2^{\lambda+1}}$ for all $\lambda \geq 2$, and if $a \equiv b \pmod 2$ then $a^2 \equiv b^2 \pmod 8$. In this way if we write the binary decomposition of m bit by bit from the least significant digits to the highest significant digits, then we can also deduce at the same time the binary decomposition of $2^{\ell-1}m^2$ bit by bit again from the least significant digits to the highest significant digits. The algorithm tests if the next bit in the binary decomposition of m could be a **1**-bit or a **0**-bit in order to satisfy (17). Since we have supposed a bound on the weight of m, the algorithm stops when the allowed amount of **1**-bits is reached.

We show an example where we suppose $\ell = 1$ and the $\ell(m)$ least significant digits of S to be **0**. The first (rightmost) bit we add in the binary structure of m is a **1**-bit in order to propagate the carry. We can deduce the second bit of m^2 but in this case it is already determined, it is a **0**-bit. Since we have supposed that there are no **1**-bits on the lower significant part of the sum S, the third bit of m is necessarily a **1**-bit, and $m \equiv 7 \pmod 8$. Thus $m^2 \equiv 1 \pmod{16}$, and the fourth bit of m^2 is a **0**-bit. By iterating this argument we see that we can only add **1**-bits in m and we obtain the following sum with a block of $(k-1)$ **1**-bits in m (as before, we write (*) for an arbitrary finite string of bits).

$$\begin{array}{rrccccccc}
 & & \mathbf{1} & \cdots & \mathbf{1} & \mathbf{1} & \mathbf{1} & = & m \\
+ & (*) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{1} & = & m^2 \\
\hline
 & (*) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & = & S.
\end{array}$$

The algorithm also considers the cases where the right part of S contains one and two **1**-bits, see Section 6, the case above describes the main idea of the algorithm.

For the search algorithm to work efficiently, we are interested in finding good bounds for ℓ. In fact, there is a much better bound for ℓ than the one given by (18):

Lemma 10. *Let n be an odd integer such that $s(n) = k \geq 4$, $s(n^2) = 4$ and $n = 1 + 2^\ell m$ with $\ell \geq 2$ and m an odd integer. Then we have $\ell \leq 2k$.*

Proof. Suppose that $\ell > 2k$ and set $S = m + 2^{\ell-1}m^2$. Then $s(S) = 3$ and S is an odd integer. We consider the following addition (ω, ω' are binary words, and $\varepsilon_i \in \{\mathbf{0}, \mathbf{1}\}$):

$$\begin{array}{llllllllll} & & & \omega & \varepsilon_{\ell-3} & \cdots & \varepsilon_0 & \mathbf{1} & = & m \\ + & & (*) & \omega' & \mathbf{1} & & & & = & 2^{\ell-1}m^2 \\ \hline & & (*) & (*) & (*) & \varepsilon_{\ell-3} & \cdots & \varepsilon_0 & \mathbf{1} & = S. \end{array}$$

The block $\varepsilon_{\ell-3} \cdots \varepsilon_0$ is composed of at most one **1**-bit since S contains three **1**-bits and carries propagate only to the higher significant digits. We distinguish two cases according to the $\ell(m)$ lowest significant bits of S (note that this part contains well the contribution of ω from the first summand m and its interference with the rightmost **1**-bit of $2^{\ell-1}m^2$). This part will be called the (binary) *right part of S*. It contains at least one **1**-bit (the parity bit), and at most two **1**-bits (which includes the parity bit). It cannot contain three **1**-bits since the second summand has a binary expansion strictly longer than the first summand.

We write w for the integer whose binary expansion corresponds to ω.

(1) The right part of S contains only the parity **1**-bit. This implies that $\varepsilon_i = \mathbf{0}$ for all $0 \leq i \leq \ell - 3$. This means that $m = 1 + 2^{\ell-1}w$ and $m^2 = 1 + 2^\ell w + 2^{2\ell-2}w^2$. Thus the ℓ lower significant bits of m^2 are all **0**-bits except the parity **1**-digit:

$$\begin{array}{llllllllllll} & & & & & & & \omega & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} & = 1 + 2^{\ell-1}w \\ + & & & \omega & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} & & & & & = 2^{\ell-1}(1 + 2^\ell w) \\ + & \omega^2 & \mathbf{0} & \cdots & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & & & & & = 2^{\ell-1} \times 2^{2\ell-2}w^2 \\ \hline & & (*) & (*) & & (*) & & & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{1} & = S. \end{array}$$

Now, consider the additions of ω and **1** in the middle part between the first and the second summand. Since $\ell > 2k > k$, the word ω in the second summand does not interfere with the ω of the first summand. This implies that ω is a single block of $(k-1)$ consecutive **1**-bits since otherwise the carry does propagate sufficiently far. This implies that

$$m = 1 + 2^{\ell-1}(2^{k-1} - 1).$$

Thus $m^2 = 1 + 2^\ell(2^{k-1} - 1) + 2^{2\ell-2}(2^{2k-2} - 2^k + 1)$, and we have

$$\begin{aligned} m + 2^{\ell-1}m^2 &= 1 + 2^{\ell-1}(2^{k-1} - 1) + 2^{\ell-1} + 2^{2\ell-1}(2^{k-1} - 1) + 2^{3\ell-3}(2^{2k-2} - 2^k + 1) \\ &= 1 + 2^{\ell-2+k} + 2^{2\ell-1}(2^{k-1} - 1) + 2^{3\ell-3}(2^{2k-2} - 2^k + 1). \end{aligned}$$

Since $\ell > 2k$, the terms in the above sum are non-interfering and therefore $m + 2^{\ell-1}m^2$ has too many **1**-bits.

(2) The right part of S contains two isolated **1**-bits.

There are two cases:

If this **1**-bit is located within the block $\varepsilon_{\ell-3} \cdots \varepsilon_0$ then there exists i_0 such that $\varepsilon_{i_0} = \mathbf{1}$ and $\varepsilon_i = \mathbf{0}$ for $i \neq i_0$. Thus $m = 1 + 2^{i_0} + 2^{\ell-1}w$, with $1 \leq i_0 < \ell - 1$, and

$$2^{\ell-1}m^2 = 2^{\ell-1} + 2^{i_0+\ell} + 2^{2i_0+\ell-1} + 2^{2\ell-1}w + 2^{i_0+2\ell-1}w + 2^{3\ell-3}w^2.$$

(a) If $2i_0 \geq \ell$ then we have $2^{\ell-1}m^2 = 2^{\ell-1} + 2^{i_0+\ell} + 2^{2\ell-1}w'$ for some integer w'. With a similar argument as in the former case, we get

$$m = 1 + 2^{i_0} + 2^{\ell-1}((2^{i_0} - 1) + 2^{i_0+1}(2^{k'} - 1)),$$

for some $k' \geq 0$ with $k' + i_0 = k - 2$. This leads to $i_0 \geq \ell/2 > k > k - 2$, which gives a contradiction.

(b) If $\ell/4 < i_0 < \ell/2$ then we have $2^{2i_0+\ell-1} < 2^{2\ell}$ and this implies that m has the form

$$m = 1 + 2^{i_0} + 2^{\ell-1}((2^{i_0} - 1) + 2^{i_0+1}(2^{i_0-1} - 1) + 2^{2i_0+1}(2^{k'} - 1)).$$

This leads to $s(m) \geq 2i_0 > \ell/2 > k$, which gives again a contradiction.

(c) If $i_0 \leq \ell/4$, then we have $2^{\ell-1}m^2 = 2^{\ell-1} + 2^{i_0+\ell} + 2^{2i_0+\ell-1} + 2^{2\ell-1}w'$ for some integer w' and we obtain

$$m = 1 + 2^{i_0} + 2^{\ell-1}((2^{i_0} - 1) + 2^{i_0+1}(2^{i_0-1} - 1) + 2^{2i_0+1}(2^{\ell-2i_0-1} - 1)).$$

This leads to $s(m) \geq \ell - 2i_0 \geq \ell/2 > k$, which gives a contradiction.

If $\varepsilon_i = \mathbf{0}$ for all i then we have two remaining cases. If w is even then we can use the same reasoning as before in the case (2) (a) since S has two isolated **1**-bits. If w is odd then we write $\omega = \omega'\mathbf{01}^\lambda$, with a possibly empty binary word ω' and $1 \leq \lambda \leq k-1$. We can suppose that ω' is not empty since $\omega = \mathbf{1}^{k-1}$ is already done in the case (1). Otherwise, S has two isolated **1**-bits and ω' is composed by a **0**-block of length at least ℓ. Therefore we can conclude with the same argument as before.

□

Our implementation of the algorithm `max-integer` shows that all odd solutions n such that $s(n) \leq 17$ are $(\ell, m) \in \{(3,2), (1,7), (1,23), (1,55)\}$ and this translates into

$$\bigcup_{\lambda \leq 17} E_{4,\lambda} = \{13,\ 15,\ 47,\ 111\},$$

which is (10) in Theorem 2.

5.2. **The case** $s(n^2) = 5$**.** The following result implies (12) in Theorem 2.

Lemma 11. *Let n be an odd integer such that $s(n^2) = 5$. Then:*

(1) *There is only a finite number of odd n such that $s(n) \geq 4$.*

(2) *If $s(n) = 3$, then n is of the form*

$$1 + 2^\ell + 2^{\ell+1}, \quad 1 + 2 + 2^\ell, \quad \text{or} \quad 1 + 2^\ell + 2^{2\ell-1},$$

for some $\ell \geq 3$.

Proof. We adapt Lemma 2 when the amount of **1**-bits in the square is fixed to be 5. The implied constant (i.e. the constant N_k appearing in its proof) will be different but we still we get that if there were an infinite number of solutions, then almost all (i.e. all but a finite number) of these solutions can be factorized this way. We again distinguish according to the number m of blocks in the factorization.

- $\underline{m = 1.}$ We have $(n)_2 = \mathsf{x}_1\mathbf{0}\cdots\mathbf{0}\mathsf{x}_0$, with a large inner block of **0**-bits. By symmetry we can suppose that $s(x_1) \geq s(x_0)$. Since we have the three independents contributions x_1^2, $x_1 \cdot x_0$ and x_0^2 for n^2, we see that exactly one of them has to contain one single **1**-bit. Thus $x_0 = 1$ andl $s(x_1^2) + s(x_1) = 4$, i.e $x_1 = 3$. Then n is on the form $1 + 2^\ell + 2^{\ell+1}$ for sufficiently large ℓ. We can easily check that this form is valid for all $\ell \geq 3$. By symmetry we have a second infinite family, namely $1 + 2 + 2^\ell$ for $\ell \geq 3$.
- $\underline{m = 2.}$ We use the interference graph given in Figure 1 to deduce that $x_2 = x_1 = x_0 = 1$ and the contribution of $x_1^2 + x_2 \cdot x_0$ is only of one **1**-bit. This implies that n is of the form $1 + 2^\ell + 2^{2\ell-1}$ for sufficiently large ℓ. As before, we can check that this form is valid for all $\ell \geq 3$.

□

We now show how to obtain (11) via the algorithm `max-integer`. The method used here is similar to the previous case. We suppose $k \geq 4$ to avoid the infinite families in Lemma 11. Let n be an odd integer such that $s(n) = k \geq 4$ and $s(n^2) = 5$. Let us write $n = 1 + 2^{\ell_1} + 2^{\ell_1+\ell_2} m$ with m an odd integer with $s(m) = k - 2$ and $\ell_1, \ell_2 \geq 1$. We have

$$n^2 = 1 + 2^{\ell_1+1} + 2^{2\ell_1} + 2^{\ell_1+\ell_2+1} m + 2^{2\ell_1+\ell_2+1} m + 2^{2\ell_1+2\ell_2} m^2. \tag{19}$$

We evaluate the number of isolated bits and deal with different cases according to the values of ℓ_1 and ℓ_2.

(1) $\underline{\ell_1 = 1.}$ Here (19) becomes

$$n^2 = 1 + 2^3 + 2^{\ell_2+2} m + 2^{\ell_2+3} m + 2^{2\ell_2+2} m^2.$$

Two subcases cases arise:

(a) $\underline{\ell_2 > 1.}$ We here have two isolated bits (associated with the powers 1 and 2^3) and this leads to

$$s(m \cdot (3 + 2^{\ell_2} m)) = 3.$$

By a small adaptation of the algorithm `max-integer`, we find that the only solutions are $\ell_2 = 2, m = 11$ and $\ell_2 = 3, m = 3$. Thus $n = 51$ and $n = 91$ satisfy $s(n^2) = 5$.

(b) $\underline{\ell_2 = 1.}$ We have

$$s(1 + 3m + 2m^2) = s((2m + 1) \cdot (m + 1)) = 4.$$

Again, we adapt the algorithm `max-integer` and the solutions for m is the set

$$\{7,\ 19,\ 23,\ 55,\ 69,\ 119,\ 181,\ 367\}.$$

The set of solutions for n is therefore

$$\{31,\ 79,\ 95,\ 223,\ 279,\ 479,\ 727,\ 1471\}.$$

We mention that it is this case that motivated us to create the algorithm `max-integer` since the results from [10] are not sufficient to conclude.

(2) $\underline{\ell_1 > 1.}$ This leads to two isolated bits (corresponding to the power 1 and 2^{ℓ_1+1}). Thus (19) becomes

$$s(2^{\ell_1} + 2^{\ell_2+1} m + 2^{\ell_1+\ell_2+1} m + 2^{\ell_1+2\ell_2} m^2) = 3.$$

A last adaptation of the algorithm gives $n \in \{29,\ 157,\ 5793\}$ as the solution set.

Thus we have

$$\bigcup_{4 \leq \lambda \leq 15} E_{5,\lambda} = \{29,\ 31,\ 51,\ 79,\ 91,\ 95,\ 157,\ 223,\ 279,\ 479,\ 727,\ 1471,\ 5793\},$$

which is (11). Note that the solution with the largest weight is 1471 with $s(1471) = 9$.

6. Description of the algorithms

6.1. **Algorithm `next`.** The aim is to generate efficiently all odd integers smaller than a fixed bound with a fixed weight, and most importantly, the sets

$$\Delta_{\ell_1,\ell_2,m} = \{n \in \mathbb{N} : \quad s(n) = \ell_1, \quad s(n^2) \leq \ell_2, \quad n < 2^m, \quad n \text{ odd}\},$$

that we needed for our applications. The following result gives, starting from a given integer, the smallest integer with same weight larger than the given integer .

Lemma 12. *Let $n \geq 1$ be an integer. Write $(n)_2 = \mathtt{x}\mathbf{01}^{b+1}\mathbf{0}^c$ for some $b, c \geq 0$ and $\mathtt{x}$ a possibly empty binary word. Then the next integer by increasing order, denoted by m, with $s(m) = s(n)$ is $(m)_2 = \mathtt{x}\mathbf{10}^{c+1}\mathbf{1}^b$.*

Proof. It is clear that $s(m) = s(n)$ and suppose there exist an integer p such that $s(p) = s(m)$ and $n < p \leq m$. Since $p > n$ and $s(p) = s(n)$, a bit of index $\geq c + b + 1$ of p is **1** because $\mathbf{1}^{b+1}\mathbf{0}^c$ is the expansion of the largest integer of length $c + b + 1$ with weight $b + 1$. Since $p \leq m$, this index is exactly $c + b + 1$. By $p \leq m$, the binary expansion of p begins with a **1**-block of length b. This implies $p = m$. □

The algorithm `next` is a translation of Lemma 12. Given an integer n, the algorithm constructs the next integer by increasing order with same weight.

Algorithm 1: `next`

```
1 Procedure next(n):
2 c = index of the least significant set bit n ;
3 n = n/2^c ;
4 n = n + 1 ;
5 b = index of the least significant set bit n -1 ;
6 n = 2^c · n ;
7 n = n|2^b ;                                  /* | is the OR operator */
8 n = n − 1 ;
```

Now, having constructed the set of integers with fixed weight, the second step is to determine the weight of their squares. The program uses the fact that for an integer n of the form $n = m + 2^L p$ and $m < 2^L$, we have $n^2 = m^2 + 2^{L+1}mp + 2^{2L}p^2$. Thus the $L + 1$ lowest significant digits of n^2 are determined by m, i.e. the lower part of n. In our study, we are interested in integers whose squares contain only a small number of **1**-bits. As a consequence, if the lower part of n^2 contains already too many **1**-bits, then we can already reject the integer as a solution, and it is not necessary to compute explicitly all the square n^2. This preliminary calculus reduces drastically the computation time. For efficiency and practical issues, we have implemented this algorithm with $L = 64$.

We have parallelized our program and distributed the calculation on multiple threads according to a suffix before making the `next` procedure. Indeed, for a fixed a we can consider integers of the form $n = 1 + 2^a + 2^{a+1}m$ for odd m and to find `next`(n) it is sufficient to execute `next`(m). This is equivalent to fix the place of the second **1**-bit in n. The cutting is therefore done via

$$\Delta_{\ell_1,\ell_2,a,m} = \{n \in \mathbb{N} :\ s(n) = \ell_1,\ s(n^2) \leq \ell_2,\ \text{ the second bit of } n \text{ is } a\ ,\ n < 2^m,\ n \text{ odd}\}$$

and

$$\Delta_{\ell_1,\ell_2,m} = \bigcup_{a=1}^{m-\ell_1+1} \Delta_{\ell_1,\ell_2,a,m}.$$

This cutting was necessary to conclude for the case $\boxed{k = 11}$ of (1), when $m = 1$.

Another issue arises with this parallelization. The number of integers in $\Delta_{\ell_1,\ell_2,a,m}$ is not equivalent. The smaller the value if a, the larger is the cardinality of $\Delta_{\ell_1,\ell_2,a,m}$. We have supposed that the time of computation for each integer is similar (this is heuristically supported by the use of the same binomial coefficients). With a preliminary calculation, we designed specific implementations for each thread. By doing so, we could again reduce the global computation time.

6.2. **Algorithm `max-integer`.** We describe the algorithm for the equation $s(n^2) = 4$, which can be written as $s(y + 2^{\ell-1}y^2) = 3$ for a fixed weight $s(y) = k$ (see Section 5.1). The other cases are similar and only need some minor changes in the implementation.

Denote by λ the unique integer such that $2^{\lambda-1} \leq y < 2^{\lambda}$ and consider the following scheme for $y + 2^{\ell-1}y^2$:

$$\begin{array}{ccc|cccccc|l} & & & \mathbf{1} & \cdots & \varepsilon_\ell & \cdots & \varepsilon_1 & \mathbf{1} & = y \\ \mathbf{1} & \cdots & \eta_{\lambda-1} & \cdots & \cdots & \mathbf{1} & & & & = 2^{\ell-1}y^2 \\ \hline & y_1 & & & & y_2 & & & & \end{array}$$

We cut the sum into two binary blocks, y_1 and y_2. We have to allocate in total three **1**-bits for y_2 and y_1. We know that $s(y_1) \geq 1$ since the most significant digit of $y + 2^{\ell-1}y^2$ lies in the y_1-part. Let us focus on the case where $\ell = 1$, the other cases are similar.

As explained before, we tackle this problem step by step by adding bits in the binary decomposition of y. In this algorithm, we consider the binary blocks such as **011** and **11** to be different since we have more knowledge for the first block. In fact, in this context, it is more useful to see them as words rather than integers.

We say that a binary word ω is a *candidate* if the right part of the sum of $\omega+\omega^2$ (by a slight abuse of the notation) has at most two **1**-bits for a certain length of the block. If ω is a candidate then we can extend ω to $\mathbf{0}\omega$ and $\mathbf{1}\omega$ to the left and check if these two new words are again candidates. If a word ω is not a candidate, then it is not possible to extend it to a candidate word since the lower bits contains already too many **1**-bits and these bits are not influenced by adding new bits to ω since carry propagation is directed towards the higher significant digits. The algorithm starts with the word $\omega = \mathbf{1}$, constructs candidates, translates them into integers and checks whether they satisfy $s(n^2) = 4$. The algorithm stops when candidates cannot be extended.

For the algorithm to stop, we have two conditions. The first condition is at the core of the algorithm: a word that is already of weight k cannot be extended anymore with additional **1**-bits, so candidates have $\leq k$ **1**-bits. The second condition is on the length of the possible leading block of **0**-bits of a candidate of the form $\mathbf{0}\cdots\mathbf{0}\omega$. We have the following result.

Lemma 13. *Let ω be a candidate of length λ. Then the word $\mathbf{10}^{\lambda}\omega$ is not a candidate.*

Proof. In this case we have the following sum

$$\begin{array}{cccccccc|l} & & & & & \mathbf{1} & \mathbf{0}\cdots\mathbf{0}\ \omega & & = m \\ + & \mathbf{1} & \mathbf{0}\cdots\mathbf{0} & \omega & \mathbf{0}\cdots\mathbf{0} & \mathbf{0} & \omega^2 & & = m^2 \\ \hline & \mathbf{1} & & \omega & & \mathbf{1} & \omega^2+\omega & & \end{array}$$

The sum contains always more than three **1**-bits. □

Thus the algorithm is the following.

Algorithm 2: `max-integer`

```
1  Procedure max-integer(k):
2  S=[1] ;                                   /* Stack of all candidates */
3  while S is not empty do
4  |  ω=S.top() ;                           /* Top element of the stack */
5  |  n = 1 + 2ω ;
6  |  if s(n^2) = 4 then
7  |  |  print(n)
8  |  S.pop() ;                             /* remove ω from the stack */
9  |  if 1ω is a candidate and s(ω) < k then
10 |  |  S.push(1ω) ;                       /* add 1ω to the top of the stack */
11 |  if 0ω is a candidate and 2*(length of leading zeros of 0ω) < length of ω then
12 |  |  S.push(0ω) ;                       /* add 0ω to the top of the stack */
```

With respect to Theorem 2, (10), our implementation of the algorithm takes 102 sec to end for $s(n) = 16$ and 2h 50min for $s(n) = 17$ with a desk machine Intel(R) Core(TM) i9-9980HK CPU @ 2.40GHz. The code program is available here:
`https://gitlab.inria.fr/jamet/on-the-binary-digits-of-n-and-n2`.

7. The remaining cases $k = 14, 15$

We here consider the problem of determining the solutions of

$$s(n) = s(n^2) \in \{14, 15\},$$

which are the last two remaining cases in the original problem. These cases are much more difficult than the previous ones since we cannot rely on the former cases to resolve the problem. As already mentioned in Section 1, (2)–(4), there are infinitely many solutions for $s(n) = s(n^2) \in \{12, 13\}$.

To tackle these remaining cases, we improved our programs that determine the sets $\Delta_{\ell_1,\ell_2,m}$. For the case $k = 14$, there are many more subcases than before, and the investigation gets extremely cumbersome. We still can rely on Lemma 2 which gives decompositions into m blocks with $1 \leq m \leq 6$ for sufficiently large solutions.

For the case $m = 1$, we give in the sequel the constants implied by Lemma 7 and Lemma 8. This demonstrates the difficulty of the computation. Such as before, write $(n)_2 = \mathsf{x}_1\mathbf{0}\ldots\mathbf{0}\mathsf{x}_0$. By symmetry we can suppose that $s(x_1) \geq s(x_0)$. In this case we have $s(x_1)^2, s(x_0^2) \leq 10$ and the following table (see also Section 4.1):

$s(x_1)$	$s(x_0)$	Sets Δ for $s(x_1x_0) = 2$	Sets Δ for $s(x_1x_0) = 3$
7	7	$\Delta_{7,10,93} \times \Delta_{7,10,93}$	$\Delta_{7,9,182} \times \Delta_{7,9,182}$
8	6	$\Delta_{8,10,91} \times \Delta_{6,10,91}$	$\Delta_{8,9,178} \times \Delta_{6,9,178}$
9	5	$\Delta_{9,10,85} \times \Delta_{5,10,85}$	$\Delta_{9,9,166} \times \Delta_{5,9,166}$
10	4	$\Delta_{10,10,75} \times \Delta_{4,10,75}$	$\Delta_{10,9,146} \times \Delta_{4,9,146}$
11	3	$\Delta_{11,10,61} \times \Delta_{3,10,61}$	$\Delta_{11,9,118} \times \Delta_{3,9,118}$
12	2	$\Delta_{12,10,43} \times \Delta_{2,10,43}$	$\Delta_{12,9,82} \times \Delta_{2,9,82}$

The algorithm `next` gives us no solution for $s(x_1 \cdot x_0) = 2$ and only the couple $(3695, 143)$ for $s(x_1 \cdot x_0) = 3$. But we have $s(3695^2) + s(143^2) = 17 > 11$ and then this couple is not a solution. Therefore there is no infinite family such that $m = 1$ such as in (2)–(4).

For $k = 15$, we have the following sets to determine:

$s(x_1)$	$s(x_0)$	Sets Δ for $s(x_1x_0) = 2$	Sets Δ for $s(x_1x_0) = 3$
8	7	$\Delta_{8,11,107} \times \Delta_{7,11,107}$	$\Delta_{8,10,210} \times \Delta_{7,10,210}$
9	6	$\Delta_{9,11,103} \times \Delta_{6,11,103}$	$\Delta_{9,10,202} \times \Delta_{6,10,202}$
10	5	$\Delta_{10,11,95} \times \Delta_{5,11,95}$	$\Delta_{10,10,186} \times \Delta_{5,10,186}$
11	4	$\Delta_{11,11,83} \times \Delta_{4,11,83}$	$\Delta_{11,10,163} \times \Delta_{4,10,163}$
12	3	$\Delta_{12,11,67} \times \Delta_{3,11,67}$	$\Delta_{12,10,130} \times \Delta_{3,10,130}$
13	2	$\Delta_{13,11,47} \times \Delta_{2,11,47}$	$\Delta_{13,10,90} \times \Delta_{2,10,90}$

Finally, we have made a global search for $s(n^2) = s(n) = 11, 12, 13, 14, 15$ for $n < 2^{80}$ with a triple cutting. We found the following proportions:

k	Proportion of odd integers such that $s(n) = s(n^2) = k$ for $n < 2^{80}$	Running time	Number of cores used
11	$4 \cdot 10^{-10}$	2min 19sec	659
12	$1.5 \cdot 10^{-10}$	4min 58sec	480
13	$7.2 \cdot 10^{-11}$	32min 25sec	360
14	$2.6 \cdot 10^{-11}$	3h 34min 43sec	277
15	$1.2 \cdot 10^{-11}$	23h 24min 47sec	218

These five proportions are similar but there is a clear difference between the cases $k = 11$ and $k = 12, 13$. For $k = 11$ our algorithm finds that the largest solution is $n = 35463511416833$ of binary length 46. The structure in the solutions for $k = 12$ and $k = 13$ is clearly different since we can see a threshold between sporadic solutions and infinite families that are composed by small blocks. These infinite families already appear before this threshold. For example, for $k = 12$, we have that any solution of binary length larger than 55 is of the form $111 \cdot 2^t + 111$, but this form is already valid and appears for $t \geq 15$.

For $k = 14$, we still find solutions of binary length 80, such as $n = 605643510452789079965697$ for example. Nevertheless, no infinite family occurs clearly. For $k = 15$, the situation is similar: we have a solution of length 80, for example, $n = 605642350760526229274625$, again there is no obvious infinite family and the **1**-bits in the solutions do not follow an apparent rule. We believe that if an infinite family exists for $k = 14$ or $k = 15$, it should appear clearly for $n < 2^{80}$ already, as it is the case for $k = 12, 13$ and 16. It is therefore likely that there is only a finite number of solution. We formulated this in Conjecture 1.

This dichotomy between finite and infinite number of solutions for the problem (1) is rather surprising but seems at the same time to occur frequently in this context, for example, such as between E_3 (see (8)) and the conjectured set E_4 (see Conjecture 9). Interestingly enough, there exist again infinite (independent) families for the twisted system

$$s(n) = 14, \quad s(n^2) = 15,$$

namely

$$n = 23 \cdot 2^t + 2943, \text{ with } t \geq 13,$$

and

$$n = 727 \cdot 2^t + 727 \text{ with } t \geq 21.$$

To perform all calculations in our article, we used the cluster gros that consists of 123 nodes, Intel Xeon Gold 5220 and 18 cores / CPU, with 96 GiB of memory, see
`https://www.grid5000.fr/w/Nancy:Hardware#gros` and the code program is available
`https://gitlab.inria.fr/jamet/on-the-binary-digits-of-n-and-n2`.

Acknowledgements

The authors would like to thank Lukas Spiegelhofer for discussions and a very useful `C`-program. This work was supported partly by the French PIA project "Lorraine Université d'Excellence", reference ANR-15-IDEX-04-LUE, and by the projects ANR-18-CE40-0018 (EST) and ANR-20-CE91-0006 (ArithRand). The third author was supported by JSPS KAKENHI Grant Number 19K03439.

References

1. N. L. Bassily and I. Kátai, *Distribution of the values of q-additive functions on polynomial sequences*, Acta Math. Hung. **68** (1995), no. 4, 353–361 (English).
2. M. A. Bennett, *The polynomial-exponential equation* $1 + 2^a + 6^b = y^q$, Period. Math. Hungar. **75** (2017), no. 2, 387–397.
3. M. A. Bennett and Y. Bugeaud, *Perfect powers with three digits*, Mathematika **60** (2014), no. 1, 66–84.
4. M. A. Bennett, Y. Bugeaud, and M. Mignotte, *Perfect powers with few binary digits and related diophantine problems, ii*, Mathematical Proceedings of the Cambridge Philosophical Society **153** (2012), no. 3, 525–540.
5. A. Bérczes, L. Hajdu, T. Miyazaki, and I. Pink, *On the Diophantine equation* $1+x^a+z^b = y^n$, J. Comb. Number Theory **8** (2016), no. 2, 145–154.
6. P. Corvaja and U. Zannier, *Finiteness of odd perfect powers with four nonzero binary digits*, Ann. Inst. Fourier **63** (2013), no. 2, 715–731 (English).
7. L. Hajdu and I. Pink, *On the Diophantine equation* $1 + 2^a + x^b = y^n$, J. Number Theory **143** (2014), 1–13.
8. K. G. Hare, S. Laishram, and T. Stoll, *Stolarsky's conjecture and the sum of digits of polynomial values*, Proc. Am. Math. Soc. **139** (2011), no. 1, 39–49 (English).
9. K. G. Hare, S. Laishram, and T. Stoll, *The sum of digits of n and* n^2, Int. J. Number Theory **7** (2011), no. 7, 1737–1752.
10. H. Kaneko and T. Stoll, *Products of integers with few binary digits*, Uniform Distribution Theory (2022), to appear.
11. B. Lindström, *On the binary digits of a power*, J. Number Theory **65** (1997), no. 2, 321–324.
12. F. Luca, *The Diophantine equation* $x^2 = p^a \pm p^b + 1$, Acta Arith. **112** (2004), no. 1, 87–101.
13. M. Madritsch and T. Stoll, *On simultaneous digital expansions of polynomial values*, Acta Math. Hung. **143** (2014), no. 1, 192–200 (English).
14. S.-Y. Mei, *The sum of digits of polynomial values*, Integers **15** (2015), Paper No. A32, 12.
15. G. Melfi, *On simultaneous binary expansions of n and* n^2, J. Number Theory **111** (2005), no. 2, 248–256.
16. M. Peter, *The summatory function of the sum-of-digits function on polynomial sequences*, Acta Arith. **104** (2002), no. 1, 85–96 (English).
17. J. C. Saunders, *Sums of digits in q-ary expansions*, Int. J. Number Theory **11** (2015), no. 2, 593–611.
18. K. B. Stolarsky, *The binary digits of a power*, Proc. Am. Math. Soc. **71** (1978), 1–5 (English).
19. L. Szalay, *The equations* $2^n \pm 2^m \pm 2^l = z^2$, Indag. Math. (N.S.) **13** (2002), no. 1, 131–142.
20. ———, *Computational algorithm for solving the diophantine equations* $2^n \pm \alpha \cdot 2^m + \alpha^2 = x^2$, Houston J. Math. **46** (2020), no. 2, 295–306.

1. Université de Tunis El Manar, Institut Supérieur des Technologies Médicales de Tunis, 9 Rue Zouhair Essafi, 1006, Tunis, Tunisia; 2. Université de Sfax, Laboratoire d'Algèbre, Géométrie et Théorie Spectrale, Route de la Soukra, km 3.5, 3000, Sfax, Tunisia
Email address: alouikaram@yahoo.fr

LORIA, Campus Scientifique BP 239, F-54506 Vandœuvre-lès-Nancy, France;
Email address: damien.jamet@loria.fr

Institute of Mathematics, University of Tsukuba, 1-1-1, Tennodai, Tsukuba, Ibaraki, 305-8571, JAPAN; Research Core for Mathematical Sciences, University of Tsukuba, 1-1-1, Tennodai, Tsukuba, Ibaraki, 305-8571, JAPAN
Email address: kanekoha@math.tsukuba.ac.jp

Email address: steffen.kopecki@gmail.com

1. Université de Lorraine, Institut Elie Cartan de Lorraine, UMR 7502, Vandoeuvre-lès-Nancy, F-54506, France; 2. CNRS, Institut Elie Cartan de Lorraine, UMR 7502, Vandoeuvre-lès-Nancy, F-54506, France
Email address: pierre.popoli@univ-lorraine.fr

1. Université de Lorraine, Institut Elie Cartan de Lorraine, UMR 7502, Vandoeuvre-lès-Nancy, F-54506, France; 2. CNRS, Institut Elie Cartan de Lorraine, UMR 7502, Vandoeuvre-lès-Nancy, F-54506, France
Email address: thomas.stoll@univ-lorraine.fr