DIVISION SUDOKUS: INVARIANTS, ENUMERATION, AND MULTIPLE PARTITIONS

ALEŠ DRÁPAL

Department of Mathematics, Charles University, Sokolovská 83, 186 75, Praha 8, Czech Republic
e-mail: drapal@karlin.mff.cuni.cz

PETR VOJTĚCHOVSKÝ

Department of Mathematics, University of Denver, 2390 S. York St, Denver, CO 80208, USA
e-mail: petr@math.du.edu

Abstract. A division sudoku is a latin square whose all six conjugates are sudoku squares. We enumerate division sudokus up to a suitable equivalence, introduce powerful invariants of division sudokus, and also study latin squares that are division sudokus with respect to multiple partitions at the same time. We use nearfields and affine geometry to construct division sudokus of prime power rank that are rich in sudoku partitions.

1991 *Mathematics Subject Classification*. Primary: 05B15; Secondary: 20N05

1. Introduction. Sudoku is the most popular puzzle of our time. Many nontrivial results about sudoku squares and sudoku puzzles have been obtained by mathematicians and sudoku enthusiasts [2, 5, 11, 14, 16, 18, 22].

In this paper, we study a special type of sudokus that we propose to call *division sudokus*. Division sudokus can be thought of as sudokus for which the roles of rows, columns, and symbols are fully interchangeable. We have chosen the name division sudoku because a latin square is a division sudoku if and only if, when viewed as a quasigroup, the operation tables for multiplication, left division, and right division are all sudokus. (See Definition 2.1 for a more precise statement and Proposition 2.2 for a list of equivalent conditions.)

It appears that division sudokus were for the first time considered by Günter Stertenbrink in 2005 [22, p. 27] under the name 3dokus, which has since come to mean something else, namely a generalization of sudokus to three dimensions [1, 13]. Stertenbrink correctly determined the number of standard division sudokus and asked several natural questions concerning enumeration and completability of division sudokus. Bailey, Cameron, and Connelly [2] looked at a small class of sudokus (so called symmetric sudoku puzzles) from a design-theoretic point of view, and discussed an example [2, Figure 5] that happens to be a division sudoku.

We arrived at the notion of division sudokus independently by studying properties of the following, very interesting example of a sudoku square:

$$L_0$$

1	4	7	3	8	5	2	9	6
8	2	5	6	1	9	4	3	7
6	9	3	7	4	2	8	5	1
5	3	9	4	7	1	6	2	8
7	6	1	2	5	8	9	4	3
2	8	4	9	3	6	1	7	5
9	5	2	8	6	3	7	1	4
3	7	6	1	9	4	5	8	2
4	1	8	5	2	7	3	6	9

Not only is L_0 a division sudoku, but it happens to be the "least associative" latin square of order 9. In more detail, an *associative triple* in a quasigroup $(X, \cdot)$ is an ordered triple (x, y, z) such that $x \cdot (y \cdot z) = (x \cdot y) \cdot z$. It is easy to show that in a quasigroup of order n the number of associative triples is at least n. Until recently, it was an open problem to find a quasigroup of order $n > 1$ containing precisely n associative triples [12]. No such quasigroup exists for $n \leq 8$ [7]. With rows and columns labeled by $1, \ldots, 9$ in this order, the square L_0 is a multiplication table of a unique quasigroup of order 9 up to isomorphism with precisely nine associative triples [6, 8].

A *sudoku of rank m* is a latin square of order m^2 consisting of m^2 subsquares, each of size $m \times m$, such that each subsquare contains all m^2 symbols. The usual sudoku is therefore a sudoku of rank 3. Not much is known about sudokus of rank $m > 3$. In contrast, division sudokus are sufficiently restrictive so that many of their aspects can be investigated for larger ranks. Most of this paper is devoted to division sudokus of rank 3 but it may well turn out that questions about division sudokus of larger rank will lead to more interesting mathematics. We start this line of investigation in Section 6.

The content of this paper is as follows. In Section 2, we characterize division sudokus in various ways. In Section 3, we introduce two powerful invariants of division sudokus of rank 3, namely the intercalate structure invariant and the minisquare structure invariant. We use these invariants and several computational tools in Section 4 to enumerate division sudokus of rank 3 up to suitable equivalences. There are 104, 015, 259, 648 standard division sudokus of rank 3 (confirming Stertenbrink's result) forming 186 ds-isotopism classes (see Definition 3.1), 45 main ds-classes, and belonging to 183 isotopism classes of quasigroups.

In Section 5, we for the first time consider separate sudoku partitions for rows, columns, and symbols, so-called sudoku tri-partitions. (This paper could have started by defining division sudokus via tri-partitions. We opted for quasigroups since they provide a more formal setting for concepts like isotopisms.) We find a division sudoku of rank 3 with 24 sudoku tri-partitions. We then show how to synchronize as many sudoku tri-partitions as possible by group actions to obtain division sudokus with the largest number of sudoku partitions. There are only three essentially different division sudokus of rank 3 with more than one sudoku partition, namely with 3, 4, and 4 sudoku partitions, respectively.

Building upon ideas of Stein [21], in Section 6 we show how to construct division sudokus of prime power rank m that have at least $m + 1$ sudoku partitions. The construction is based on nearfields, affine planes, and subspaces. The underlying set is the finite field of order m^2 or, when m is odd, the quadratic nearfield of order m^2. If the rank m is a prime power of the form p^{2s}, we can improve the lower bound on the number of sudoku partitions to $m^2 + m$. For instance, there exists a division sudoku of rank 4 with 20 sudoku partitions.

Precisely one ds-isotopism class of division sudokus of rank 3 consists of sudokus isotopic to a group (namely to $C_3 \times C_3$). This class contains the example in [2, Figure 5].

1.1. Notation and terminology.   We start by recalling the standard terminology for latin squares and quasigroups. We refer the reader to [**4**, **9**] for more details.

Let X be a finite set. A *latin square* on X is an array $L = (L(x, y))_{x,y}$ with entries $L(x, y) \in X$ such that every row of L contains all symbols of X and every column of L contains all symbols of X. The algebraic counterpart of a latin square is a quasigroup. A *quasigroup* on X is a pair $(X, \cdot)$, where $\cdot$ is a binary operation on X such that for every a, $b \in X$ there are unique $x, y \in X$ satisfying $a \cdot x = b$ and $y \cdot a = b$. Once a labeling of rows and columns is fixed, there is a one-to-one correspondence between latin squares on X and (multiplication tables of) quasigroups on X. Throughout the paper, when L is a latin square on $X = \{1, 2, \ldots, n\}$, the associated quasigroup is obtained from L by labeling the rows and columns by $1, \ldots, n$, in this order.

A quasigroup can be defined equationally as a set X with three binary operations $\cdot, /, \backslash$ such that the identities $(x \cdot y)/y = x$, $(x/y) \cdot y = x$, $x \cdot (x \backslash y) = y$ and $x \backslash (x \cdot y) = y$ are satisfied. We refer to $/$ and $\backslash$ as *right division* and *left division*, respectively. If $(Q, \cdot)$ is a quasigroup, so are $(Q, /)$ and $(Q, \backslash)$.

The *orthogonal array* of a latin square L on X is the set

$$O(L) = \{(x, y, L(x, y)) : x, \ y \in X\} \subseteq X \times X \times X.$$

The latin square L can be reconstructed from $O(L)$ by setting $L(x, y)$ to be the unique z such that $(x, y, z) \in O(L)$.

The symmetric group on a set X will be denoted by $\mathrm{Sym}(X)$ and we also write $\mathrm{Sym}(\{1, \ldots, n\}) = \mathrm{Sym}(n)$. Given $\theta \in \mathrm{Sym}(3)$ and a latin square L, we denote by L^{θ} the latin square whose orthogonal array is equal to

$$\{(x_{\theta(1)}, x_{\theta(2)}, x_{\theta(3)}) : (x_1, x_2, x_3) \in O(L)\}.$$

For instance, $L^{(1,2)}$ is the matrix transpose of L. The six not necessarily distinct latin squares L^{θ} for $\theta \in \mathrm{Sym}(3)$ are called the *conjugates* of L. Algebraically speaking, the conjugates of $L = (X, \cdot)$ are precisely the quasigroups

$$L = (X, \cdot), \quad L^{(1,3)} = (X, /), \quad L^{(2,3)} = (X, \backslash)$$

and their opposites. Indeed, we have $x_1/x_2 = x_3$ iff $x_3 x_2 = x_1$ iff $(x_3, x_2, x_1) \in O(L)$, so $L^{(1,3)} = (X, /)$, and similarly for the left division.

An *isotopism* from a latin square L_1 to a latin square L_2 on X is an ordered triple (α, β, γ) of permutations of X such that

$$L_2(\alpha(x), \beta(y)) = \gamma(L_1(x, y))$$

holds for every $x, y \in X$, in which case the latin squares L_1, L_2 are said to be *isotopic*. Hence two latin squares are isotopic if and only if one is obtained from the other by a permutation of rows, columns, and symbols. Since isotopisms preserve combinatorial properties of latin squares, the classification of latin squares is often pursued up to isotopism. From the point of view of orthogonal arrays, it is also natural to permute the roles of rows, columns, and symbols. Two latin squares are said to be *paratopic* if one is isotopic to a conjugate of the other. The equivalence classes of latin squares induced by paratopy are called *main classes*.

Let us now introduce notation and terminology for sudokus. We will rely in part on the wonderfully descriptive concepts used by sudoku enthusiasts.

Let X be a set of size m^2. A partition $\{X_1, \ldots, X_m\}$ of X is a *sudoku partition* if $|X_i| = m$ for every $1 \le i \le m$. The subsets X_i are called *blocks*. A sudoku partition is *standard* if

$$X_i = \{(i-1)m + 1, (i-1)m + 2, \ldots, im\}$$

for every $1 \le i \le m$. A permutation $\theta \in \mathrm{Sym}(X)$ is said to *preserve the partition* $\{X_1, \ldots, X_m\}$ if the image $\theta(X_i)$ of every block X_i is a block.

When $m = 3$, we take advantage of the fact that $m^2 < 10$ and use compact notation for sudoku partitions. For instance, $\{123\ 456\ 789\}$ will denote the sudoku partition $\{\{1, 2, 3\}, \{4, 5, 6\}, \{7, 8, 9\}\}$.

Let L be a latin square on X and let $O(L)$ be its orthogonal array. The *band* of L corresponding to the block X_i is the set

$$B_i = \{(x, y, z) \in O(L) : x \in X_i\},$$

the *stack* of L corresponding to X_i is the set

$$S_i = \{(x, y, z) \in O(L) : y \in X_i\},$$

and the *pile* of L corresponding to X_i is the set

$$P_i = \{(x, y, z) \in O(L) : z \in X_i\}.$$

An intersection of a band and a stack is a *minisquare*. The rows and columns in a minisquare are *minirows* and *minicolumns*, respectively. In the following figure, we have highlighted the band B_2, stack S_2, and pile P_2 in the standard division sudoku L_0.
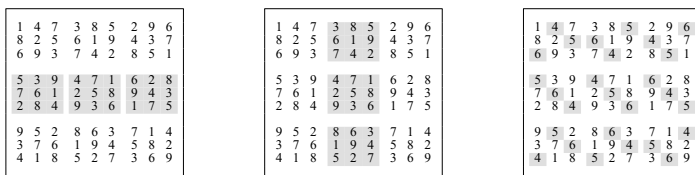
We will use informal terminology for subsets of orthogonal arrays. For instance, we say that $S \subseteq O(L)$ contains all symbols of X if for every $z \in X$, there exist $x, y \in X$ such that $(x, y, z) \in S$.

We use the icon ⌨ to indicate results obtained by computer computation. The calculations were carried out with Mace4 [15] and GAP [10], some using the package LOOPS [17].

1.2. Sudokus.

DEFINITION 1.1. Let $\{X_1, \ldots, X_m\}$ be a sudoku partition of X. A *sudoku (square) of rank m* is a latin square L on X such that every minisquare of L contains all symbols of X. A sudoku is *standard* if the partition is standard.

The following result goes back to 2003 and it was confirmed independently several times.

THEOREM 1.2 (Stertenbrink ⌨). *There are* 6, 670, 903, 752, 021, 072, 936, 960 *standard sudokus of rank* 3.

Just as the notions of isotopy and paratopy are the correct concepts for classifying latin squares and orthogonal arrays while preserving their combinatorial properties, so are the following two notions the correct concepts for classifying sudokus:

DEFINITION 1.3. Let $\{X_1, \ldots, X_m\}$ be a sudoku partition of X. Two sudokus L_1, L_2 defined on X are *s-isotopic* (short for *sudoku-isotopic*) if there exists an isotopism (α, β, γ) from L_1 to L_2 such that α, β are partition preserving. Two sudokus L_1, L_2 are *s-paratopic* (short for *sudoku-paratopic*) if L_1 is s-isotopic to L_2 or to $L_2^{(1,2)}$.

It is natural in the context of sudokus to demand that the permutations α, β be partition preserving, since the definition of sudoku depends on a fixed partition of rows and columns. The symbols of a sudoku are not partitioned in any way, however, hence the permutation γ is not restricted. Similarly, it is natural to consider only the conjugates L and $L^{(1,2)}$ of a sudoku L, not all six conjugates as in the case of latin square paratopisms, because the roles of rows and columns are symmetric in the definition of sudoku while the role of symbols is different.

THEOREM 1.4 (Russell and Jarvis [19] ▦). *There are* $10, 945, 437, 157$ *standard sudokus of rank* 3 *up to s-isotopism and* $5, 472, 730, 538$ *standard sudokus of rank* 3 *up to s-paratopism.*

2. Division sudokus and their characterization.

We are interested in those sudokus for which rows, columns, and symbols play symmetric roles.

DEFINITION 2.1. Let $\{X_1, \ldots, X_m\}$ be a sudoku partition of X. A *division sudoku (square) of rank* m is a latin square L on X such that the intersection of every band with every stack contains all symbols of X, the intersection of every band with every pile contains all columns of X, and the intersection of every stack with every pile contains all rows of X.

PROPOSITION 2.2. *Let* $\{X_1, \ldots, X_m\}$ *be a sudoku partition of* X. *Then the following conditions are equivalent for a latin square* L *on* X:

(i) *L is a division sudoku;*
(ii) *the conjugates L, $L^{(1,3)}$ and $L^{(2,3)}$ are sudokus;*
(iii) *all conjugates of L are sudokus;*
(iv) *L is a sudoku in which every minirow and every minicolumn contains precisely one entry from every pile.*

Proof. The definition of division sudoku says that L, $L^{(1,3)}$, and $L^{(2,3)}$ are sudokus. A latin square L is a sudoku if and only if the conjugate $L^{(1,2)}$ is a sudoku. Hence (i), (ii), and (iii) are equivalent.

Let now L be a sudoku of rank m and fix a band B_i and a pile P_j. Every minisquare contained in B_i contains all symbols, hence precisely m symbols from the pile P_j. Altogether, $|B_i \cap P_j| = m^2$. If L is a division sudoku then $B_i \cap P_j$ intersects all m^2 columns of X and it follows that every minicolumn of B_i contains precisely one element of P_j. Conversely, if every minicolumn of B_i contains precisely one element of P_j then $B_i \cap P_j$ intersects all columns of X. The argument for minirows is analogous by considering a stack S_i and a pile P_j. □

Condition (iv) of Proposition 2.2 is particularly suitable from a visual perspective since it does not require us to calculate and inspect conjugates.

We could develop a fully symmetric terminology in which minisquares, minirows, and minicolumns would make sense not only for the intersections of bands and stacks but also for the intersections of bands and piles, and stacks and piles. Then condition (iv) could be equivalently restated with respect to the new kind of miniobjects.

The next proposition offers a quasi-equational characterization of division sudokus in the variety of quasigroups.

PROPOSITION 2.3. *Let $\{X_1, \ldots, X_m\}$ be a sudoku partition of X. Write $x \sim y$ if x and y belong to the same block of X. Let $Q = (X, \cdot, /, \backslash)$ be a quasigroup. Consider the following conditions:*

$$\text{for all } x, y, u, v \in X \text{ if } x \sim u, y \sim v \text{ and } x \cdot y = u \cdot v \text{ then } (x, y) = (u, v), \quad (2.1)$$

$$\text{for all } x, y, u, v \in X \text{ if } x \sim u, y \sim v \text{ and } x/y = u/v \text{ then } (x, y) = (u, v), \quad (2.2)$$

$$\text{for all } x, y, u, v \in X \text{ if } x \sim u, y \sim v \text{ and } x\backslash y = u\backslash v \text{ then } (x, y) = (u, v). \quad (2.3)$$

Then Q is a sudoku if and only if (2.1) holds. Furthermore, Q is a division sudoku if and only if all (2.1), (2.2), and (2.3) hold.

Proof. We have $x \sim u$ and $y \sim v$ if and only if $(x, y, x \cdot y)$, $(u, v, u \cdot v)$ belong to the same minisquare, say M. The implication

$$x \cdot y = u \cdot v \implies (x, y) = (u, v)$$

then says that no symbol appears twice in M. Equivalently, every symbol appears in M, which is the sudoku condition by definition. The rest follows from Proposition 2.2 and the fact that $Q^{(1,3)} = (Q, /)$, $Q^{(2,3)} = (Q, \backslash)$. $\qquad\qquad \square$

For a quasigroup $(X, \cdot, /, \backslash)$ and $x \in X$ consider the three bijections $L_x, R_x, D_x \in \text{Sym}(X)$ defined by

$$L_x(y) = x \cdot y, \quad R_x(y) = y \cdot x, \quad D_x(y) = x/y.$$

The translations L_x, R_x are well known and the bijections D_x have been studied in [3, 20], for instance.

DEFINITION 2.4. *Given an equivalence relation $\sim$ on X and $\theta \in \text{Sym}(X)$, we say that θ shreds $\sim$ if whenever $u \sim v$ and $u \neq v$ then $\theta(u) \not\sim \theta(v)$. Equivalently, θ shreds $\sim$ if whenever $u \sim v$ and $\theta(u) \sim \theta(v)$ then $u = v$.*

Note that if $\{X_1, \ldots, X_m\}$ is a sudoku partition of X and $\sim$ is the corresponding equivalence relation, then $\theta \in \text{Sym}(X)$ shreds $\sim$ if and only if $|\theta(X_i) \cap X_j| = 1$ for every $1 \leq i, j \leq m$.

PROPOSITION 2.5. *Let $\{X_1, \ldots, X_m\}$ be a sudoku partition of X and let $\sim$ be the corresponding equivalence relation. Let $Q = (X, \cdot, /, \backslash)$ be a quasigroup. Consider the following conditions:*

$$\text{for all } x, u, v \in X \text{ if } u \sim v \text{ and } x/u \sim x/v \text{ then } u = v, \quad (2.4)$$

$$\text{for all } x, u, v \in X \text{ if } u \sim v \text{ and } x \cdot u \sim x \cdot v \text{ then } u = v, \quad (2.5)$$

$$\text{for all } x, u, v \in X \text{ if } u \sim v \text{ and } u \cdot x \sim v \cdot x \text{ then } u = v. \quad (2.6)$$

Then Q is a sudoku if and only if (2.4) holds, which happens if and only if for every $x \in X$ the bijection D_x shreds $\sim$. Furthermore, Q is a division sudoku if and only if all (2.4), (2.5), and (2.6) hold, which happens if and only if for every $x \in X$ the bijections D_x, L_x, and R_x shred $\sim$.

Proof. Upon substituting x/y for x and u/v for u into (2.1), we obtain

for all x, y, u, $v \in X$ if $x/y \sim u/v$, $y \sim v$ and $x = u$ then $(x/y, y) = (u/v, v)$.

Using $x = u$ and eliminating the variable u, we get

for all x, y, $v \in X$ if $x/y \sim x/v$ and $y \sim v$ then $y = v$.

This is (2.4) up to renaming of variables, and it says that every D_x shreds $\sim$.

Similarly, upon substituting $x \cdot y$ for x and $u \cdot v$ for u into (2.2), we get

for all x, y, u, $v \in X$ if $x \cdot y \sim u \cdot v$, $y \sim v$ and $x = u$ then $(x \cdot y, y) = (u \cdot v, v)$,

which can be rewritten as

for all x, y, $v \in X$ if $x \cdot y \sim x \cdot v$ and $y \sim v$ then $y = v$.

This is (2.5), which says that every L_x shreds $\sim$.

Finally, upon substituting $x \cdot y$ for y and $u \cdot v$ for v into (2.3), we get

for all x, y, u, $v \in X$ if $x \sim u$, $x \cdot y \sim u \cdot v$ and $y = v$ then $(x, x \cdot y) = (u, u \cdot v)$,

which can be rewritten as

for all x, y, $u \in X$ if $x \sim u$ and $x \cdot y \sim u \cdot y$ then $x = u$.

This is (2.6), which says that every R_x shreds $\sim$.   □

3. Invariants of division sudokus.   In this section, we introduce powerful invariants of division sudokus and enumerate division sudokus of rank 3. Our intent is to use only invariants that can be calculated by hand with a reasonable effort. We again start with two notions for classifying division sudokus while preserving their combinatorial properties.

DEFINITION 3.1. Let $\{X_1, \ldots, X_m\}$ be a sudoku partition of X. Two division sudokus L_1, L_2 on X are *ds-isotopic* (short for *division sudoku-isotopic*) if there exists an isotopism (α, β, γ) from L_1 to L_2 such that α, β, γ are partition preserving. Two division sudokus L_1, L_2 are *ds-paratopic* (short for *division sudoku-paratopic*) if L_1 is ds-isotopic to a conjugate of L_2. The equivalence classes of division sudokus induced by ds-paratopy are called *main ds-classes*.

An *intercalate* in a latin square is a 2×2 latin subsquare. Obviously, a given intercalate is contained in one or two bands, and similarly for stacks and piles.

LEMMA 3.2. *Let I be an intercalate of a division sudoku. Then at most one of the following situations can occur:*

  (i) *I is contained in a band;*
  (ii) *I is contained in a stack;*
  (iii) *I is contained in a pile.*

Proof. Suppose that both (i) and (ii) hold. Let I be contained in a band B_i and stack S_j. Let a be one of the two symbols contained in I. Then a occurs at least twice in the minisquare $B_i \cap S_j$, a contradiction.

We proceed similarly if any two conditions from among (i), (ii), and (iii) hold, using the fact that the concept of division sudokus is invariant under conjugacy.   □

When L is a division sudoku of rank 3 and I is an intercalate of L, we define $\text{band}_L(I)$ (resp. $\text{stack}_L(I)$, $\text{pile}_L(I)$) to be the unique band (resp. stack, pile) of L that either contains I or is disjoint from I. Note that rank 3 is necessary for this definition to make sense.

DEFINITION 3.3. Let L be a standard division sudoku of rank 3. The *intercalate structure invariant* ι_L of L is a $\{0, 1\}$-array indexed by bands, stacks, and piles, where $\iota_L(B_i, S_j, P_k) = 1$ if and only if there exists an intercalate I of L such that $(B_i, S_j, P_k) = (\text{band}_L(I), \text{stack}_L(I), \text{pile}_L(I))$.

Two intercalate structure invariants are said to be *equivalent* if one is obtained from the other by a permutation of band coordinates (rows), stack coordinates (columns), and pile coordinates (symbols).

The array ι_L can be visualized as a hypergraph with vertices consisting of bands, stacks, and piles, and hyperedges formed by those triangles (B_i, S_j, P_k) for which $\iota_L(B_i, S_j, P_k) = 1$. These graphs become unwieldy when L is rich in intercalates.

EXAMPLE 3.4. In the following is a standard division sudoku and its intercalate structure invariant.
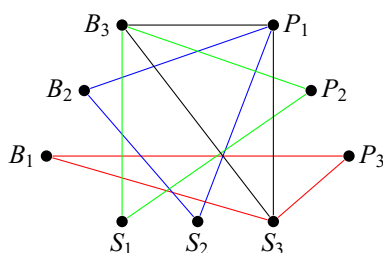
We have colored the hyperedges to improve legibility. For instance, the black hyperedge (B_3, S_3, P_1) is there on account of the highlighted intercalate that avoids B_3, avoids S_3, and lies within P_1.

PROPOSITION 3.5. *If L_1, L_2 are ds-isotopic standard division sudokus of rank* 3 *then ι_{L_1}, ι_{L_2} are equivalent intercalate structure invariants.*

Proof. Let (α, β, γ) be a ds-isotopism from L_1 to L_2. Let I_1 be an intercalate of L_1 and let I_2 be the image of I_1 under (α, β, γ). If I_1 belongs to a band of L_1 then I_2 belongs to a band of L_2 since α is partition preserving. We can argue similarly for stacks and piles. □

REMARK 3.6. It is possible to strengthen the intercalate structure invariant in various ways. For instance, the entry $\iota_L(B_i, S_j, P_k)$ could *count* the number of intercalates I of L such that $(B_i, S_j, P_k) = (\text{band}_L(I), \text{stack}_L(I), \text{pile}_L(I))$, rather than just indicate whether such an intercalate exists. Alternatively, taking Lemma 3.2 into account, we could define $\text{type}_L(I)$ by

$$\text{type}_L(I) = \begin{cases} 1, & \text{if } I \text{ is contained in a band of } L, \\ 2, & \text{if } I \text{ is contained in a stack of } L, \\ 3, & \text{if } I \text{ is contained in a pile of } L, \\ 4, & \text{otherwise}, \end{cases}$$

and consider a $3 \times 3 \times 3 \times 4$ array, where a given intercalate I contributes 1 to the cell indexed by $(\text{band}_L(I), \text{stack}_L(I), \text{pile}_L(I), \text{type}_L(I))$. It turns out that we do not need these refined invariants for our purposes.

DEFINITION 3.7. Let L be a standard division sudoku of rank 3. The *minisquare structure invariant* μ_L of L is a tripartite digraph obtained as follows. The nine vertices of μ_L are the bands, stacks, and piles of L. Let B be a band and S a stack of L, and let S_i, S_j be the two stacks different from S. For $1 \leq k \leq 3$ let $C_{i,k}$ (resp. $C_{j,k}$) be the set of symbols in the kth minicolumn of $B \cap S_i$ (resp. $B \cap S_j$). Then $B \to S$ is a directed edge if and only if $\{C_{i,1}, C_{i,2}, C_{i,3}\} = \{C_{j,1}, C_{j,2}, C_{j,3}\}$. Directed edges between stacks and bands, bands and piles, piles and bands, stacks and piles, and piles and stacks are determined analogously using conjugacy.

Two minisquare structure invariants are said to be *equivalent* if one is obtained from the other by a permutation of band vertices, stack vertices, and pile vertices.

EXAMPLE 3.8. In the following is a standard division sudoku and its minisquare structure invariant.
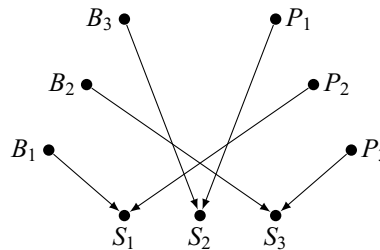
For instance, there is a directed edge $B_1 \to S_1$ because in the two highlighted minisquares $B_1 \cap S_2$, $B_1 \cap S_3$, the minicolumns contain the same sets of symbols, namely $\{1, 5, 9\}$, $\{2, 6, 7\}$, and $\{3, 4, 8\}$.

We immediately see the following.

PROPOSITION 3.9. *If L_1, L_2 are ds-isotopic standard division sudokus of rank 3 then μ_{L_1}, μ_{L_2} are equivalent minisquare structure invariants*

4. Enumeration. We enumerate division sudokus of rank 3.

4.1. Partial isotopisms, s-isotopisms, and ds-isotopisms.

LEMMA 4.1. *Let $Q_1 = (X, \cdot, /, \backslash)$, $Q_2 = (X, \circ, /\!\!/, \backslash\!\backslash)$ be quasigroups and let (α, β, γ) be an isotopism from Q_1 onto Q_2. If A, $B \subseteq X$ are such that $A \cdot B = \{a \cdot b : a \in A, b \in B\} = X$ then (α, β, γ) is determined by the values $(\alpha(a) : a \in A)$ and $(\beta(b) : b \in B)$.*

Proof. Suppose that α is known on A and β is known on B. Then $\gamma(a \cdot b) = \alpha(a) \circ \beta(b)$ shows that γ is known on $A \cdot B = X$. For any $b \in B$ and $x \in X$, we have $\alpha(x) = \gamma(x \cdot b)/\!\!/\beta(b)$ and thus α is known on X. Similarly, for any $a \in A$ and $y \in X$, we have $\beta(y) = \alpha(a)\backslash\!\backslash \gamma(a \cdot y)$, so β is also known on X. $\square$

COROLLARY 4.2. *Let $\{X_1, \ldots, X_m\}$ be a sudoku partition of X. Let Q_1 be a sudoku on X and Q_2 a quasigroup. Then for any $1 \leq i, j \leq m$, an isotopism (α, β, γ) from Q_1 to Q_2 is determined by the values $(\alpha(x) : x \in X_i)$ and $(\beta(x) : x \in X_j)$.*

```
1 4 7    2 . .    3 . .
8 2 5    . 3 .    . 1 .
6 9 3    . . 1    . . 2

2 . .    . . .    . . .
. 3 .    . . .    . . .
. . 1    . . .    . . .

3 . .    . . .    . . .
. 1 .    . . .    . . .
. . 2    . . .    . . .
```

Figure 1. Partial table of every standard division sudoku of rank 3 up to ds-isotopism.

LEMMA 4.3. *Let $\{X_1, \ldots, X_m\}$ be a sudoku partition of X and let $Q_1 = (X, \cdot)$, $Q_2 = (X, \circ)$ be sudokus. Then for any $1 \leq i, j \leq m$ and any $(m-1)$-element subsets $X_i' \subseteq X_i$ and $X_j' \subseteq X_j$, an s-isotopism (α, β, γ) from Q_1 to Q_2 is determined by the values $(\alpha(x) : x \in X_i')$ and $(\beta(x) : x \in X_j')$.*

Proof. Since α maps blocks to blocks, it maps X_i to some block X_k. We have $X_i = X_i' \cup \{x_i\}$ and $X_k = \alpha(X_i') \cup \{x_k\}$ for some $x_i \in X_i$, $x_k \in X_k$, and then $\alpha(x_i) = x_k$ is forced. Similarly for β and we are done by Corollary 4.2. □

Note that in the above three results we have assumed that the quasigroup operations for Q_1, Q_2 are fully known. However, the results can be sharpened by specifying exactly which parts of the multiplication and division tables of Q_1 and Q_2 need to be known. Lemma 4.4 is an example of such a result. It is based on the partial division sudoku in Figure 1. We shall see later that every standard division sudoku of rank 3 is ds-isotopic to a standard division sudoku that extends the partial table in Figure 1.

LEMMA 4.4. *Let $\{X_1, X_2, X_3\}$ be a sudoku partition of X and let $Q_1 = (X, \cdot)$, $Q_2 = (X, \circ)$ be standard division sudokus whose tables extend the partial division sudoku table of Figure 1. Then a ds-isotopism (α, β, γ) from Q_1 to Q_2 is determined by the values $\alpha^{-1}(1)$, $\alpha^{-1}(2)$, $\beta^{-1}(1)$, $\beta^{-1}(2)$, by the multiplication table of Q_1 and by the partial multiplication table of Q_2 given in Figure 1.*

Proof. Let $/, \backslash$ be the division operations in Q_1, which are certainly determined by the multiplication in Q_1. The permutation α maps blocks to blocks and hence the knowledge of $\alpha^{-1}(1), \alpha^{-1}(2)$ determines $\alpha^{-1}(3)$. The entry $\beta^{-1}(3)$ is determined similarly from $\beta^{-1}(1)$, $\beta^{-1}(2)$. We have $\alpha(x) \circ \beta(y) = \gamma(x \cdot y)$ and thus $\gamma^{-1}(x \circ y) = \alpha^{-1}(x) \cdot \beta^{-1}(y)$ for every x, y. With $x, y \in \{1, 2, 3\}$, the values $\alpha^{-1}(x) \cdot \beta^{-1}(y)$ and $x \circ y$ are known and therefore determine $\gamma^{-1}(z)$ for every z, using the sudoku property in the top left minisquare of Q_2. We can now determine β by focusing on the rest of the top band of Q_2. For every $4 \leq y \leq 9$, there is a unique cell (x, y) with $1 \leq x \leq 3$ for which $x \circ y$ is given in the partial table. For instance, $1 \circ 4 = 2$, so $\alpha^{-1}(1) \cdot \beta^{-1}(4) = \gamma^{-1}(2)$, which yields $\beta^{-1}(4) = \alpha^{-1}(1) \backslash \gamma^{-1}(2)$. The permutation α is determined dually by focusing on the left stack of Q_2. □

4.2. The enumeration.

LEMMA 4.5. *Every standard division sudoku of rank 3 is ds-isotopic to a standard division sudoku representative that extends the partial table in Figure 1. The ds-isotopism class of each of these representatives consists of $2^{11} \cdot 3^8 = 13,436,928$ standard division sudokus.*

Proof. Using ds-isotopisms, we will bring a given standard division sudoku into the desired form while keeping track of the number of division sudokus identified in the process.

First consider the minisquare $M_{1,1} = B_1 \cap S_1$. Either the symbols on the main diagonal of $M_{1,1}$ form a block, or the symbols on the main antidiagonal of $M_{1,1}$ form a block. By permuting the three blocks of symbols and then permuting within each of the blocks of symbols, we can therefore assume that $M_{1,1}$ takes on one of the following two forms:

$$
\begin{matrix}
1\ 4\ 7 & & 1\ 7\ 4 \\
8\ 2\ 5 & \text{or} & 8\ 5\ 2\,. \\
6\ 9\ 3 & & 6\ 3\ 9
\end{matrix}
$$

We have identified $|\mathrm{Sym}(3)|^4 = 6^4$ division sudokus so far. By applying the column permutation $(2, 3)$ to the second form, we obtain the first form, contributing another factor of $|C_2| = 2$.

By permuting the columns 4, 5, and 6, we can assume that the symbols on the main diagonal of the minisquare $M_{1,2} = B_1 \cap S_2$ form the block $\{1, 2, 3\}$. By permuting the columns 7, 8, 9, we can assume that the symbols on the main diagonal of the minisquare $M_{1,3} = B_1 \cap S_3$ form the block $\{1, 2, 3\}$. By exchanging the column blocks $\{4, 5, 6\}$ and $\{7, 8, 9\}$, if necessary, we can assume that B_1 is as in Figure 1, taking advantage of the latin property in the top three rows. The factor here is $|\mathrm{Sym}(3)|^2 \cdot |C_2| = 6^2 \cdot 2$.

We proceed analogously in the stack S_1, encountering another factor of $|\mathrm{Sym}(3)|^2 \cdot |C_2| = 6^2 \cdot 2$. □

LEMMA 4.6 (🖥). *There are* 7741 *standard division sudokus extending the partial table in Figure* 1.

Proof. This can be shown in a few seconds with a finite model builder. We used Mace4 [15] with the following input file to search for all models of size 9. The predicate B keeps track of the blocks of the sudoku partition. Note that Mace4 labels elements in models of size n with $\{0, \ldots, n-1\}$. The division sudoku conditions in the input file correspond to those of Proposition 2.3.

```
% latin square (quasigroup)
x*(x\y) = y.    x\(x*y) = y.    (x*y)/y = x.    (x/y)*y = x.
% predicate B defines standard blocks
B(0)=0. B(1)=0. B(2)=0. B(3)=1. B(4)=1. B(5)=1. B(6)=2. B(7)=2. B(8)=2.
% division sudoku conditions
all x all y all u all v ((B(x)=B(u) & B(y)=B(v) & x*y = u*v)->(x=u & y=v)).
all x all y all u all v ((B(x)=B(u) & B(y)=B(v) & x/y = u/v)->(x=u & y=v)).
all x all y all u all v ((B(x)=B(u) & B(y)=B(v) & x\y = u\v)->(x=u & y=v)).
% partial table
0*0=0. 0*1=3. 0*2=6. 1*0=7. 1*1=1. 1*2=4. 2*0=5. 2*1=8. 2*2=2.
0*3=1. 1*4=2. 2*5=0. 0*6=2. 1*7=0. 2*8=1.
3*0=1. 4*1=2. 5*0=2. 6*0=2. 7*1=0. 8*2=1.
```

□

Since 7741 happens to be a prime number, we deduce that it is not possible to further identify division sudokus in the proof of Lemma 4.5 without splitting the argument into cases.

THEOREM 4.7 (☞). *There are* $104, 015, 259, 648 = 2^{11} \cdot 3^8 \cdot 7741$ *standard division sudokus of rank* 3. *There are* 186 *standard division sudokus of rank* 3 *up to ds-isotopism. There are* 45 *standard division sudokus of rank* 3 *up to ds-paratopism.*

Proof. The first result follows from Lemmas 4.5 and 4.6.

The 7741 representatives from Lemma 4.6 must be filtered up to ds-isotopism. In general, filtering thousands of quasigroups of order 9 up to isotopism is a nontrivial problem. Even the case of ds-isotopisms is hard, since there are $1296 = 6^4$ permutations in Sym(9) that preserve the standard partition. We could take two of the 7741 quasigroups at a time and check for a ds-isotopism using Lemma 4.3. Each of these checks is fast since only the parameters $\alpha(1), \alpha(2), \beta(1)$, and $\beta(2)$ need to be given, but there are $\binom{7741}{2}$ checks to be conducted.

Taking advantage of Lemma 4.4 is much faster. Let $Q_1 = (X, \cdot)$ be one of the 7741 division sudokus. We choose $\alpha^{-1}(1)$ arbitrarily and then $\alpha^{-1}(2)$ in the same block as $\alpha^{-1}(1)$. Similarly for $\beta^{-1}(1)$ and $\beta^{-1}(2)$. We then complete α, β, and γ from the knowledge of Q_1 and the partial table of Q_2 as in the proof of Lemma 4.4. If the resulting maps are partition preserving permutations, we set $x \circ y = \gamma(\alpha^{-1}(x) \cdot \beta^{-1}(y))$, hence constructing the unique quasigroup $Q_2 = (X, \circ)$ such that (α, β, γ) is a ds-isotopism from Q_1 to Q_2. If Q_2 is among the 7741 representatives, we add it to the ds-isotopism class of Q_1. We implemented this algorithm in GAP. In 3 s, the algorithm returns 186 representatives with classes of sizes 1 (once), 3 (9 times), 6 (3 times), 9 (22 times), 18 (7 times), 27 (15 times), and 54 (129 times).

Finally, for main ds-classes, we first subdivide the 186 representatives according to the intercalate structure invariant, cf. Section 3. Then, given a subset of representatives with the same invariant, we check if any two lie in the same main ds-class by generating all six conjugates and checking for a ds-isotopism. Note that the conjugates do not necessarily have the form of Figure 1 and we therefore cannot use Lemma 4.4, but Lemma 4.3 applies. We obtained 45 representatives in about 1 min. □

The 186 standard division sudokus of rank 3 up to ds-isotopism are given in the Appendix, labeled DS(9, 1), ..., DS(9, 186). These are the pairwise structurally different division sudokus of rank 3 when the roles of rows, columns, and symbols are fixed.

REMARK 4.8. Each division sudoku DS(9, i) is ds-isotopic to an idempotent quasigroup, that is, a quasigroup in which the identity $x \cdot x = x$ holds. (It suffices to permute the rows and columns in the second block and the rows and columns in the third block.) For convenience, we will sometimes work with idempotent versions of the division sudoku DS(9, i).

Table 1 lists the 45 main ds-classes of standard division sudokus of rank 3. These are the pairwise structurally different division sudokus of rank 3 when the roles of rows, columns, and symbols are allowed to be permuted.

Finally, it is reasonable to ask which of the 186 standard division sudokus of rank 3 up to ds-isotopism are isotopic as quasigroups. Using Lemma 4.3, several hours of computing time yield:

PROPOSITION 4.9 (☞). *There are* 183 *standard division sudokus of rank* 3 *up to isotopism. Among the* 186 *standard division sudokus of rank* 3 *up to ds-isotopism, only the following pairs are isotopic:* DS(9, 18) *and* DS(9, 19), DS(9, 21) *and* DS(9, 60), *and* DS(9, 33) *and* DS(9, 117).

Table 1. The main ds-classes of standard division sudokus of rank 3. Integers i, j are listed in the same set if and only if $DS(9, i)$, $DS(9, j)$ belong to the same main ds-class.

{1, 53, 135}	{2, 35, 156}	{3, 40, 42, 149, 163, 180}
{4}	{5, 24, 46, 64, 123, 126}	{6, 12, 14, 29, 43, 102}
{7, 37, 157}	{8, 32, 47, 142, 161, 172}	{9, 10, 49}
{11, 31, 52, 134, 159, 171}	{13, 26, 56, 65, 125, 127}	{15, 39, 58, 73, 98, 101}
{16, 55, 150}	{17}	{18, 19, 21, 33, 60, 117}
{20, 62, 131}	{22, 34, 61, 75, 118, 119}	{23, 36, 63, 132, 158, 170}
{25, 28, 67, 100, 111, 113}	{27, 69, 148}	{30, 41, 80, 155, 164, 181}
{38, 45, 103}	{44, 48, 77, 109, 137, 139}	{50, 54, 133}
{51}	{57, 140, 174}	{59, 99, 107}
{66, 124, 168}	{68}	{70, 130, 165, 166, 173, 177}
{71, 122, 162, 169, 184, 185}	{72, 78, 83, 95, 96, 97}	{74, 82, 85, 104, 138, 145}
{76, 120}	{79, 84, 93, 116, 151, 152}	{81, 115, 146}
{86, 88, 106, 112, 143, 147}	{87, 144, 178}	{89, 92, 110, 114, 141, 154}
{90, 153, 182}	{91, 105, 121, 129, 160, 186}	{94, 108, 128, 136, 167, 176}
{175}	{179}	{183}

4.3. The distinguishing power of the invariants. The two invariants introduced in Section 3 are quite powerful. By calculating the invariants for the 186 standard division sudokus of rank 3 up to ds-isotopism, we obtain the following.

PROPOSITION 4.10 (⌨). *The intercalate structure invariant ι attains precisely* 148 *distinct values on the set of standard division sudokus of rank* 3. *The minisquare structure invariant μ attains precisely* 139 *distinct values on the set of standard division sudokus of rank* 3. *The combined invariant* (ι, μ) *attains precisely* 183 *distinct values on the set of standard division sudokus of rank* 3.

Only the following pairs of division sudokus cannot be distinguished by the two invariants:

$DS(9, 91)$ and $DS(9, 105)$, $DS(9, 121)$ and $DS(9, 129)$, and $DS(9, 160)$ and $DS(9, 186)$.

These six division sudokus belong to the same main ds-class. Let us have a closer look at them.

We start with the division sudoku $(Q, \cdot)$ next which is ds-isotopic to $DS(9, 121)$. For the convenience of the reader, we have also displayed the right division $/$ and the left division $\backslash$ for $(Q, \cdot)$.

```
 ·   1 2 3   4 5 6   7 8 9
 1   1 6 7   8 3 4   9 2 5
 2   8 2 4   5 9 1   6 7 3
 3   5 9 3   2 6 7   1 4 8

 4   9 1 5   4 7 2   8 3 6
 5   6 7 2   3 5 8   4 9 1
 6   3 4 8   9 1 6   2 5 7

 7   2 5 9   6 8 3   7 1 4
 8   7 3 6   1 4 9   5 8 2
 9   4 8 1   7 2 5   3 6 9
```

```
 /   1 2 3   4 5 6   7 8 9
 1   1 4 9   8 6 2   3 7 5
 2   7 2 5   3 9 4   6 1 8
 3   6 8 3   5 1 7   9 4 2

 4   9 6 2   4 8 1   5 3 7
 5   3 7 4   2 5 9   8 6 1
 6   5 1 8   7 3 6   2 9 4

 7   8 5 1   9 4 3   7 2 6
 8   2 9 6   1 7 5   4 8 3
 9   4 3 7   6 2 8   1 5 9
```

```
 \   1 2 3   4 5 6   7 8 9
 1   1 8 5   6 9 2   3 4 7
 2   6 2 9   3 4 7   8 1 5
 3   7 4 3   8 1 5   6 9 2

 4   2 6 8   4 3 9   5 7 1
 5   9 3 4   7 5 1   2 6 8
 6   5 7 1   2 8 6   9 3 4

 7   8 1 6   9 2 4   7 5 3
 8   4 9 2   5 7 3   1 8 6
 9   3 5 7   1 6 8   4 2 9
```

The following result can be verified by inspection of the multiplication table $(Q, \cdot)$. The main diagonal of the table will be called just *diagonal*, and a minisquare intersecting the diagonal will be called a *diagonal minisquare*.

LEMMA 4.11. *The quasigroup $(Q, \cdot)$ is idempotent and contains exactly 18 intercalates. Each intercalate contains exactly one cell from the diagonal. For each diagonal cell, there are exactly two intercalates that include that cell. In each intercalate, the corner opposite from the diagonal cell lies in a diagonal minisquare. Each nondiagonal cell that is inside a diagonal minisquare is contained in exactly one intercalate.*

LEMMA 4.12. *Suppose that (α, β, γ) is an isotopism of $(Q, \cdot)$ onto either $(Q, \cdot)$ or the conjugate $(Q, \cdot)^{(1,2)}$. Then (α, β, γ) is a ds-isotopism and $\alpha = \beta = \gamma$.*

Proof. The intercalate structure of $(Q, \cdot)$ implies that every isotopism sends a diagonal minisquare upon a diagonal minisquare. Hence every isotopism is a ds-isotopism. Since every isotopism sends intercalates onto intercalates and since the only cells with two intercalates through them are on the diagonal, we have $\alpha = \beta$. Using idempotency, we then have $\alpha(x) = \alpha(x) \cdot \alpha(x) = \alpha(x) \cdot \beta(x) = \gamma(x \cdot x) = \gamma(x)$ for each $x \in Q$. Note that the calculation goes through even if the target quasigroup is $(Q, \cdot)^{(1,2)}$ since $\alpha(x) \cdot \beta(x) = \beta(x) \cdot \alpha(x)$ here. $\square$

LEMMA 4.13. *The quasigroup $(Q, \cdot)$ is not isotopic to $(Q, \cdot)^{(1,2)}$.*

Proof. Suppose for a contradiction that $(Q, \cdot)$ is isotopic to $(Q, \cdot)^{(1,2)}$. Then the quasigroups are isomorphic by Lemma 4.12. But that is not possible since all left translations of $(Q, \cdot)$ contain a 4-cycle, while no right translation of $(Q, \cdot)$ contains a 4-cycle. (We remark that the automorphism group of $(Q, \cdot)$ acts transitively on Q.) $\square$

By inspection of the three tables $(Q, \cdot)$, $(Q, /)$ and $(Q, \backslash)$, we observe the following.

LEMMA 4.14. *Let $\mu_{(Q,\cdot)}$ be the minisquare structure invariant of $(Q, \cdot)$. Then the only directed edges of $\mu_{(Q,\cdot)}$ are $P_i \to B_i$ and $P_i \to S_i$, for $1 \le i \le 3$.*

PROPOSITION 4.15. *The intercalate structure invariant and the minisquare structure invariant do not distinguish $(Q, \cdot)$ from $(Q, \cdot)^{(1,2)}$, $(Q, /)$ from $(Q, \backslash)^{(1,2)}$, and $(Q, \backslash)$ from $(Q, /)^{(1,2)}$.*

Proof. By Lemma 4.11, the intercalate structure invariants of $(Q, \cdot)$ and $(Q, \cdot)^{(1,2)}$ coincide. By Lemma 4.14, the minisquare structure invariant of $(Q, \cdot)$ distinguishes piles from bands and stacks, but not bands from stacks. The rest follows by considering the conjugates of $(Q, \cdot)$. $\square$

It can now be checked (by hand) that $(Q, \cdot)$ is indeed ds-isotopic to DS(9, 121), $(Q, \cdot)^{(1,2)}$ is ds-isotopic to DS(9, 129), $(Q, /)$ is ds-isotopic to DS(9, 186), $(Q, /)^{(1,2)}$ is ds-isotopic to DS(9, 105), $(Q, \backslash)$ is ds-isotopic to DS(9, 91), and $(Q, \backslash)^{(1,2)}$ is ds-isotopic to DS(9, 160).

5. Division sudokus with multiple sudoku partitions.

5.1. Sudoku tri-partitions.
So far we have used the same sudoku partition for rows, columns, and symbols. We will now find it convenient to use three separate partitions.

DEFINITION 5.1. A *sudoku tri-partition* is an ordered triple (X^B, X^S, X^P) of sudoku partitions $\{X_1^B, \ldots, X_m^B\}$, $\{X_1^S, \ldots, X_m^S\}$, and $\{X_1^P, \ldots, X_m^P\}$ on the same underlying set X. Given a latin square L on X, the *bands*, *stacks*, and *piles* are defined by

$$B_i = \{(x, y, z) \in O(L) : x \in X_i^B\},$$
$$S_i = \{(x, y, z) \in O(L) : y \in X_i^S\},$$
$$P_i = \{(x, y, z) \in O(L) : z \in X_i^P\},$$

respectively.

A latin square L on X is a *sudoku with respect to a sudoku tri-partition* if the intersection of every band with every stack contains all symbols of X. A latin square L on X is a *division sudoku with respect to a sudoku tri-partition* if the intersection of every band with every stack contains all symbols of X, the intersection of every band with every pile contains all columns of X, and the intersection of every stack with every pile contains all rows of X.

Results from Section 2 remain valid in this more general setting when suitably interpreted. For instance, a latin square L is a division sudoku with respect to the sudoku tri-partition (X^B, X^S, X^P) if and only if the conjugates L, $L^{(1,3)}$, $L^{(2,3)}$ are sudokus with respect to the sudoku tri-partition (X^B, X^S, X^P).

For a latin square L on X, denote by $\pi(L)$ the number of sudoku tri-partitions with respect to which L is a division sudoku. In this subsection, we will determine $\pi(L)$ for every division sudoku L of rank 3.

LEMMA 5.2. *If L, L' are paratopic latin squares then $\pi(L) = \pi(L')$.*

Proof. Suppose that L is a division sudoku with respect to the sudoku tri-partition $(X^B, X^S, X^P) = (X^1, X^2, X^3)$. If $(\alpha_1, \alpha_2, \alpha_3)$ is an isotopism from L to L', then L' is a division sudoku with respect to the sudoku tri-partition (Y^1, Y^2, Y^3), where $Y_i^j = \alpha_j(X_i^j)$ for every $1 \le i \le m$ and $1 \le j \le 3$. Similarly, if $\theta \in \mathrm{Sym}(3)$ and $L' = L^\theta$, then L' is a division sudoku with respect to the sudoku tri-partition $(X^{\theta(1)}, X^{\theta(2)}, X^{\theta(3)})$. □

For instance, $\pi(\mathrm{DS}(9, 18)) = 4$ and we can visualize the situation as follows:

Here, all three latin squares represent $L = \mathrm{DS}(9, 18)$ with the indicated labeling of rows and columns. The first square shows that L is a division sudoku with respect to the tri-partition

$$X^B = \{123\ 456\ 789\}, \quad X^S = \{123\ 456\ 789\}, \quad X^P = \{123\ 456\ 789\}$$

but also with respect to the tri-partition

$$X^B = \{123\ 456\ 789\}, \quad X^S = \{123\ 456\ 789\}, \quad X^P = \{159\ 267\ 348\}.$$

(The first partition $X^P = \{123\ 456\ 789\}$ can be seen on the broken diagonals of the minisquare $M = B_1 \cap S_1$ and then in all minisquares, while the second partition $X^P = \{159\ 267\ 348\}$ can be seen on the broken antidiagonals of the minisquare M and then in all minisquares. It is not always the case that for a given X^B, X^S, there are two suitable

X^Ps obtained in this fashion.) The second square shows that L is a division sudoku with respect to the tri-partition

$$X^B = \{123\ 456\ 789\}, \ X^S = \{147\ 258\ 369\}, \ X^P = \{147\ 258\ 369\}.$$

Finally, the third square shows that L is a division sudoku with respect to the tri-partition

$$X^B = \{123\ 456\ 789\}, \ X^S = \{159\ 267\ 348\}, \ X^P = \{147\ 258\ 369\}.$$

PROPOSITION 5.3 (⌨). *Among the* 45 *main ds-classes of division sudokus of rank* 3 *(cf. Table 1), only* 7 *classes consist of latin squares that are division sudokus with respect to more than* 1 *sudoku tri-partition, namely:*

i	2	17	18	20	27	175	179
$\pi(\mathrm{DS}(9, i))$	2	24	4	9	2	3	4

Proof. Given a latin square L of order 9, all sudoku tri-partitions with respect to which L is a division sudoku can be found quickly as follows. For a given set B of three rows, let $\mathcal{S}(B)$ be the set of all subsets S of columns such that $B \times S$ is a 3×3 latin subsquare of L. There are $\binom{8}{2}\binom{5}{2} = 280$ partitions of rows into three-element subsets. Given a partition (X_1^B, X_2^B, X_3^B) of rows, we consider all partitions of columns (X_1^S, X_2^S, X_3^S) such that $X_i^S \in \mathcal{S}(X_i^B)$. Once (X_1^B, X_2^B, X_3^B) and (X_1^S, X_2^S, X_3^S) are given, there are only two candidates for (X_1^P, X_2^P, X_3^P), one corresponding to the broken diagonals of the minisquare $M = X_1^B \times X_1^S$, and another corresponding to the broken antidiagonals of M. □

Table 2 lists all sudoku tri-partitions with respect to which the latin squares of Proposition 5.3 are division sudokus.

5.2. Synchronizing tri-partitions.   A sudoku tri-partition (X^B, X^S, X^P) is *synchronized* if $X^B = X^S = X^P$. Of course, a synchronized sudoku tri-partition can be identified with a sudoku partition.

For a latin square L denote by $\sigma(L)$ the number of synchronized sudoku tri-partitions with respect to which L is a division sudoku. In other words, $\sigma(L)$ is the number of sudoku partitions with respect to which L is a division sudoku.

The value of σ is preserved under conjugation but not necessarily under ds-isotopisms. For a ds-isotopism class $\mathcal{C}$ with respect to the standard sudoku partition, let $\sigma(\mathcal{C}) = \max\{\sigma(L) : L \in \mathcal{C}\}$. Our task is to determine $\sigma(\mathcal{C})$ for every ds-isotopism class $\mathcal{C}$ and to find at least one $L \in \mathcal{C}$ such that $\sigma(L) = \sigma(\mathcal{C})$.

If $L = \mathrm{DS}(9, i) \in \mathcal{C}$ for some $1 \le i \le 186$, then $\sigma(L) \ge 1$ on account of the standard sudoku partition of L. It can certainly happen that $\sigma(L) < \pi(L)$ and, in fact, $\sigma(\mathcal{C}) < \pi(L)$.

PROPOSITION 5.4 (⌨). *For* $1 \le i \le 186$*, let* $\mathcal{C}_i$ *be the ds-isotopism class of the division sudoku* $\mathrm{DS}(9, i)$*. Then* $\sigma(\mathcal{C}_i) = 1$ *except in the cases* $\sigma(\mathcal{C}_{17}) = 4$*,* $\sigma(\mathcal{C}_{175}) = 3$ *and* $\sigma(\mathcal{C}_{179}) = 4$*.*

Proof. Let L be a division sudoku with sudoku tri-partitions $X = (X^B, X^S, X^P)$, $Y = (Y^B, Y^S, Y^P)$, $Z = (Z^B, Z^S, Z^P)$, etc. Suppose without loss of generality that $X^B = X^S = X^P$ is the standard sudoku partition.

Suppose that we wish to find a ds-isotopic copy L' of L for which X is still a (synchronized) sudoku tri-partition and the image of Y is also synchronized. Let G be the subgroup of $\mathrm{Sym}(9)$ that preserves the partition X^B. This group acts naturally on the 280-element set of sudoku partitions, and $G \times G \times G$ acts naturally on the set of sudoku tri-partitions

Table 2. All sudoku tri-partitions with respect to which a given latin square is a division sudoku.

DS(9,2)	DS(9,18)
({123 456 789}, {123 456 789}, {123 456 789})	({123 456 789}, {123 456 789}, {123 456 789})
({123 456 789}, {123 456 789}, {159 267 348})	({123 456 789}, {123 456 789}, {159 267 348})
	({123 456 789}, {147 258 369}, {147 258 369})
DS(9,17)	({123 456 789}, {159 267 348}, {147 258 369})
({123 456 789}, {123 456 789}, {123 456 789})	
({123 456 789}, {123 456 789}, {159 267 348})	
({123 456 789}, {147 258 369}, {159 267 348})	DS(9,20)
({123 456 789}, {147 258 369}, {147 258 369})	({123 456 789}, {123 456 789}, {123 456 789})
({123 456 789}, {159 267 348}, {123 456 789})	({123 456 789}, {125 389 467}, {127 346 589})
({123 456 789}, {159 267 348}, {147 258 369})	({123 456 789}, {127 346 589}, {179 236 458})
({147 258 369}, {123 456 789}, {159 267 348})	({123 456 789}, {134 278 569}, {139 256 478})
({147 258 369}, {123 456 789}, {168 249 357})	({123 456 789}, {139 256 478}, {125 389 467})
({147 258 369}, {159 267 348}, {168 249 357})	({123 456 789}, {145 238 679}, {134 278 569})
({147 258 369}, {159 267 348}, {147 258 369})	({123 456 789}, {147 258 369}, {159 267 348})
({147 258 369}, {168 249 357}, {159 267 348})	({123 456 789}, {159 267 348}, {147 258 369})
({147 258 369}, {168 249 357}, {147 258 369})	({123 456 789}, {179 236 458}, {145 238 679})
({159 267 348}, {123 456 789}, {123 456 789})	
({159 267 348}, {123 456 789}, {168 249 357})	DS(9,27)
({159 267 348}, {147 258 369}, {147 258 369})	({123 456 789}, {123 456 789}, {123 456 789})
({159 267 348}, {147 258 369}, {168 249 357})	({123 456 789}, {147 258 369}, {147 258 369})
({159 267 348}, {168 249 357}, {147 258 369})	
({159 267 348}, {168 249 357}, {123 456 789})	DS(9,175)
({168 249 357}, {147 258 369}, {159 267 348})	({123 456 789}, {123 456 789}, {123 456 789})
({168 249 357}, {147 258 369}, {168 249 357})	({147 258 369}, {169 247 358}, {148 259 367})
({168 249 357}, {159 267 348}, {168 249 357})	({168 249 357}, {148 259 367}, {169 247 358})
({168 249 357}, {159 267 348}, {123 456 789})	
({168 249 357}, {168 249 357}, {123 456 789})	DS(9,179)
({168 249 357}, {168 249 357}, {159 267 348})	({123 456 789}, {123 456 789}, {123 456 789})
	({149 257 368}, {167 248 359}, {158 269 347})
	({158 269 347}, {158 269 347}, {167 248 359})
	({167 248 359}, {149 257 368}, {149 257 368})

with X a fixed point. Our task is to find f, g, $h \in G$ such that $(Y^B)^f = (Y^S)^g = (Y^P)^h$, which happens if and only if $Y^B = (Y^S)^{gf^{-1}} = (Y^P)^{hf^{-1}}$. It is therefore possible to synchronize Y while stabilizing X if and only if Y^B, Y^S, Y^P are in the same orbit of G.

If Y cannot be synchronized while stabilizing X, we backtrack. Otherwise we find g, $h \in G$ such that $Y^B = (Y^S)^g = (Y^P)^h$ and we let L' be the uniquely determined latin square for which $(1, g, h)$ is a ds-isotopism from L to L'. Then L' is a division sudoku with respect to the tri-partitions $X' = X$, $Y' = (Y^B, Y^B, Y^B)$, $Z' = (Z^B, (Z^S)^g, (Z^P)^h)$, etc, and both X', Y' are synchronized. Attempting to synchronize Z', we replace the group G with the stabilizer of both X^B and Y^B, and we proceed as above. $\square$

The algorithm described in the proof of Proposition 5.4 determines $\sigma(\mathcal{C}_i)$ and $L_i \in \mathcal{C}_i$ with $\sigma(L_i) = \sigma(\mathcal{C}_i)$ in a matter of seconds. In particular, it finds the latin squares

L_{17}

```
5 2 8   7 4 1   3 9 6
9 6 3   2 8 5   4 1 7
1 7 4   6 3 9   8 5 2

6 3 9   8 5 2   1 7 4
7 4 1   3 9 6   5 2 8
2 8 5   4 1 7   9 6 3

4 1 7   9 6 3   2 8 5
8 5 2   1 7 4   6 3 9
3 9 6   5 2 8   7 4 1
```

L_{175}

```
2 5 8   4 9 1   3 7 6
9 3 6   2 5 7   4 1 8
4 7 1   8 3 6   9 5 2

6 1 9   5 8 2   7 3 4
7 4 2   3 6 9   5 8 1
3 8 5   7 1 4   2 6 9

1 6 7   9 4 3   8 2 5
8 2 4   1 7 5   6 9 3
5 9 3   6 2 8   1 4 7
```

L_{179}

```
7 1 4   9 5 3   8 2 6
5 8 2   1 7 6   4 9 3
3 6 9   4 2 8   1 5 7

8 3 5   2 4 9   7 6 1
6 9 1   7 3 5   2 8 4
2 4 7   6 8 1   5 3 9

9 5 3   8 1 4   6 7 2
1 7 6   5 9 2   3 4 8
4 2 8   3 6 7   9 1 5
```

The square $L_{17} \in \mathcal{C}_{17}$ is a division sudoku with respect to each of the following four sudoku partitions:

$$\{123\ 456\ 789\}, \ \{147\ 258\ 369\}, \ \{159\ 267\ 348\}, \ \{168\ 249\ 357\}.$$

The square $L_{175} \in \mathcal{C}_{175}$ is a division sudoku with respect to each of the following three sudoku partitions:

$$\{123\ 456\ 789\}, \ \{147\ 258\ 369\}, \ \{168\ 249\ 357\}.$$

The square $L_{179} \in \mathcal{C}_{179}$ is a division sudoku with respect to each of the following four sudoku partitions:

$$\{123\ 456\ 789\}, \ \{149\ 257\ 368\}, \ \{158\ 269\ 347\}, \ \{167\ 248\ 359\}.$$

REMARK 5.5. The division sudoku L_0 from the introduction (which is ds-isotopic to DS(9, 179)) is a division sudoku with respect to exactly the same sudoku partitions as L_{17}. Upon considering the ds-isotopism $((7, 8, 9), (7, 8, 9), (7, 8, 9))$, we can transform L_{179} into a square L'_{179} which also has exactly the same sudoku partitions as L_{17}.

For $m \geq 3$, let

$$\sigma(m) = \max\{\sigma(L) : L \text{ is a division sudoku of rank } m\}$$

be the maximum number of division sudoku partitions that a division sudoku of rank m can possess.

COROLLARY 5.6. $\sigma(3) = 4$.

6. Division sudokus constructed from fields and nearfields.
In this section, we construct division sudokus of prime power rank that are rich in division sudoku partitions.

6.1. Division sudokus constructed from fields.
For a prime power q, let $\mathbb{F}_q$ be the field of order q. Given $c \in \mathbb{F}_{q^2} \setminus \{0, 1\}$, define a multiplication $*$ on $\mathbb{F}_{q^2}$ by

$$x * y = x + (y - x)c. \tag{6.1}$$

It is then easy to see that $(\mathbb{F}_{q^2}, *)$ is a quasigroup with left and right divisions given by

$$x \backslash y = x + (y - x)c^{-1}, \quad x/y = x + (y - x)c(c - 1)^{-1} = y + (x - y)(1 - c)^{-1},$$

respectively.

DEFINITION 6.1. Let q be a prime power and $c \in \mathbb{F}_{q^2} \setminus \{0, 1\}$. Then the quasigroup $(\mathbb{F}_{q^2}, *)$ with multiplication (6.1) will be denoted by DS($\mathbb{F}_{q^2}, c$).

We will show that $DS(\mathbb{F}_{q^2}, c)$ can be equipped with a number of sudoku partitions with respect to which it is a division sudoku.

LEMMA 6.2. *Let V be a field seen as a vector space over some subfield F. Let W be an F-subspace of V and $c \in V \setminus \{0, 1\}$. Then the following conditions are equivalent: $W \cap Wc = 0$, $W \cap W(c - 1) = 0$, and $Wc \cap W(c - 1) = 0$.*

Proof. Let w, w' be elements of W. Note that $w' = wc$ if and only if $w' - w = w(c - 1)$, and $w'(c - 1) = wc$ if and only if $w' = (w' - w)c$. □

PROPOSITION 6.3. *Let r be a prime power, $q = r^s$, $V = \mathbb{F}_{q^2}$ a vector space over $\mathbb{F}_r$ and W an $\mathbb{F}_r$-subspace of V of dimension s. Let $c \in V$ be such that $W \cap Wc = 0$. For x, $y \in V$, write $x \sim y$ if and only if $x - y \in W$. Then $DS(\mathbb{F}_{q^2}, c)$ is a division sudoku of rank q with respect to the partition induced by the equivalence relation $\sim$.*

Proof. Since $|V| = q^2$ and every coset of W has cardinality $q = r^s$, the partition induced by $\sim$ is a sudoku partition with blocks of size q. We will use Proposition 2.5 to check that $DS(\mathbb{F}_{q^2}, c)$ is a division sudoku with respect to $\sim$. Suppose that $u \sim v$, i.e., $u - v \in W$.

If $x/u \sim x/v$, then $(x + (u - x)c(c - 1)^{-1}) - (x + (v - x)c(c - 1)^{-1}) = w \in W$, so $(u - v)c = w(c - 1) \in Wc \cap W(c - 1) = 0$ by Lemma 6.2. If $x * u \sim x * v$, then $(x + (u - x)c) - (x + (v - x)c) = w' \in W$, so $(u - v)c = w' \in W \cap Wc = 0$. If $u * x \sim v * x$, then $(u + (x - u)c) - (v + (x - v)c) = w'' \in W$, so $(u - v)(1 - c) = w'' \in W \cap W(1 - c) = 0$ by Lemma 6.2. In each case, we conclude that $u = v$. □

THEOREM 6.4. *Let q be a prime power and $c \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$. Then $L = DS(\mathbb{F}_{q^2}, c)$ is a division sudoku of rank q satisfying $\sigma(L) \geq q + 1$. In particular, $\sigma(q) \geq q + 1$.*

Proof. Consider $V = \mathbb{F}_{q^2}$ as a vector space over $\mathbb{F}_q$ and let W be a one-dimensional $\mathbb{F}_q$-subspace of V. Note that there are $(q^2 - 1)/(q - 1) = q + 1$ such subspaces. We claim that $W \cap Wc = 0$. Indeed, we have $W = w\mathbb{F}_q$ for some $0 \neq w \in V$ and if $W \cap Wc \neq 0$ then $w\lambda = w\mu c$ for some λ, $\mu \in \mathbb{F}_q$, which implies $c = \mu^{-1}\lambda \in \mathbb{F}_q$, a contradiction. The rest follows from Proposition 6.3 with $r = q$ (and thus with $s = 1$). □

The division sudokus $DS(\mathbb{F}_{q^2}, c)$ from Theorem 6.4 are isotopic to the elementary abelian group $C_q \times C_q$. For $c \in \mathbb{F}_{3^2} \setminus \mathbb{F}_3$, the division sudoku $DS(\mathbb{F}_{3^2}, c)$ is ds-isotopic to the division sudoku L_{17} from the proof of Proposition 5.4 and also to the example in [2, Figure 5].

We proceed to find more suitable subspaces and thus more sudoku partitions in quartic field extensions.

PROPOSITION 6.5. *Let r be a prime power, $q = r^2$ and $\mathbb{F}_r \leq \mathbb{F}_q \leq \mathbb{F}_{q^2}$. Let $c \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$. Then there are exactly $q^2 + q$ two-dimensional $\mathbb{F}_r$-subspaces W of $\mathbb{F}_{q^2}$ such that $W \cap Wc = 0$.*

Proof. Let Y be the set of all two-dimensional $\mathbb{F}_r$-subspaces W of $V = \mathbb{F}_{q^2}$. Then

$$|Y| = \frac{(r^4 - 1)(r^4 - r)}{(r^2 - 1)(r^2 - r)} = r^4 + r^3 + 2r^2 + r + 1.$$

Let $Y' = \{v\mathbb{F}_q : v \in V\}$ be the set of all one-dimensional $\mathbb{F}_q$-subspaces W' of V. We have

$$|Y'| = \frac{r^4 - 1}{r^2 - 1} = r^2 + 1.$$

The cyclic group $\langle c \rangle$ acts on Y and Y' by multiplication. Note that $W' \cap W'c = 0$ for every $W' \in Y'$, else $v\mathbb{F}_q \cap v\mathbb{F}_q c \neq 0$ implies $c \in \mathbb{F}_q$, a contradiction.

Consider a "bad" subspace $W \in Y$ such that $W \cap Wc \neq 0$, say $W \cap Wc = W_1 c$ for some one-dimensional $\mathbb{F}_r$-subspace W_1 of W. We have $W_1 c \neq W_1$ (else $c \in \mathbb{F}_r$) and hence $W = W_1 + W_1 c$. Since $\bigcup Y' = V$, there is $W' \in Y'$ such that $W_1 \leq W'$. Summarizing, we have $W = W_1 + W_1 c$, where W_1 is a one-dimensional $\mathbb{F}_r$-subspace of some $W' \in Y'$.

Conversely, let $W = W_1 + W_1 c$, where W_1 is a one-dimensional $\mathbb{F}_r$-subspace of some $W' \in Y'$. Then $W \in Y$ is bad since $W \cap Wc = (W_1 + W_1 c) \cap (W_1 c + W_1 c^2) \neq 0$.

Let us count the bad subspaces. We have $|Y'| = r^2 + 1$ choices of W' and then $(r^2 - 1)/(r - 1) = r + 1$ choices of W_1 in W'. Note that $W_1 \neq W_1 c^2$ (else $c^2 \in \mathbb{F}_r$ and $c \in \mathbb{F}_q$). Hence we did not count any bad subspace more than once. Altogether, there are $|Y| - (r^2 + 1)(r + 1) = r^4 + r^2$ subspaces $W \in Y$ such that $W \cap Wc = 0$. $\square$

Combining Propositions 6.3 and 6.5, we obtain the following.

THEOREM 6.6. *Let $q = p^{2s}$ be a prime power and $c \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$. Then $L = \mathrm{DS}(\mathbb{F}_{q^2}, c)$ is a division sudoku of rank q satisfying $\sigma(L) \geq q^2 + q = p^{4s} + p^{2s}$. In particular, $\sigma(p^{2s}) \geq p^{4s} + p^{2s}$.*

Applying Theorem 6.6 with $q = 4$, we obtain a division sudoku L of rank 4 with $\sigma(L) \geq 20$.

6.2. Division sudokus constructed from nearfields. A (*left*) *nearfield* is an algebra $(F, +, \circ, 0, 1)$ such that $(F, +, 0)$ is an abelian group, $(F, \circ, 1)$ is a monoid, every nonzero element x has an inverse $\bar{x}$ such that $x \circ \bar{x} = \bar{x} \circ x = 1$, and $x \circ (y + z) = (x \circ y) + (x \circ z)$ holds for every $x, y, z \in F$. All finite nearfields were constructed by Dickson and their classification was completed by Zassenhaus [23].

We will use the following lemma without reference.

LEMMA 6.7. *Let F be a nearfield, then*

(i) $0 \circ x = x \circ 0 = 0$ *for every $x \in F$,*
(ii) $(-1) \circ x = -x = x \circ (-1)$ *for every $x \in F$,*
(iii) $(-x) \circ y = x \circ (-y) = -(x \circ y)$ *for every $x, y \in F$.*

Proof. For (i) and (ii), see [23]. Then $-(x \circ y) = (-1) \circ x \circ y$ is equal to $(-x) \circ y$ and also to $x \circ (-1) \circ y = x \circ (-y)$. $\square$

Stein proved in [21, Theorem 2.5] that for every *finite* nearfield $(F, +, \circ, 0, 1)$ and every $c \in F \setminus \{0, 1\}$, the groupoid $(F, *)$ defined by

$$x * y = x + (y - x) \circ c \tag{6.2}$$

is a quasigroup. (We remark that the conclusion remains true for infinite nearfields as well. The left division is given by $x \backslash y = x + (y - x) \circ \bar{c}$, and the right division by $x / y = x + (y - x) \circ (1 - \overline{1 - c})$, cf. [8].)

DEFINITION 6.8. Let q be an odd prime power. Define $\mathbb{D}_{q^2} = (\mathbb{F}_{q^2}, +, \circ, 0, 1)$ by modifying the multiplication in the field $\mathbb{F}_{q^2}$ as follows:

$$x \circ y = \begin{cases} xy, & \text{if } x \text{ is a square in } \mathbb{F}_{q^2}, \\ xy^q, & \text{otherwise,} \end{cases}$$

where the power y^q is taken in $\mathbb{F}_{q^2}$. Then $\mathbb{D}_{q^2}$ is a nearfield, so called *quadratic nearfield*.

It is not difficult to see that $\lambda x = \lambda \circ x = x \circ \lambda$ for every $\lambda \in \mathbb{F}_q$ and $x \in \mathbb{D}_{q^2}$.

DEFINITION 6.9. Let q be an odd prime power, $\mathbb{D}_{q^2} = (\mathbb{F}_{q^2}, +, \circ, 0, 1)$ the quadratic nearfield, and $c \in \mathbb{F}_{q^2} \setminus \{0, 1\}$. Then the quasigroup $(\mathbb{D}_{q^2}, *)$ with Stein's multiplication (6.2) will be denoted by $\mathrm{DS}(\mathbb{D}_{q^2}, c)$.

LEMMA 6.10. *Let $a, b \in \mathbb{F}_{q^2}$, $c \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$, and $0 \neq w \in \mathbb{F}_{q^2}$. Then the following conditions hold in $\mathbb{D}_{q^2}$:*

(i) *If $a + b \in \mathbb{F}_q$ and $a \circ c + b \circ c \in \mathbb{F}_q$ then $a + b = 0$.*
(ii) *If $a + b \in w\mathbb{F}_q$ and $a \circ c + b \circ c \in w\mathbb{F}_q$ then $a + b = 0$.*

Proof. (i) If both a, b are squares in $\mathbb{F}_{q^2}$ then $(a + b)c = ac + bc = a \circ c + b \circ c \in \mathbb{F}_q$, which implies $a + b = 0$, otherwise we deduce from $0 \neq a + b \in \mathbb{F}_q$ and $(a + b)c \in \mathbb{F}_q$ than $c \in \mathbb{F}_q$, a contradiction.

If both a, b are nonsquares then $a \circ c + b \circ c = ac^q + bc^q = (a + b)c^q \in \mathbb{F}_q$. But then $\mathbb{F}_q$ contains $((a + b)c^q)^q = (a + b)^q c^{q^2} = (a + b)c$, too, where we have used $a + b \in \mathbb{F}_q$. We conclude $a + b = 0$ as above.

Now suppose that a is a square and b is a nonsquare, let $\lambda = a + b$ and $d = a \circ c + b \circ c = ac + bc^q \in \mathbb{F}_q$. Then $ac + bc^q = d = d^q = a^q c^q + (\lambda - a^q)c$ and hence $0 = (a + a^q - \lambda)(c - c^q) = (a^q - b)(c - c^q)$. We have $c \neq c^q$ since $c \notin \mathbb{F}^q$ and therefore $a^q = b$, a contradiction with the assumption that a is a square while b is a nonsquare.

(ii) Let v be the multiplicative inverse of w in $\mathbb{D}_{q^2}$, i.e., $v = \overline{w}$. Then the assumptions of (ii) can be rewritten as $v \circ a + v \circ b \in \mathbb{F}_q$ and $(v \circ a) \circ c + (v \circ b) \circ c \in \mathbb{F}_q$, using associativity of $\circ$. By (i), $0 = v \circ a + v \circ b = v \circ (a + b)$, which implies $a + b = 0$. □

PROPOSITION 6.11. *Let q be an odd prime power, $V = \mathbb{F}_{q^2}$ a vector space over $\mathbb{F}_q$, W a one-dimensional $\mathbb{F}_q$-subspace of V, and $c \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$. For $x, y \in V$ write $x \sim y$ if and only if $x - y \in W$. Then $\mathrm{DS}(\mathbb{D}_{q^2}, c)$ is a division sudoku of rank q with respect to the partition induced by the equivalence relation $\sim$.*

Proof. Let $W = w\mathbb{F}_q$ for some $0 \neq w \in V$. We will use (2.1), (2.5), and (2.6) to show that $\mathrm{DS}(\mathbb{D}_{q^2}, c)$ is a division sudoku with respect to $\sim$.

Suppose that $x \sim u$, $y \sim v$, and $x * y = u * v$. Then $x + (y - x) \circ c = u + (v - u) \circ c$ and therefore $(y - x) \circ c + (u - v) \circ c = u - x \in w\mathbb{F}_q$. We also have $(y - x) + (u - v) = (u - x) + (y - v) \in w\mathbb{F}_q$. By Lemma 6.10(ii), $(u - x) + (y - v) = 0$, or, equivalently, $y - x = v - u$. Combining this with $x + (y - x) \circ c = u + (v - u) \circ c$ yields $x = u$ and then $y = v$.

Now suppose that $u \sim v$ and $x * u \sim x * v$. Then $(x + (u - x) \circ c) - (x + (v - x) \circ c) = (u - x) \circ c + (x - v) \circ c$ is in $w\mathbb{F}_q$ and so is $(u - x) + (x - v) = u - v$. By Lemma 6.10(ii), $u = v$.

Finally, suppose that $u \sim v$ and $u * x \sim v * x$. Then $(u + (x - u) \circ c) - (v + (x - v) \circ c)$ is in $w\mathbb{F}_q$ and so is $u - v$. Combining, we deduce that $(x - u) \circ c + (v - x) \circ c \in w\mathbb{F}_q$. We also have $(x - u) + (v - x) = v - u \in w\mathbb{F}_q$, and Lemma 6.10(ii) yields $u = v$. □

THEOREM 6.12. *Let q be an odd prime power and $c \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$. Then $L = \mathrm{DS}(\mathbb{D}_{q^2}, c)$ is a division sudoku satisfying $\sigma(L) \geq q + 1$.*

Proof. Apply Proposition 6.11 to each of the $q + 1$ one-dimensional $\mathbb{F}_q$-subspaces of $\mathbb{F}_{q^2}$. □

For an odd prime power q and $c_1, c_2 \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$, the division sudokus $\mathrm{DS}(\mathbb{F}_{q^2}, c_1)$, $\mathrm{DS}(\mathbb{D}_{q^2}, c_2)$ are not isotopic since the former is isotopic to a group while the latter is

not. For every $c \in \mathbb{F}_{3^2} \setminus \mathbb{F}_3$, the division sudoku $DS(\mathbb{D}_{3^2}, c)$ is ds-isotopic to the division sudoku L_{179} from the proof of Proposition 5.4 and also to the example L_0 from the introduction.

7. Open problems.

PROBLEM 7.1. *Determine the number of standard division sudokus of rank* 4*; absolutely, up to ds-isotopism and up to ds-paratopism.*

PROBLEM 7.2. *For $m > 3$, find lower and upper bounds for the number of standard division sudokus of rank m.*

Recall that $\sigma(L)$ is the number of sudoku partitions with respect to which L is a division sudoku and

$$\sigma(m) = \max\{\sigma(L) : L \text{ is a division sudoku of rank } m\}.$$

Our results imply that $\sigma(3) = 4$, $\sigma(q) \geq q + 1$ for any prime power q and $\sigma(p^{2s}) \geq p^{4s} + p^{2s}$ for any prime p.

PROBLEM 7.3. *Investigate $\sigma(m)$ for $m > 3$.*

For a prime power q and $c \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q$, consider the division sudoku $L = DS(\mathbb{F}_{q^2}, c)$. By Theorem 6.4, $\sigma(L) \geq q + 1$. For a fixed one-dimensional $\mathbb{F}_q$-subspace W of $\mathbb{F}_{q^2}$, the blocks of the corresponding sudoku partition are all the lines in $\mathbb{F}_{q^2}$ parallel to W. Taken together, the blocks of the $q + 1$ sudoku partitions for L form an affine plane. Consequently, two blocks from distinct sudoku partitions of L intersect in precisely one point.

Let C be a collection of sudoku partitions on a latin square L of rank m (not necessarily a prime power) such that L is a division sudoku with respect to each element of C. Then C is said to be *affine* if any two blocks from two distinct sudoku partitions of C intersect in precisely one point.

PROBLEM 7.4. *For a division sudoku L, let $\tau(L)$ be the cardinality of a largest affine collection of sudoku partitions on L. Let*

$$\tau(m) = \max\{\tau(L) : L \text{ is a division sudoku of rank } m\}.$$

What is $\tau(m)$? Is $\tau(m) = N(m) + 2$, where $N(m)$ is the maximal number of mutually orthogonal latin squares of order m?

Let L be a standard division sudoku. A subset S of L is a *critical set* if S determines L as a standard division sudoku and no proper subset of S does.

PROBLEM 7.5. *For $m \geq 3$, determine the cardinality of a smallest critical set among all standard division sudokus of rank m.*

REFERENCES

1. 3Doku, https://www.tellini.org/products/mac/3doku.

2. R. A. Bailey, P. J. Cameron and R. Connelly, Sudoku, gerechte designs, resolutions, affine space, spreads, reguli, and Hamming codes, *Amer. Math. Month.* **115**(5) (2008), 383–404.

3. V. D. Belousov, The group associated with a quasigroup (Russian), *Mat. Issled.* **4/3** (1969), 21–39.

4. R. H. Bruck, *A survey of binary systems*, Ergebnisse der Mathematik und ihrer Grenzgebiete, Neue Folge, Heft 20 (Springer Verlag, Berlin-Göttingen-Heidelberg, 1958).

5. P. J. Cameron, A. J. W. Hilton and E. R. Vaughan, An analogue of Ryser's theorem for partial Sudoku squares, *J. Combin. Math. Combin. Comput.* **80** (2012), 47–69.

6. A. Drápal and P. Lisoněk, *Extreme nonassociativity via nearfields*, accepted to Finite Fields and Applications.

7. A. Drápal and V. Valent, High nonassociativity in order 8 and an associative index estimate, *J. Combin. Des.* **27** (2019), 205–228.

8. A. Drápal and V. Valent, Extreme nonassociativity in order nine and beyond, *J. Combin. Des.*, accepted.

9. C. J. Colbourn and J. H. Dinitz (eds.), *Handbook of combinatorial designs*, Discrete Mathematics and its Applications (Boca Raton), 2nd edition (Chapman and Hall/CRC, Boca Raton, FL, 2007).

10. The GAP Group, GAP – Groups, Algorithm s, and Programming, Version 4.6.3 (2013). http://www.gap-system.org.

11. A. D. Keedwell, Constructions of complete sets of orthogonal diagonal Sudoku squares, *Australas. J. Combin.* **47** (2010), 227–238.

12. T. Kepka, A note on associative triples of elements in cancellation groupoids, *Comment. Math. Univ. Carolin.* **21** (1980), 479–487.

13. T. A. Lambert and P. A. Whitlock, Generalizing Sudoku to three dimensions, *Monte Carlo Methods Appl.* **16**(3–4) (2010), 251–263.

14. J. Lorch, Magic squares and sudoku, *Amer. Math. Month.* **119**(9) (2012), 759–770.

15. W. McCune, Mace4, https://www.cs.unm.edu/~mccune/mace4.

16. G. McGuire, B. Tugemann and G. Civario, There is no 16-clue Sudoku: solving the Sudoku minimum number of clues problem via hitting set enumeration, *Exp. Math.* **23**(2) (2014), 190–217.

17. G. P. Nagy and P. Vojtěchovský, *LOOPS: Computing with quasigroups and loops in GAP*. Available at http://www.math.du.edu/~petr/loops.

18. R. M. Pedersen and T. L. Vis, *Sets of mutually orthogonal Sudoku Latin squares*, College Math. J. **40**(3) (2009), 174–180.

19. E. Russell and F. Jarvis, http://www.afjarvis.staff.shef.ac.uk/sudoku/sudgroup.html.

20. D. Stanovský and P. Vojtěchovský, Commutator theory for loops, *J. Algebra* **399** (2014), 290–322.

21. S. K. Stein, Homogeneous quasigroups, *Pacific J. Math.* **14** (1964), 1091–1102.

22. Su-Doku's maths, http://forum.enjoysudoku.com/su-doku-s-maths-t44.html.

23. H. Zassenhaus, Über endliche Fastkörper (German), *Abh. Math. Sem. Univ. Hamburg* **11**(1) (1935) 187–220.

Appendix: Standard division sudokus of rank 3 up to ds-isotopism

DS(9,1)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 6 9   4 3 7
9 3 6   7 2 4   8 5 1
4 7 1   5 8 3   2 9 6

3 8 4   6 1 7   9 2 5
5 1 9   8 4 2   6 7 3
7 6 2   3 9 5   1 4 8
```

DS(9,2)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 4 7   6 9 3
9 3 6   8 2 5   1 4 7
4 7 1   6 9 3   8 2 5

3 6 9   5 8 2   4 7 1
7 1 4   3 6 9   2 5 8
5 8 2   7 1 4   9 3 6
```

DS(9,3)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 4 7   6 9 3
9 3 6   8 2 5   1 4 7
4 7 1   6 9 3   8 2 5

3 8 4   5 1 9   2 7 6
5 1 9   7 6 2   4 3 8
7 6 2   3 8 4   9 5 1
```

DS(9,4)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 4 9   6 3 7
9 3 6   7 2 5   8 4 1
4 7 1   6 8 3   2 9 5

3 6 9   5 1 7   4 2 8
7 1 4   8 6 2   9 5 3
5 8 2   3 9 4   1 7 6
```

DS(9,5)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 4 9   6 3 7
9 3 6   7 2 5   8 4 1
4 7 1   6 8 3   2 9 5

3 8 4   5 1 7   9 2 6
5 1 9   8 6 2   4 7 3
7 6 2   3 9 4   1 5 8
```

DS(9,6)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 4 9   6 3 7
9 3 6   7 2 5   8 4 1
4 7 1   6 8 3   2 9 5

3 8 4   5 9 2   1 7 6
5 1 9   3 6 7   4 2 8
7 6 2   8 1 4   9 5 3
```

DS(9,7)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 4 9   6 7 3
9 3 6   7 2 5   1 4 8
4 7 1   6 8 3   9 2 5

3 6 9   8 1 4   2 5 7
7 1 4   5 9 2   8 3 6
5 8 2   3 6 7   4 9 1
```

DS(9,8)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 4 9   6 7 3
9 3 6   7 2 5   1 4 8
4 7 1   6 8 3   9 2 5

3 8 4   5 1 9   2 7 6
5 1 9   8 6 2   4 3 7
7 6 2   3 9 4   8 5 1
```

DS(9,9)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 6 9   4 3 7
9 3 6   7 2 4   8 5 1
4 7 1   5 8 3   2 9 6

3 6 9   8 4 2   1 7 5
7 1 4   3 9 5   6 2 8
5 8 2   6 1 7   9 4 3
```

DS(9,10)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 6 9   4 3 7
9 3 6   7 2 4   1 5 8
4 7 1   5 8 3   2 9 6

3 8 4   6 9 2   1 7 5
5 1 9   3 4 7   6 2 8
7 6 2   8 1 5   9 4 3
```

DS(9,11)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 6 9   4 7 3
9 3 6   7 2 4   1 5 8
4 7 1   5 8 3   9 2 6

3 8 4   6 1 7   2 9 5
5 1 9   8 4 2   6 3 7
7 6 2   3 9 5   8 4 1
```

DS(9,12)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 9 4   6 3 7
9 3 6   5 2 7   8 4 1
4 7 1   8 6 3   2 9 5

3 6 9   5 1 7   4 2 8
7 1 4   6 8 2   9 5 3
5 8 2   3 4 9   1 7 6
```

DS(9,13)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 9 4   6 3 7
9 3 6   5 2 7   8 4 1
4 7 1   8 6 3   2 9 5

3 8 4   7 1 5   9 2 6
5 1 9   6 8 2   4 3 7
7 6 2   3 4 9   1 5 8
```

DS(9,14)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 9 4   6 7 3
9 3 6   5 2 7   1 4 8
4 7 1   8 6 3   9 2 5

3 6 9   7 4 2   8 5 1
7 1 4   3 8 5   2 9 6
5 8 2   6 1 9   4 3 7
```

DS(9,15)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 9 4   6 7 3
9 3 6   5 2 7   1 4 8
4 7 1   8 6 3   9 2 5

3 8 4   6 1 9   2 5 7
5 1 9   7 4 2   8 3 6
7 6 2   3 8 5   4 9 1
```

DS(9,16)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   1 9 4   6 7 3
9 3 6   5 2 7   1 4 8
4 7 1   8 6 3   9 2 5

3 8 4   7 1 5   2 9 6
5 1 9   6 8 2   4 3 7
7 6 2   3 4 9   8 5 1
```

DS(9,17)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 6 9   1 4 7
9 3 6   7 1 4   8 2 5
4 7 1   5 8 2   6 9 3

3 6 9   1 4 7   2 5 8
7 1 4   8 2 5   9 3 6
5 8 2   6 9 3   4 7 1
```

DS(9,18)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 6 9   1 4 7
9 3 6   7 1 4   8 2 5
4 7 1   5 8 2   6 9 3

3 6 9   8 2 5   4 7 1
7 1 4   6 9 3   2 5 8
5 8 2   1 4 7   9 3 6
```

DS(9,19)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 6 9   1 4 7
9 3 6   7 1 4   8 2 5
4 7 1   5 8 2   6 9 3

3 8 4   1 9 5   2 7 6
5 1 9   6 2 7   4 3 8
7 6 2   8 4 3   9 5 1
```

DS(9,20)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 6 9   1 4 7
9 3 6   7 1 4   8 2 5
4 7 1   5 8 2   6 9 3

3 8 4   6 2 7   9 5 1
5 1 9   8 4 3   2 7 6
7 6 2   1 9 5   4 3 8
```

DS(9,21)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 6 9   4 7 1
9 3 6   7 1 4   2 5 8
4 7 1   5 8 2   9 3 6

3 6 9   1 4 7   8 2 5
7 1 4   8 2 5   9 3 6
5 8 2   6 9 3   1 4 7
```

DS(9,22)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 6 9   4 7 1
9 3 6   7 1 4   2 5 8
4 7 1   5 8 2   9 3 6

3 8 4   1 9 5   6 2 7
5 1 9   6 2 7   8 4 3
7 6 2   8 4 3   1 9 5
```

DS(9,23)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 6 9   4 7 1
9 3 6   7 1 4   2 5 8
4 7 1   5 8 2   9 3 6

3 8 4   6 2 7   1 9 5
5 1 9   8 4 3   6 2 7
7 6 2   1 9 5   8 4 3
```

DS(9,24)

```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 9 4   1 7 6
9 3 6   5 1 7   4 2 8
4 7 1   8 6 2   9 5 3

3 8 4   7 2 5   6 9 1
5 1 9   6 8 3   2 4 7
7 6 2   1 4 9   8 3 5
```

DS(9,25)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 9 4   6 7 1
9 3 6   5 1 7   2 4 8
4 7 1   8 6 2   9 3 5

3 8 4   6 2 9   1 5 7
5 1 9   7 4 3   8 2 6
7 6 2   1 8 5   4 9 3
```

DS(9,26)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   3 9 4   6 7 1
9 3 6   5 1 7   2 4 8
4 7 1   8 6 2   9 3 5

3 8 4   7 2 5   1 9 6
5 1 9   6 8 3   4 2 7
7 6 2   1 4 9   8 5 3
```

DS(9,27)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 8   7 1 4   6 9 3
9 3 6   5 8 2   1 4 7
4 7 1   3 6 9   8 2 5

3 8 4   7 2 5   1 9 6
5 1 9   6 2 7   4 3 8
7 6 2   8 4 3   9 5 1
```

DS(9,28)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   1 6 7   4 3 8
7 3 6   8 2 4   9 5 1
4 8 1   5 9 3   2 7 6

3 6 8   7 4 2   1 9 5
9 1 4   3 8 5   6 2 7
5 7 2   6 1 9   8 4 3
```

DS(9,29)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   1 6 7   4 3 8
7 3 6   8 2 4   9 5 1
4 8 1   5 9 3   2 7 6

3 7 4   6 8 2   1 9 5
5 1 8   3 4 9   6 2 7
9 6 2   7 1 5   8 4 3
```

DS(9,30)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   1 8 4   6 3 7
7 3 6   5 2 9   8 4 1
4 8 1   7 6 3   2 9 5

3 7 4   6 9 2   1 5 8
5 1 8   3 4 7   9 2 6
9 6 2   8 1 5   4 7 3
```

DS(9,31)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   1 8 4   6 3 7
7 3 6   5 2 9   8 4 1
4 8 1   7 6 3   2 9 5

3 7 4   8 1 5   9 2 6
5 1 8   6 9 2   4 7 3
9 6 2   3 4 7   1 5 8
```

DS(9,32)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   1 8 4   6 7 3
7 3 6   5 2 9   1 4 8
4 8 1   7 6 3   9 2 5

3 7 4   8 1 5   2 9 6
5 1 8   6 9 2   4 3 7
9 6 2   3 4 7   8 5 1
```

DS(9,33)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   3 6 7   1 4 8
7 3 6   8 1 4   9 2 5
4 8 1   5 9 2   6 7 3

3 6 8   1 4 9   2 5 7
9 1 4   7 2 5   8 3 6
5 7 2   6 8 3   4 9 1
```

DS(9,34)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   3 6 7   1 4 8
7 3 6   8 1 4   9 2 5
4 8 1   5 9 2   6 7 3

3 6 8   7 2 5   4 9 1
9 1 4   6 8 3   2 5 7
5 7 2   1 4 9   8 3 6
```

DS(9,35)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   3 6 7   1 4 8
7 3 6   8 1 4   9 2 5
4 8 1   5 9 2   6 7 3

3 7 4   8 1 5   2 9 6
5 1 8   6 2 9   4 3 7
9 6 2   7 4 3   8 5 1
```

DS(9,36)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   3 6 7   1 4 8
7 3 6   8 1 4   9 2 5
4 8 1   5 9 2   6 7 3

3 7 4   8 1 5   2 9 6
5 1 8   7 4 3   2 9 6
9 6 2   1 8 5   4 3 7
```

DS(9,37)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   3 6 7   8 4 1
7 3 6   8 1 4   2 9 5
4 8 1   5 9 2   6 3 7

3 7 4   1 8 5   9 2 6
5 1 8   6 2 9   4 7 3
9 6 2   7 4 3   1 5 8
```

DS(9,38)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   3 8 4   6 7 1
7 3 6   5 1 9   2 4 8
4 8 1   7 6 2   9 3 5

3 6 8   1 9 5   4 2 7
9 1 4   6 2 7   8 5 3
5 7 2   8 4 3   1 9 6
```

DS(9,39)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   3 8 4   6 7 1
7 3 6   5 1 9   2 4 8
4 8 1   7 6 2   9 3 5

3 7 4   1 9 5   8 2 6
5 1 8   6 2 7   4 9 3
9 6 2   8 4 3   1 5 7
```

DS(9,40)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 5 9   8 1 4   6 7 3
7 3 6   5 9 2   1 4 8
4 8 1   3 6 7   9 2 5

3 7 4   1 8 5   2 9 6
5 1 8   6 2 9   4 3 7
9 6 2   7 4 3   8 5 1
```

DS(9,41)
```
1 4 7   2 5 8   3 6 9
8 2 5   9 3 6   7 1 4
6 9 3   4 7 1   5 8 2

2 7 4   1 9 5   6 3 8
5 3 8   6 2 7   9 4 1
9 6 1   8 4 3   2 7 5

3 8 6   5 1 9   4 2 7
4 1 9   7 6 2   8 5 3
7 5 2   3 8 4   1 9 6
```

DS(9,42)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 8   1 4 7   6 3 9
9 3 6   8 2 5   7 4 1
4 7 1   6 9 3   2 8 5

3 6 9   5 8 2   1 7 4
7 1 4   3 6 9   5 2 8
5 8 2   7 1 4   9 6 3
```

DS(9,43)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 8   1 4 9   6 7 3
9 3 6   7 2 5   1 4 8
4 7 1   6 8 3   9 2 5

3 6 9   5 1 7   2 8 4
7 1 4   8 6 2   5 3 9
5 8 2   3 9 4   7 6 1
```

DS(9,44)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 8   1 4 9   6 7 3
9 3 6   7 2 5   1 4 8
4 7 1   6 8 3   9 2 5

3 8 4   5 1 7   2 6 9
5 1 9   8 6 2   7 3 4
7 6 2   3 9 4   5 8 1
```

DS(9,45)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 8   1 4 9   6 7 3
9 3 6   7 2 5   1 4 8
4 7 1   6 8 3   9 2 5

3 8 4   5 9 2   7 6 1
5 1 9   3 6 7   2 8 4
7 6 2   8 1 4   5 3 9
```

DS(9,46)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 8   1 4 9   7 6 3
9 3 6   7 2 5   1 8 4
4 7 1   6 8 3   5 2 9

3 6 9   5 1 7   2 4 8
7 1 4   8 6 2   9 3 5
5 8 2   3 9 4   6 7 1
```

DS(9,47)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 8   1 4 9   7 6 3
9 3 6   7 2 5   1 8 4
4 7 1   6 8 3   5 2 9

3 6 9   8 1 4   2 7 5
7 1 4   5 9 2   6 3 8
5 8 2   3 6 7   9 4 1
```

DS(9,48)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 8   1 4 9   7 6 3
9 3 6   7 2 5   1 8 4
4 7 1   6 8 3   5 2 9

3 8 4   5 9 2   6 7 1
5 1 9   3 6 7   2 4 8
7 6 2   8 1 4   9 3 5
```

DS(9,49)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 6 9  7 3 4
9 3 6  7 2 4  5 8 1
4 7 1  5 8 3  2 6 9
3 6 9  8 4 2  1 7 5
7 1 4  3 9 5  6 2 8
5 8 2  6 1 7  9 4 3
```

DS(9,50)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 6 9  7 3 4
9 3 6  7 2 4  5 8 1
4 7 1  5 8 3  2 6 9
3 8 4  6 1 7  9 2 5
5 1 9  8 4 2  6 7 3
7 6 2  3 9 5  1 4 8
```

DS(9,51)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 6 9  7 3 4
9 3 6  7 2 4  5 8 1
4 7 1  5 8 3  2 6 9
3 8 4  6 9 2  1 7 5
5 1 9  3 4 7  6 2 8
7 6 2  8 1 5  9 4 3
```

DS(9,52)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 6 9  7 4 3
9 3 6  7 2 4  1 8 5
4 7 1  5 8 3  6 2 9
3 6 9  8 1 5  2 7 4
7 1 4  6 9 2  5 3 8
5 8 2  3 4 7  9 6 1
```

DS(9,53)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 6 9  7 4 3
9 3 6  7 2 4  1 8 5
4 7 1  5 8 3  6 2 9
3 6 9  8 4 2  5 7 1
7 1 4  3 9 5  2 6 8
5 8 2  6 1 7  9 3 4
```

DS(9,54)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 6 9  7 4 3
9 3 6  7 2 4  1 8 5
4 7 1  5 8 3  6 2 9
3 8 4  6 9 2  5 7 1
5 1 9  3 4 7  2 6 8
7 6 2  8 1 5  9 3 4
```

DS(9,55)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 9 4  6 7 3
9 3 6  5 2 7  1 4 8
4 7 1  8 6 3  9 2 5
3 6 9  7 1 5  2 8 4
7 1 4  6 8 2  5 3 9
5 8 2  3 4 9  7 6 1
```

DS(9,56)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 9 4  6 7 3
9 3 6  5 2 7  1 4 8
4 7 1  8 6 3  9 2 5
3 6 9  7 4 2  5 8 1
7 1 4  3 8 5  2 6 9
5 8 2  6 1 9  7 3 4
```

DS(9,57)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 9 4  6 7 3
9 3 6  5 2 7  1 4 8
4 7 1  8 6 3  9 2 5
3 8 4  7 1 5  2 6 9
5 1 9  6 8 2  7 3 4
7 6 2  3 4 9  5 8 1
```

DS(9,58)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 9 4  7 6 3
9 3 6  5 2 7  1 8 4
4 7 1  8 6 3  5 2 9
3 6 9  7 1 5  2 4 8
7 1 4  6 8 2  9 3 5
5 8 2  3 4 9  6 7 1
```

DS(9,59)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  1 9 4  7 6 3
9 3 6  5 2 7  1 8 4
4 7 1  8 6 3  5 2 9
3 8 4  6 1 9  2 7 5
5 1 9  7 4 2  6 3 8
7 6 2  3 8 5  9 4 1
```

DS(9,60)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  3 6 9  1 7 4
9 3 6  7 1 4  5 2 8
4 7 1  5 8 2  9 6 3
3 6 9  1 8 5  7 2 4
7 1 4  8 2 5  6 3 9
5 8 2  6 9 3  7 4 1
```

DS(9,61)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  3 6 9  1 7 4
9 3 6  7 1 4  5 2 8
4 7 1  5 8 2  9 6 3
3 6 9  8 2 5  7 4 1
7 1 4  6 9 3  2 8 5
5 8 2  1 4 7  6 3 9
```

DS(9,62)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  3 6 9  7 4 1
9 3 6  7 1 4  2 8 5
4 7 1  5 8 2  6 3 9
3 6 9  1 4 7  5 2 8
7 1 4  8 2 5  9 6 3
5 8 2  6 9 3  1 7 4
```

DS(9,63)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  3 6 9  7 4 1
9 3 6  7 1 4  2 8 5
4 7 1  5 8 2  6 3 9
3 6 9  7 2 5  1 8 4
7 1 4  6 9 3  5 2 8
5 8 2  1 4 7  9 6 3
```

DS(9,64)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  3 9 4  6 7 1
9 3 6  5 1 7  2 4 8
4 7 1  8 6 2  9 3 5
3 6 9  1 8 5  7 2 4
7 1 4  6 2 9  5 8 3
5 8 2  7 4 3  1 6 9
```

DS(9,65)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  3 9 4  6 7 1
9 3 6  5 1 7  2 4 8
4 7 1  8 6 2  9 3 5
3 6 9  7 2 5  1 8 4
7 1 4  6 8 3  5 2 9
5 8 2  1 4 9  7 6 3
```

DS(9,66)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  3 9 4  6 7 1
9 3 6  5 1 7  2 8 4
4 7 1  8 6 2  9 3 5
3 8 4  7 2 5  1 6 9
5 1 9  6 8 3  7 2 4
7 6 2  1 4 9  5 8 3
```

DS(9,67)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  3 9 4  7 6 1
9 3 6  5 1 7  2 8 4
4 7 1  8 6 2  5 3 9
3 6 9  7 2 5  1 4 8
7 1 4  6 8 3  9 2 5
5 8 2  1 4 9  6 7 3
```

DS(9,68)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  3 9 4  7 6 1
9 3 6  5 1 7  2 8 4
4 7 1  8 6 2  5 3 9
3 8 4  6 2 9  1 7 5
5 1 9  7 4 3  6 2 8
7 6 2  1 8 5  9 4 3
```

DS(9,69)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 8  6 9 3  1 7 4
9 3 6  1 4 7  5 2 8
4 7 1  8 2 5  9 6 3
3 6 9  7 1 4  2 8 5
7 1 4  5 8 2  6 3 9
5 8 2  3 6 9  7 4 1
```

DS(9,70)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 9  1 8 4  6 7 3
7 3 6  5 2 9  1 4 8
4 8 1  7 6 3  9 2 5
3 7 4  6 9 2  5 8 1
5 1 8  3 4 7  2 6 9
9 6 2  8 1 5  7 3 4
```

DS(9,71)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 9  1 8 4  6 7 3
7 3 6  5 2 9  1 4 8
4 8 1  7 6 3  9 2 5
3 7 4  8 1 5  2 6 9
5 1 8  6 9 2  7 3 4
9 6 2  3 4 7  5 8 1
```

DS(9,72)
```
1 4 7  2 5 8  3 9 6
8 2 5  9 3 6  4 1 7
6 9 3  4 7 1  8 5 2
2 5 9  3 4 7  1 6 8
7 3 6  8 1 5  9 2 4
4 8 1  6 9 2  5 7 3
3 6 8  5 2 9  7 4 1
9 1 4  7 6 3  2 8 5
5 7 2  1 8 4  6 3 9
```

DS(9,73)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   3 4 7   1 6 8
7 3 6   8 1 5   9 2 4
4 8 1   6 9 2   5 7 3

3 7 4   1 6 9   2 8 5
5 1 8   7 2 4   6 3 9
9 6 2   5 8 3   7 4 1
```

DS(9,74)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   3 4 7   1 6 8
7 3 6   8 1 5   9 2 4
4 8 1   6 9 2   5 7 3

3 7 4   5 2 9   6 8 1
5 1 8   7 6 3   2 4 9
9 6 2   1 8 4   7 3 5
```

DS(9,75)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   3 6 7   1 8 4
7 3 6   8 1 4   5 2 9
4 8 1   5 9 2   7 6 3

3 6 8   1 4 9   2 7 5
9 1 4   7 2 5   6 3 8
5 7 2   6 8 3   9 4 1
```

DS(9,76)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   3 6 7   1 8 4
7 3 6   8 1 4   5 2 9
4 8 1   5 9 2   7 6 3

3 6 8   7 2 5   9 4 1
9 1 4   6 8 3   2 7 5
5 7 2   1 4 9   6 3 8
```

DS(9,77)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   3 8 4   7 6 1
7 3 6   5 1 9   2 4 8
4 8 1   7 6 2   9 3 5

3 6 8   1 9 5   7 2 4
9 1 4   6 2 7   5 8 3
5 7 2   8 4 3   1 6 9
```

DS(9,78)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   3 8 4   7 6 1
7 3 6   5 1 9   2 8 4
4 8 1   7 6 2   5 3 9

3 7 4   1 9 5   6 2 8
5 1 8   6 2 7   9 4 3
9 6 2   8 4 3   1 7 5
```

DS(9,79)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   6 8 3   1 7 4
7 3 6   1 4 9   5 2 8
4 8 1   7 2 5   9 6 3

3 6 8   5 1 7   2 4 9
9 1 4   8 6 2   7 3 5
5 7 2   3 9 4   6 8 1
```

DS(9,80)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   6 8 3   1 7 4
7 3 6   1 4 9   5 2 8
4 8 1   7 2 5   9 6 3

3 6 8   5 9 2   7 4 1
9 1 4   3 6 7   2 8 5
5 7 2   8 1 4   6 3 9
```

DS(9,81)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   6 8 3   1 7 4
7 3 6   1 4 9   5 2 8
4 8 1   7 2 5   9 6 3

3 7 4   5 9 2   6 8 1
5 1 8   3 6 7   2 4 9
9 6 2   8 1 4   7 3 5
```

DS(9,82)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 1 4   6 3 8
7 3 6   5 8 2   9 4 1
4 8 1   3 6 9   2 7 5

3 6 8   1 9 5   7 2 4
9 1 4   6 2 7   5 8 3
5 7 2   8 4 3   1 6 9
```

DS(9,83)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 1 4   6 8 3
7 3 6   5 8 2   1 4 9
4 8 1   3 6 9   7 2 5

3 6 8   1 9 5   2 7 4
9 1 4   6 2 7   5 3 8
5 7 2   8 4 3   9 6 1
```

DS(9,84)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 1 4   6 8 3
7 3 6   5 8 2   1 4 9
4 8 1   3 6 9   7 2 5

3 7 4   1 9 5   2 6 8
5 1 8   6 2 7   9 3 4
9 6 2   8 4 3   5 7 1
```

DS(9,85)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 4 3   1 6 8
7 3 6   1 8 5   9 2 4
4 8 1   6 2 9   5 7 3

3 6 8   5 1 7   2 4 9
9 1 4   8 6 2   7 3 5
5 7 2   3 9 4   6 8 1
```

DS(9,86)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 4 3   1 6 8
7 3 6   1 8 5   9 2 4
4 8 1   6 2 9   5 7 3

3 6 8   5 9 2   7 4 1
9 1 4   3 6 7   2 8 5
5 7 2   8 1 4   6 3 9
```

DS(9,87)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 4 3   1 6 8
7 3 6   1 8 5   9 2 4
4 8 1   6 2 9   5 7 3

3 7 4   5 9 2   6 8 1
5 1 8   3 6 7   2 4 9
9 6 2   8 1 4   7 3 5
```

DS(9,88)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 4 3   6 8 1
7 3 6   1 8 5   9 2 4
4 8 1   6 2 9   7 3 5

3 6 8   5 9 2   7 4 1
9 1 4   8 6 2   5 7 3
5 7 2   3 9 4   1 6 8
```

DS(9,89)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 4 3   6 8 1
7 3 6   1 8 5   2 4 9
4 8 1   6 2 9   7 3 5

3 6 8   5 9 2   1 7 4
9 1 4   3 6 7   5 2 8
5 7 2   8 1 4   9 6 3
```

DS(9,90)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 4 3   6 8 1
7 3 6   1 8 5   2 4 9
4 8 1   6 2 9   7 3 5

3 7 4   5 9 2   1 6 8
5 1 8   3 6 7   9 2 4
9 6 2   8 1 4   5 7 3
```

DS(9,91)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   7 6 3   1 4 8
7 3 6   1 8 5   9 2 4
4 8 1   5 2 9   6 7 3

3 7 4   8 1 5   2 6 9
5 1 8   6 9 2   7 3 4
9 6 2   3 4 7   5 8 1
```

DS(9,92)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   8 4 3   6 7 1
7 3 6   1 9 5   2 8 4
4 8 1   6 2 7   9 3 5

3 6 8   5 1 9   7 2 4
9 1 4   7 6 2   5 8 3
5 7 2   3 8 4   1 6 9
```

DS(9,93)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   8 4 3   6 7 1
7 3 6   1 9 5   2 4 8
4 8 1   6 2 7   9 3 5

3 6 8   7 1 4   5 2 9
9 1 4   5 8 2   7 6 3
5 7 2   3 6 9   1 8 4
```

DS(9,94)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   8 4 3   6 7 1
7 3 6   1 9 5   2 8 4
4 8 1   6 2 7   9 3 5

3 7 4   5 8 2   1 6 9
5 1 8   3 6 9   7 2 4
9 6 2   7 1 4   5 8 3
```

DS(9,95)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   8 4 3   7 6 1
7 3 6   1 9 5   2 8 4
4 8 1   6 2 7   5 3 9

3 6 8   7 1 4   9 2 5
9 1 4   5 8 2   6 7 3
5 7 2   3 6 9   1 4 8
```

DS(9,96)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   8 4 3   7 6 1
7 3 6   1 9 5   2 8 4
4 8 1   6 2 7   5 3 9

3 7 4   5 1 9   6 2 8
5 1 8   7 6 2   9 4 3
9 6 2   3 8 4   1 7 5
```

DS(9,97)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   8 6 3   1 7 4
7 3 6   1 9 4   5 2 8
4 8 1   5 2 7   9 6 3

3 6 8   7 1 5   2 4 9
9 1 4   6 8 2   7 3 5
5 7 2   3 4 9   6 8 1
```

DS(9,98)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   8 6 3   1 7 4
7 3 6   1 9 4   5 2 8
4 8 1   5 2 7   9 6 3

3 7 4   6 1 9   2 8 5
5 1 8   7 4 2   6 3 9
9 6 2   3 8 5   7 4 1
```

DS(9,99)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   8 6 3   7 4 1
7 3 6   1 9 4   2 8 5
4 8 1   5 2 7   6 3 9

3 6 8   7 1 5   9 2 4
9 1 4   6 8 2   5 7 3
5 7 2   3 4 9   1 6 8
```

DS(9,100)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 5 9   8 6 3   7 4 1
7 3 6   1 9 4   2 8 5
4 8 1   5 2 7   6 3 9

3 7 4   6 1 9   5 2 8
5 1 8   7 4 2   9 6 3
9 6 2   3 8 5   1 7 4
```

DS(9,101)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 6 9   3 8 4   1 7 5
7 3 4   5 1 9   6 2 8
5 8 1   7 6 2   9 4 3

3 7 6   1 9 5   2 8 4
4 1 8   6 2 7   5 3 9
9 5 2   8 4 3   7 6 1
```

DS(9,102)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 6 9   3 8 5   1 7 4
7 3 4   6 1 9   5 2 8
5 8 1   7 4 2   9 6 3

3 7 6   1 9 4   2 8 5
4 1 8   5 2 7   6 3 9
9 5 2   8 6 3   7 4 1
```

DS(9,103)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 6 9   3 8 5   7 4 1
7 3 4   6 1 9   2 8 5
5 8 1   7 4 2   6 3 9

3 7 6   1 9 5   2 8 4
4 1 8   5 2 7   9 6 3
9 5 2   8 6 3   1 7 4
```

DS(9,104)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 6 9   7 1 4   5 8 3
7 3 4   5 8 2   1 6 9
5 8 1   3 6 9   7 2 4

3 7 6   1 9 5   2 4 8
4 1 8   6 2 7   9 3 5
9 5 2   8 4 3   6 7 1
```

DS(9,105)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 6 9   7 4 3   5 8 1
7 3 4   1 8 5   2 6 9
5 8 1   6 2 9   7 3 4

3 7 6   8 1 4   9 2 5
4 1 8   5 9 2   6 7 3
9 5 2   3 6 7   1 4 8
```

DS(9,106)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 6 9   8 1 4   5 7 3
7 3 4   5 9 2   1 6 8
5 8 1   3 6 7   9 2 4

3 7 6   1 8 5   2 4 9
4 1 8   6 2 9   7 3 5
9 5 2   7 4 3   6 8 1
```

DS(9,107)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 6 9   8 4 3   1 7 5
7 3 4   1 9 5   6 2 8
5 8 1   6 2 7   9 4 3

3 7 6   5 1 9   2 8 4
4 1 8   7 6 2   5 3 9
9 5 2   3 8 4   7 6 1
```

DS(9,108)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 6 9   8 4 3   5 7 1
7 3 4   1 9 5   2 6 8
5 8 1   6 2 7   9 3 4

3 7 6   5 8 2   1 4 9
4 1 8   3 6 9   7 2 5
9 5 2   7 1 4   6 8 3
```

DS(9,109)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 7 4   1 8 5   6 3 9
5 3 8   6 2 9   7 4 1
9 6 1   7 4 3   2 8 5

3 8 6   5 1 7   9 2 4
4 1 9   8 6 2   5 7 3
7 5 2   3 9 4   1 6 8
```

DS(9,110)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 7 4   1 9 5   6 3 8
5 3 8   6 2 7   9 4 1
9 6 1   8 4 3   2 7 5

3 8 6   5 1 9   7 2 4
4 1 9   7 6 2   5 8 3
7 5 2   3 8 4   1 6 9
```

DS(9,111)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 7 4   1 9 5   6 3 8
5 3 8   6 2 7   9 4 1
9 6 1   8 4 3   2 7 5

3 8 6   7 1 4   5 2 9
4 1 9   5 8 2   7 6 3
7 5 2   3 6 9   1 8 4
```

DS(9,112)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 7 4   1 9 5   6 8 3
5 3 8   6 2 7   1 4 9
9 6 1   8 4 3   7 2 5

3 8 6   5 1 9   2 7 4
4 1 9   7 6 2   5 3 8
7 5 2   3 8 4   9 6 1
```

DS(9,113)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 7 4   3 6 9   1 8 5
5 3 8   7 1 4   6 2 9
9 6 1   5 8 2   7 4 3

3 8 6   1 9 5   2 7 4
4 1 9   6 2 7   5 3 8
7 5 2   8 4 3   9 6 1
```

DS(9,114)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 7 4   3 9 5   1 6 8
5 3 8   7 6 2   9 4 1
9 6 1   8 4 2   5 7 3

3 8 6   5 2 9   7 4 1
4 1 9   7 6 3   2 8 5
7 5 2   1 8 4   6 3 9
```

DS(9,115)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 7 4   5 1 9   6 3 8
5 3 8   7 6 2   9 4 1
9 6 1   3 8 4   2 7 5

3 8 6   1 9 5   7 2 4
4 1 9   6 2 7   5 8 3
7 5 2   8 4 3   1 6 9
```

DS(9,116)
```
1 4 7   2 5 8   3 9 6
8 2 5   9 3 6   4 1 7
6 9 3   4 7 1   8 5 2

2 7 4   5 1 9   6 8 3
5 3 8   7 6 2   1 4 9
9 6 1   3 8 4   7 2 5

3 8 6   1 9 5   2 7 4
4 1 9   6 2 7   5 3 8
7 5 2   8 4 3   9 6 1
```

DS(9,117)
```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 8   3 6 7   1 4 9
9 3 6   8 1 4   7 2 5
4 7 1   5 9 2   6 8 3

3 6 9   1 4 8   2 5 7
7 1 4   9 2 5   8 3 6
5 8 2   6 7 3   4 9 1
```

DS(9,118)
```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 8   3 6 7   1 4 9
9 3 6   8 1 4   7 2 5
4 7 1   5 9 2   6 8 3

3 8 4   1 7 5   2 9 6
5 1 9   6 2 8   4 3 7
7 6 2   9 4 3   8 5 1
```

DS(9,119)
```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 8   3 6 7   4 9 1
9 3 6   8 1 4   2 5 7
4 7 1   5 9 2   8 3 6

3 6 9   1 4 8   7 2 5
7 1 4   9 2 5   6 8 3
5 8 2   6 7 3   1 4 9
```

DS(9,120)
```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 8   3 6 7   4 9 1
9 3 6   8 1 4   2 5 7
4 7 1   5 9 2   8 3 6

3 8 4   1 7 5   6 2 9
5 1 9   6 2 8   7 4 3
7 6 2   9 4 3   1 8 5
```

```
DS(9,121)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   1 6 7   8 4 3
7 3 6   8 2 4   1 5 9
4 8 1   5 9 3   7 2 6

3 6 8   9 1 5   2 4 7
9 1 4   6 7 2   8 3 5
5 7 2   3 4 8   6 9 1
```

```
DS(9,122)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   1 6 7   4 8 3
7 3 6   8 2 4   1 5 9
4 8 1   5 9 3   7 2 6

3 7 4   6 1 8   2 9 5
5 1 8   9 4 2   6 3 7
9 6 2   3 7 5   8 4 1
```

```
DS(9,123)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   1 6 7   8 4 3
7 3 6   8 2 4   1 9 5
4 8 1   5 9 3   6 2 7

3 6 8   9 4 2   7 5 1
9 1 4   3 7 5   2 8 6
5 7 2   6 1 8   4 3 9
```

```
DS(9,124)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   1 6 7   8 4 3
7 3 6   8 2 4   1 9 5
4 8 1   5 9 3   6 2 7

3 7 4   6 1 8   2 5 9
5 1 8   9 4 2   7 3 6
9 6 2   3 7 5   4 8 1
```

```
DS(9,125)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   1 6 7   8 4 3
7 3 6   8 2 4   1 9 5
4 8 1   5 9 3   6 2 7

3 7 4   9 1 5   2 8 6
5 1 8   6 7 2   4 3 9
9 6 2   3 4 8   7 5 1
```

```
DS(9,126)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 4 7   1 8 6
7 3 6   8 1 5   4 2 9
4 8 1   6 9 2   7 5 3

3 6 8   1 7 4   2 9 5
9 1 4   5 2 8   6 3 7
5 7 2   9 6 3   8 4 1
```

```
DS(9,127)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 4 7   1 8 6
7 3 6   8 1 5   4 2 9
4 8 1   6 9 2   7 5 3

3 7 4   1 6 8   2 9 5
5 1 8   9 2 4   6 3 7
9 6 2   5 7 3   8 4 1
```

```
DS(9,128)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 4 7   1 8 6
7 3 6   8 1 5   4 2 9
4 8 1   6 9 2   7 5 3

3 7 4   5 2 8   6 9 1
5 1 8   9 6 3   2 4 7
9 6 2   1 7 4   8 3 5
```

```
DS(9,129)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 4 7   6 8 1
7 3 6   8 1 5   2 4 9
4 8 1   6 9 2   7 3 5

3 6 8   9 2 4   1 5 7
9 1 4   5 7 3   8 2 6
5 7 2   1 6 8   4 9 3
```

```
DS(9,130)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 4 7   6 8 1
7 3 6   8 1 5   2 4 9
4 8 1   6 9 2   7 3 5

3 7 4   5 2 8   1 9 6
5 1 8   9 6 3   4 2 7
9 6 2   1 7 4   8 5 3
```

```
DS(9,131)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 6 8   1 4 7
7 3 6   9 1 4   8 2 5
4 8 1   5 7 2   6 9 3

3 6 8   1 4 7   2 5 9
9 1 4   8 2 5   7 3 6
5 7 2   6 9 3   4 8 1
```

```
DS(9,132)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 6 8   1 4 7
7 3 6   9 1 4   8 2 5
4 8 1   5 7 2   6 9 3

3 7 4   5 2 8   6 9 1
5 1 8   6 2 7   4 3 9
9 6 2   8 4 3   7 5 1
```

```
DS(9,133)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 6 8   7 4 1
7 3 6   9 1 4   2 8 5
4 8 1   5 7 2   6 3 9

3 6 8   1 9 5   4 2 7
9 1 4   6 2 7   8 5 3
5 7 2   8 4 3   1 9 6
```

```
DS(9,134)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 6 8   7 4 1
7 3 6   9 1 4   2 8 5
4 8 1   5 7 2   6 3 9

3 7 4   1 9 5   8 2 6
5 1 8   6 2 7   4 9 3
9 6 2   8 4 3   1 5 7
```

```
DS(9,135)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 6 8   7 4 1
7 3 6   9 1 4   2 8 5
4 8 1   5 7 2   6 3 9

3 7 4   1 6 8   2 5 9
5 1 8   6 9 3   4 2 7
9 6 2   1 4 7   8 5 3
```

```
DS(9,136)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 7 4   1 8 6
7 3 6   5 1 8   4 2 9
4 8 1   9 6 2   7 5 3

3 7 4   5 2 8   6 9 1
5 1 8   6 9 3   2 4 7
9 6 2   1 4 7   8 3 5
```

```
DS(9,137)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 7 4   6 8 1
7 3 6   5 1 8   2 4 9
4 8 1   9 6 2   7 3 5

3 6 8   1 9 5   4 2 7
9 1 4   6 2 7   8 5 3
5 7 2   8 4 3   1 9 6
```

```
DS(9,138)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   3 7 4   6 8 1
7 3 6   5 1 8   2 4 9
4 8 1   9 6 2   7 3 5

3 7 4   1 9 5   8 2 6
5 1 8   6 2 7   4 9 3
9 6 2   8 4 3   1 5 7
```

```
DS(9,139)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   6 1 7   4 8 3
7 3 6   8 4 2   1 5 9
4 8 1   3 9 5   7 2 6

3 6 8   1 7 4   2 9 5
9 1 4   5 2 8   6 3 7
5 7 2   9 6 3   8 4 1
```

```
DS(9,140)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   6 1 7   4 8 3
7 3 6   8 4 2   1 5 9
4 8 1   3 9 5   7 2 6

3 7 4   1 6 8   2 9 5
5 1 8   9 2 4   6 3 7
9 6 2   5 7 3   8 4 1
```

```
DS(9,141)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   6 1 7   4 8 3
7 3 6   8 4 2   1 5 9
4 8 1   3 9 5   7 2 6

3 7 4   5 2 8   6 9 1
5 1 8   9 6 3   2 4 7
9 6 2   1 7 4   8 3 5
```

```
DS(9,142)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 5 9   8 6 3   7 4 1
7 3 6   1 9 4   2 8 5
4 8 1   5 2 7   6 3 9

3 7 4   9 1 5   8 2 6
5 1 8   6 7 2   4 9 3
9 6 2   3 4 8   1 5 7
```

```
DS(9,143)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   1 9 4   7 3 5
9 3 4   5 2 7   6 8 1
5 7 1   8 6 3   2 4 9

3 5 9   6 1 8   4 2 7
7 1 6   9 4 2   8 5 3
4 8 2   3 7 5   1 9 6
```

```
DS(9,144)
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   1 9 4   7 3 5
9 3 4   5 2 7   6 8 1
5 7 1   8 6 3   2 4 9

3 8 6   9 1 5   4 2 7
4 1 9   6 7 2   8 5 3
7 5 2   3 4 8   1 9 6
```

DS(9,145)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   1 9 4   7 3 5
9 3 4   5 2 7   6 8 1
5 7 1   8 6 3   2 4 9

3 8 6   9 4 2   1 5 7
4 1 9   3 7 5   8 2 6
7 5 2   6 1 8   4 9 3
```

DS(9,146)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   1 9 4   7 5 3
9 3 4   5 2 7   1 8 6
5 7 1   8 6 3   4 2 9

3 5 9   6 1 8   2 4 7
7 1 6   9 4 2   8 3 5
4 8 2   3 7 5   6 9 1
```

DS(9,147)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   1 9 4   7 5 3
9 3 4   5 2 7   1 8 6
5 7 1   8 6 3   4 2 9

3 5 9   6 7 2   8 4 1
7 1 6   3 4 8   2 9 5
4 8 2   9 1 5   6 3 7
```

DS(9,148)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   3 4 7   1 5 9
9 3 4   8 1 5   7 2 6
5 7 1   6 9 2   4 8 3

3 5 9   1 6 8   2 4 7
7 1 6   9 2 4   8 3 5
4 8 2   5 7 3   6 9 1
```

DS(9,149)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   3 4 7   1 5 9
9 3 4   8 1 5   7 2 6
5 7 1   6 9 2   4 8 3

3 8 6   1 7 4   2 9 5
4 1 9   5 2 8   6 3 7
7 5 2   9 6 3   8 4 1
```

DS(9,150)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   3 4 7   1 5 9
9 3 4   8 1 5   6 2 7
5 7 1   6 9 2   8 4 3

3 5 9   1 7 4   2 8 6
7 1 6   5 2 8   4 3 9
4 8 2   9 6 3   7 5 1
```

DS(9,151)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   3 4 7   1 9 5
9 3 4   8 1 5   6 2 7
5 7 1   6 9 2   8 4 3

3 8 6   9 2 4   7 5 1
4 1 9   5 7 3   2 8 6
7 5 2   1 6 8   4 3 9
```

DS(9,152)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   3 4 7   1 5 9
9 3 4   6 1 7   2 8 5
5 7 1   8 4 2   6 3 9

3 5 9   1 6 8   4 2 7
7 1 6   9 2 4   8 5 3
4 8 2   5 7 3   1 9 6
```

DS(9,153)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   3 9 5   7 4 1
9 3 4   6 1 7   2 8 5
5 7 1   8 4 2   6 3 9

3 8 6   9 2 4   1 5 7
4 1 9   5 7 3   8 2 6
7 5 2   1 6 8   4 9 3
```

DS(9,154)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 6 8   5 9 3   7 4 1
9 3 4   1 6 7   2 8 5
5 7 1   8 2 4   6 3 9

3 5 9   6 1 8   4 2 7
7 1 6   9 4 2   8 5 3
4 8 2   3 7 5   1 9 6
```

DS(9,155)

```
1 4 7   2 5 9   3 6 8
8 2 5   7 3 6   9 1 4
6 9 3   4 8 1   5 7 2

2 7 4   3 9 5   6 8 1
5 3 8   6 1 7   2 4 9
9 6 1   8 4 2   7 3 5

3 5 9   1 6 8   4 2 7
7 1 6   9 2 4   8 5 3
4 8 2   5 7 3   1 9 6
```

DS(9,156)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   3 6 7   1 9 4
9 3 6   8 1 4   5 2 7
4 7 1   5 9 2   8 6 3

3 6 9   1 4 8   2 7 5
7 1 4   9 2 5   6 3 8
5 8 2   6 7 3   9 4 1
```

DS(9,157)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   3 6 7   1 9 4
9 3 6   8 1 4   5 2 7
4 7 1   5 9 2   8 6 3

3 6 9   1 7 5   2 4 8
7 1 4   6 2 8   9 3 5
5 8 2   9 4 3   6 7 1
```

DS(9,158)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   3 6 7   9 4 1
9 3 6   8 1 4   2 7 5
4 7 1   5 9 2   6 3 8

3 6 9   1 4 8   5 2 7
7 1 4   9 2 5   8 6 3
5 8 2   6 7 3   1 9 4
```

DS(9,159)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   3 6 7   9 4 1
9 3 6   8 1 4   2 7 5
4 7 1   5 9 2   6 3 8

3 6 9   1 7 5   8 2 4
7 1 4   6 2 8   5 9 3
5 8 2   9 4 3   1 6 7
```

DS(9,160)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   3 9 4   1 6 7
9 3 6   5 1 7   8 2 4
4 7 1   8 6 2   5 9 3

3 8 4   9 2 5   6 7 1
5 1 9   6 7 3   2 4 8
7 6 2   1 4 8   9 3 5
```

DS(9,161)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   3 9 4   6 7 1
9 3 6   5 1 7   2 4 8
4 7 1   8 6 2   9 3 5

3 6 9   1 7 5   8 2 4
7 1 4   6 2 8   5 9 3
5 8 2   9 4 3   1 6 7
```

DS(9,162)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   3 9 4   6 7 1
9 3 6   5 1 7   2 4 8
4 7 1   8 6 2   9 3 5

3 8 4   9 2 5   1 6 7
5 1 9   6 7 3   8 2 4
7 6 2   1 4 8   5 9 3
```

DS(9,163)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   6 7 3   1 9 4
9 3 6   1 4 8   5 2 7
4 7 1   9 2 5   8 6 3

3 6 9   8 1 4   2 7 5
7 1 4   5 9 2   6 3 8
5 8 2   3 6 7   9 4 1
```

DS(9,164)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   6 7 3   1 9 4
9 3 6   1 4 8   5 2 7
4 7 1   9 2 5   8 6 3

3 8 4   5 9 2   6 7 1
5 1 9   3 6 7   2 4 8
7 6 2   8 1 4   9 3 5
```

DS(9,165)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 8   9 4 3   6 7 1
9 3 6   1 7 5   2 4 8
4 7 1   6 2 8   9 3 5

3 8 4   5 9 2   1 6 7
5 1 9   3 6 7   8 2 4
7 6 2   8 1 4   5 9 3
```

DS(9,166)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   1 6 7   8 4 3
7 3 6   8 2 4   1 9 5
4 8 1   5 9 3   6 2 7

3 6 8   9 1 5   2 7 4
9 1 4   6 7 2   5 3 8
5 7 2   3 4 8   9 6 1
```

DS(9,167)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   1 6 7   8 4 3
7 3 6   8 2 4   1 9 5
4 8 1   5 9 3   6 2 7

3 6 8   9 4 2   5 7 1
9 1 4   3 7 5   2 6 8
5 7 2   6 1 8   9 3 4
```

DS(9,168)

```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   3 4 7   8 6 1
7 3 6   8 1 5   2 9 4
4 8 1   6 9 2   5 3 7

3 6 8   1 7 4   9 2 5
9 1 4   5 2 8   6 7 3
5 7 2   9 6 3   1 4 8
```

DS(9,169)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   3 4 7   8 6 1
7 3 6   8 1 5   2 9 4
4 8 1   6 9 2   5 3 7

3 6 8   9 2 4   1 7 5
9 1 4   5 7 3   6 2 8
5 7 2   1 6 8   9 4 3
```

DS(9,170)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   3 6 8   1 7 4
7 3 6   9 1 4   5 2 8
4 8 1   5 7 2   9 6 3

3 6 8   1 4 7   2 9 5
9 1 4   8 2 5   6 3 7
5 7 2   6 9 3   8 4 1
```

DS(9,171)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   3 6 8   1 7 4
7 3 6   9 1 4   5 2 8
4 8 1   5 7 2   9 6 3

3 6 8   1 9 5   2 4 7
9 1 4   6 2 7   8 3 5
5 7 2   8 4 3   6 9 1
```

DS(9,172)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   3 7 4   1 6 8
7 3 6   5 1 8   9 2 4
4 8 1   9 6 2   5 7 3

3 6 8   1 9 5   2 4 7
9 1 4   6 2 7   8 3 5
5 7 2   8 4 3   6 9 1
```

DS(9,173)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   3 7 4   1 6 8
7 3 6   5 1 8   9 2 4
4 8 1   9 6 2   5 7 3

3 7 4   8 2 5   6 9 1
5 1 8   6 9 3   2 4 7
9 6 2   1 4 7   8 3 5
```

DS(9,174)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   6 1 7   8 3 4
7 3 6   8 4 2   5 9 1
4 8 1   3 9 5   2 6 7

3 6 8   9 2 4   1 7 5
9 1 4   5 7 3   6 2 8
5 7 2   1 6 8   9 4 3
```

DS(9,175)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   6 1 7   8 3 4
7 3 6   8 4 2   5 9 1
4 8 1   3 9 5   2 6 7

3 7 4   1 6 8   9 2 5
5 1 8   9 2 4   6 7 3
9 6 2   5 7 3   1 4 8
```

DS(9,176)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   8 6 3   1 4 7
7 3 6   1 9 4   8 2 5
4 8 1   5 2 7   6 9 3

3 6 8   9 1 5   2 7 4
9 1 4   6 7 2   5 3 8
5 7 2   3 4 8   9 6 1
```

DS(9,177)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 5 9   8 6 3   1 4 7
7 3 6   1 9 4   8 2 5
4 8 1   5 2 7   6 9 3

3 7 4   9 1 5   2 6 8
5 1 8   6 7 2   9 3 4
9 6 2   3 4 8   5 7 1
```

DS(9,178)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 6 8   1 9 4   5 7 3
9 3 4   5 2 7   1 6 8
5 7 1   8 6 3   9 2 4

3 5 9   6 1 8   2 4 7
7 1 6   9 4 2   8 3 5
4 8 2   3 7 5   6 9 1
```

DS(9,179)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 6 8   1 9 4   5 7 3
9 3 4   5 2 7   1 6 8
5 7 1   8 6 3   9 2 4

3 8 6   9 1 5   2 4 7
4 1 9   6 7 2   8 3 5
7 5 2   3 4 8   6 9 1
```

DS(9,180)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 6 8   3 4 7   1 9 5
9 3 4   8 1 5   6 2 7
5 7 1   6 9 2   8 4 3

3 5 9   1 8 6   2 7 4
7 1 6   9 2 4   5 3 8
4 8 2   5 7 3   9 6 1
```

DS(9,181)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 6 8   3 4 7   1 9 5
9 3 4   8 1 5   6 2 7
5 7 1   6 9 2   8 4 3

3 8 6   9 2 4   5 7 1
4 1 9   5 7 3   2 6 8
7 5 2   1 6 8   9 3 4
```

DS(9,182)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 6 8   3 7 4   5 9 1
9 3 4   5 1 8   2 6 7
5 7 1   9 6 2   8 3 4

3 5 9   6 2 7   1 4 8
7 1 6   8 4 3   9 2 5
4 8 2   1 9 5   6 7 3
```

DS(9,183)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 6 9   1 7 4   8 3 5
7 3 4   5 2 8   6 9 1
5 8 1   9 6 3   2 4 7

3 7 6   8 1 5   9 2 4
4 1 8   6 9 2   5 7 3
9 5 2   3 4 7   1 6 8
```

DS(9,184)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 6 9   3 7 5   1 4 8
7 3 4   6 1 8   9 2 5
5 8 1   9 4 2   6 7 3

3 7 6   8 2 4   5 9 1
4 1 8   5 9 3   2 6 7
9 5 2   1 6 7   8 3 4
```

DS(9,185)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 6 9   8 4 3   5 7 1
7 3 4   1 9 5   2 6 8
5 8 1   6 2 7   9 3 4

3 7 6   9 1 4   8 2 5
4 1 8   5 7 2   6 9 3
9 5 2   3 6 8   1 4 7
```

DS(9,186)
```
1 4 7   2 5 9   3 8 6
8 2 5   7 3 6   4 1 9
6 9 3   4 8 1   7 5 2

2 7 6   8 1 4   5 9 3
4 3 8   5 9 2   1 6 7
9 5 1   3 6 7   8 2 4

3 6 9   1 7 5   2 4 8
7 1 4   6 2 8   9 3 5
5 8 2   9 4 3   6 7 1
```