Abstract. Let $n$ be a positive integer and $X = \{x_{ij}\}_{1 \leq i, j \leq n}$ be an $n \times n$ sized matrix of independent random variables having joint uniform distribution

\[ \Pr(x_{ij} = k \text{ for } 1 \leq k \leq n) = \frac{1}{n} \quad (1 \leq i, j \leq n). \]

A realization $M = \{m_{ij}\}$ of $X$ is called good, if its each row and each column contains a permutation of the numbers $1, 2, \ldots, n$. We present and analyse four typical algorithms which decide whether a given realization is good.

1 Introduction

Some subsets of the elements of Latin squares $[1, 13, 23, 29, 32, 53, 54, 59, 60]$, of Sudoku squares $[6, 7, 15, 16, 20, 21, 22, 28, 31, 45, 50, 55, 57, 60, 62, 65, 66, 69, 71]$, of de Bruijn arrays $[2, 3, 4, 5, 10, 11, 18, 26, 27, 35, 38, 39, 42, 44, 48, 52, 56, 61, 64, 68, 70, 72]$ and gerechte designs, connected with agricultural and industrial experiments $[7, 8, 34]$ have to contain different elements. The one dimensional special case is also studied is several papers $[30, 33, 36, 37, 38, 40, 41, 46, 47, 49]$. 

Computing Classification System 1998: G.2.2
Mathematics Subject Classification 2010: 68M20, 05B15
Key words and phrases: random sequences, analysis of algorithms, Latin squares, Sudoku squares
The testing of these matrices raises the following problem.
Let \( m \geq 1 \) and \( n \geq 1 \) be integers and \( X = [x_{ij}]_{1 \leq i \leq m, 1 \leq j \leq n} \) be an \( m \times n \) sized matrix of independent random variables having joint uniform distribution

\[
\Pr\{x_{ij} = k \text{ for } 1 \leq k \leq n\} = \frac{1}{n} \quad (1 \leq i \leq m, 1 \leq j \leq n)
\]

A realization \( M = [m_{ij}] \) of \( X \) is called good, if its each row and each column contain different elements (in the case \( m = n \) a permutation of the numbers \( 1, 2, \ldots, n \)). We present and analyse algorithms which decide whether a given realization is good. If the realization is good then the output of the algorithms is True, otherwise is False.

The structure of the paper is as follows. Section 1 contains the introduction. In Section 2 the mathematical background of the main results is prepared. Section 3 contains the running times of the testing algorithms Linear, Backward, Bucket and Matrix in worst, best and expected cases. In Section 4 the results are summarised.

2 Mathematical background

We start with the first step of the testing of \( M \): describe and analyse several algorithms testing the first row of \( M \). The inputs of these algorithms are \( n \) (the length of the first row of \( M \)) and the elements of the first row \( m = (m_{11}, m_{12}, \ldots, m_{1n}) \). For the simplicity we use the notation \( s = (s_1, s_2, \ldots, s_n) \).

The output is always a logical variable \( g \) (its value is True, if the input sequence is good, and False otherwise).

We will denote the binomial coefficient \( \binom{n}{k} \) by \( B(n, k) \) and the function \( \log_2 n \) by \( \lg n \) [19], and usually omit the argument \( n \) from the functions \( \tau(n) \), \( \sigma(n) \), \( \kappa(n) \), \( \kappa_1(n) \), \( \kappa_2(n) \), \( \gamma(n) \), \( \lambda(n) \), \( \delta(n) \), \( \alpha(n) \), \( \mu(n) \), \( \eta(n) \), \( \phi(n) \), \( \rho(n) \), \( \beta(n) \), \( S_i(n) \), \( R_i(n) \), \( Q(n) \), \( p_k(n) \), \( y(n) \), \( q_i(k, n) \), \( A_i(n) \), \( b_i(n) \), \( f(n) \), \( p(i, j, k, n) \), \( c_j(n) \), \( c(n) \), and \( A(i_1, i_2, k, n) \).

We characterise the running time of the algorithms by the number of necessary assignments and comparisons and denote the running time of algorithm \( \text{ALG} \) by \( T_{\text{worst}}(n, \text{ALG}) \), \( T_{\text{best}}(n, \text{ALG}) \) and \( T_{\text{exp}}(n, \text{ALG}) \) in the worst, best, resp. expected case. The numbers of the corresponding assignments and comparisons are denoted by \( A \), resp. \( C \). The notations \( O \), \( \Omega \), \( \Theta \), \( o \) and \( \omega \) are used according to [19, pages 43–52] and [51, pages 107–110].

Before the investigation of the concrete algorithms we formulate several lemmas. The first lemma is the following version of the well-known Stirling’s formula.
Lemma 1 ([19]) If \( n \geq 1 \) then
\[
n! = \left( \frac{n}{e} \right)^n \sqrt{2\pi n e^x},
\]
where
\[
\frac{1}{12n + 1} < \tau < \frac{1}{12n},
\]
and \( \tau(n) = \tau \) tends monotonically decreasing to zero when \( n \) tends to infinity.

Let \( a_k(n) = a_k \) and \( S_i(n) = S_i \) defined for any positive integer \( n \) as follows:
\[
a_k = \frac{n^k}{k!} \quad (k = 0, 1, 2, \ldots),
\]
\[
S_i = \sum_{k=0}^{n-1} a_k k^i \quad (i = 0, 1, 2, \ldots).
\]

If in (2) \( k = i = 0 \), then \( k^i = 0 \).

Solving a problem posed by S. Ramanujan [63], Gábor Szegő [67] proved the following connection between \( e^n \) and \( S_0 \).

Lemma 2 ([67]) The function \( \sigma(n) = \sigma \), defined by
\[
\frac{e^n}{2} = S_0 + \left( \frac{1}{3} + \sigma \right) a_n = \sum_{k=0}^{n-1} \frac{n^k}{k!} + \left( \frac{1}{3} + \sigma \right) a_n \quad (n = 1, 2, \ldots)
\]
and
\[
\sigma(0) = \frac{1}{6},
\]
tends monotonically decreasing to zero when \( n \) tends to \( \infty \).

The following lemma shows the connection among \( S_i \) and \( S_0, S_1, \ldots, S_{i-1} \).

Lemma 3 If \( i \) and \( n \) are positive integers, then
\[
S_i = n \sum_{k=0}^{i-1} B(i - 1, k) S_k - n^{i-1} a_{n-1}
\]
and
\[
S_i = \Theta(e^n n^i).
\]
Proof. Omitting the member belonging to the index $k = 0$ in $S_i$, then simplifying by $k$ and using the substitution $k - 1 = j$ we get

$$S_i = \sum_{k=0}^{n-1} \frac{n^k}{k!} k^i = n \sum_{k=1}^{n-1} \frac{n^{k-1}}{(k-1)!} k^{i-1} = n \sum_{j=0}^{n-2} \frac{n^j}{j!} (j+1)^{i-1}.$$ 

Completing the sum with the member belonging to index $j = n - 1$ results

$$S_i = n \sum_{j=0}^{n-1} \frac{n^j}{j!} (j+1)^{i-1} - n^i a_{n-1} .$$

(6)

Now the application of the binomial theorem results (4).

According to (5) $S_0 = \Theta(e^n)$, so using induction and (6) we get (5). \hfill \Box

In this paper we need only the simple form of $S_0$, $S_1$, $S_2$ and $S_3$ what is presented in the next lemma.

**Lemma 4** If $n$ is a positive integer then

$$S_0 = e^n \left( 1 - \frac{n^n}{n!} \left( \frac{1}{3} + \sigma \right) \right),$$

(7)

$$S_1 = nS_0 - na_{n-1} , \quad S_2 = S_0(n^2 + n) - 2n^2 a_n ,$$

(8)

and

$$S_3 = S_0(n^3 + 3n^2 + n) - (3n^3 + 2n^2) a_n .$$

(9)

**Proof.** Expressing $S_0$ from (3), and using recursively Lemma 3 for $i = 1, 2$ and $3$ we get the required formula for $S_0$, $S_1$, $S_2$, and $S_3$. \hfill \Box

We introduce also another useful function $R_i(n) = R_i$

$$R_i = \sum_{k=1}^{n} p_k(n) k^i \quad (i = 0, 1, 2, \ldots) ,$$

(10)

where $p_k(n) = p_k$ is the key probability of this paper, defined in [33] as

$$p_k = \frac{n}{n} \frac{n-1}{n} \cdots \frac{n-k+1}{n} \frac{k}{(n-k)!n^{k+1}} \quad (k = 1, 2, \ldots, n).$$

(11)

The following lemma mirrors the connection between the function $R_i$ and the functions $S_0$, $S_1$, $\ldots$, $S_{i+1}$. 
Lemma 5 If \( i \) and \( n \) are positive integers, then
\[
R_i = \frac{n!}{n^{n+1}} \sum_{l=0}^{i+1} \binom{i+1}{l} (-1)^l n^{i+1-l} S_l.
\] (12)

Proof. Using (10) and (11) the substitution \( n - k = j \) results
\[
R_i = \sum_{k=1}^{n} \frac{n!k^{i+1}}{(n-k)!n^{k+1}} = \frac{n!}{n^{n+1}} \sum_{j=0}^{n-1} \frac{n^j(n-j)^{i+1}}{j!}.
\]
From here, using the binomial theorem we get (12). \( \square \)

In this paper we need only the following consequence of Lemma 5.

Lemma 6 If \( n \) is a positive integer, then
\[
R_0 = 1, \quad R_1 = \frac{n!}{n^n} S_0,
\]
and
\[
R_2 = 2n - \frac{n!}{n^n} S_0.
\] (13)

Proof. \( R_0 = 0 \) follows from the definition of the probabilities \( p_k \). Substituting \( i = 1 \) into (12) we get
\[
R_1 = \frac{n!}{n^{n+1}} \left( n^2 \sum_{j=0}^{n-1} \frac{n^j}{j!} - 2n \sum_{j=0}^{n-1} \frac{n^j}{j!} j + \sum_{j=0}^{n-1} \frac{n^j}{j!} j^2 \right).
\]
From here, using (2) we get
\[
R_1 = \frac{n!}{n^{n+1}}(n^2 S_0 - 2n S_1 + S_2),
\]
and using (6) the required formula for \( R_1 \).

Substituting \( i = 2 \) into (12) we get
\[
R_2 = \frac{n!}{n^{n+1}} \left( n^3 \sum_{j=0}^{n-1} \frac{n^j}{j!} - 3n^2 \sum_{j=0}^{n-1} \frac{n^j}{j!} j + 3n \sum_{j=0}^{n-1} \frac{n^j}{j!} j^2 - \sum_{j=0}^{n-1} \frac{n^j}{j!} j^3 \right).
\]
From here, using (2) we have
\[
R_2 = \frac{n!}{n^{n+1}}(n^3 S_0 - 3n^2 S_1 + 3n S_2 - S_3),
\] (14)
and using (8) and (9) the required formula for \( R_2 \). \( \square \)

The following lemmas give some further properties of \( R_1 \) and \( R_2 \).
Lemma 7 If \( n \) is a positive integer, then

\[
R_1 = \frac{n!}{n^n} S_0 = \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \kappa, \tag{15}
\]

where

\[
\kappa(n) = \kappa = \sqrt{\frac{\pi n}{2}} \left( e^\tau - 1 - \frac{2\sigma e^\tau}{e^n} \right), \tag{16}
\]

and \( \kappa \) tends monotonically decreasing to zero when \( n \) tends to infinity.

Proof. Substituting \( S_0 \) according to (7) in the formula (13) for \( R_1 \) we get

\[
R_1 = \frac{n!}{n^n} \left[ \frac{e^n}{2} - \frac{n^n}{n!} \left( \frac{1}{3} + \sigma \right) \right] = -\frac{1}{3} + \frac{n!}{n^n} \left( \frac{e^n}{2} - \frac{n^n}{n!} \sigma \right). \tag{17}
\]

Substitution of \( n! \) according to (1) (Stirling’s formula) and writing \( 1 + (e^\tau - 1) \) instead of \( e^\tau \) results

\[
R_1 = -\frac{1}{3} + \frac{1}{n^n} \left( \frac{n!}{e^n} \right)^n \sqrt{2\pi n} \left[ 1 + (e^\tau - 1) \right] \left[ \frac{e^n}{2} - \sigma \right]. \tag{18}
\]

The product \( P \) of the expressions in the square brackets is

\[
P = \frac{e^n}{2} + \frac{e^n}{2} (e^\tau - 1) - \sigma e^\tau, \tag{19}
\]

therefore

\[
R_1 = \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \sqrt{\frac{2\pi n}{e^n}} \left( \frac{e^n}{2} (e^\tau - 1) - \sigma e^\tau \right), \tag{20}
\]

implying

\[
R_1 = \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \sqrt{\frac{\pi n}{2}} (e^\tau - 1) - \sqrt{\frac{\pi n}{2}} 2\sigma e^\tau. \tag{21}
\]

Let

\[
\kappa_1(n) = \kappa_1 = \sqrt{\frac{\pi n}{2}} (e^\tau - 1), \quad \kappa_2(n) = \kappa_2 = \sqrt{\frac{\pi n}{2}} 2\sigma e^\tau, \quad \kappa = \kappa_1 + \kappa_2, \tag{22}
\]

and

\[
\gamma(n) = \gamma = \frac{\kappa(n + 1)}{\kappa(n)} = \frac{\kappa_1(n + 1) - \kappa_2(n + 1)}{\kappa_1(n) - \kappa_2(n)} \quad \text{for } n = 1, 2, \ldots. \tag{23}
\]
Since all $\kappa$ functions are positive for all positive integer $n$'s, therefore $\gamma < 1$ for $n \geq 1$ implies the monotonicity of $\kappa$. Numerical results in Table 1 show that $\gamma < 1$ for $n = 1, 2, \ldots, 9$, therefore it remained to show $\gamma < 1$ for $n \geq 10$.

$\kappa_2(n + 1)$ can be omitted from the numerator of (22). Since $\sigma$ and $\tau$ are monotone decreasing functions, and $0 < \sigma(5) < 0.0058$, and $0 < e^{\tau(5)} < 1.02$, and $n^2 < e^n$ for $n \geq 10$, therefore

$$\frac{2\sigma e^{\tau}}{e^n} < \frac{2 \cdot 0.0058 \cdot 1.02}{e^n} < \frac{0.012}{n^2} \quad \text{for } n \geq 10.$$  \hfill (24)

Using (23), (24) and the Lagrange remainder of the Taylor series of the function $e^x$ we have

$$\gamma < \frac{\sqrt{n + 1}}{\sqrt{n}} \frac{\tau(n + 1) + \tau^2 \xi_{n+1}/2}{\tau(n) + \tau^2 / 2 - 0.012/n^2},$$

where $0 < \xi_{n+1} < n + 1$ and $0 < \xi_n < n$, therefore using Lemma 1 we get

$$\gamma < \frac{\sqrt{n + 1}}{\sqrt{n}} \frac{1}{\frac{12(n+1)+1}{12n} + \frac{1}{2} \left( \frac{1}{12n} \right)^2 - \frac{0.012}{n^2}}.$$  \hfill (25)

Now multiplication of the denominator and denominator of the right side of (25) by $(12n)^2$ results

$$\gamma = \frac{\sqrt{n + 1}}{\sqrt{n}} \frac{12n+12n}{12n+15} = \frac{\sqrt{n + 1}}{\sqrt{n}} \frac{12n}{(12n - 1.084) \left( 1 + \frac{13}{12n} \right)}.$$  \hfill (26)

Since

$$(12n - 1.084) \left( 1 + \frac{13}{12n} \right) > 12n + 10,$$  \hfill (27)

(26) and (27) imply

$$\gamma < \frac{\sqrt{144n^3 + 144n^2}}{\sqrt{144n^3 + 240n^2}} < 1,$$

finishing the proof of the monotonicity of $\kappa$. \hfill \Box

We remark, that the monotonicity of $\kappa$ was published in [40] without proof, and was proved by E. Bokova and G. Tzaturjan in 1985 [9], and in 1988—using a formula due to E. Egorychev et al. [25] derived by the method of integral representation of combinatorial sums elaborated by E. P. Egorychev [24]—by T. T. Cirulis and A. Iványi [17]. Our proof is much simpler than the earlier ones.
Lemma 8 If \( n \) is a positive integer, then
\[
R_2 = 2n - \frac{n!}{n^n} S_0 = 2n + \frac{1}{3} - \sqrt{\frac{\pi n}{2}} e^\tau - \lambda,
\]
where
\[
\lambda = \sqrt{\frac{\pi n}{2}} (e^\tau - 1) + \sigma,
\]
and \( \lambda \) tends monotonically decreasing to zero when \( n \) tends to infinity.

Proof. The proof is omitted since it is similar to the proof of Lemma 7. \( \square \)

3 Running times of the algorithms

In the following analysis let \( n \geq 1 \) and let \( x = (x_1, x_2, \ldots, x_n) \) be independent random variables having uniform distribution on the set \( \{1, 2, \ldots, n\} \). The input sequence of the algorithms is \( s = (s_1, s_2, \ldots, s_n) \) (a realization of \( x \)).

We derive exact formulas for the expected numbers of comparisons \( C_{\text{exp}}(n, \text{LINEAR}) = C_L, \ C_{\text{exp}}(n, \text{BACKWARD}) = C_W, \) and \( C_{\text{exp}}(n, \text{BUCKET}) = C_B \), further for the expected running times \( T_{\text{exp}}(n, \text{LINEAR}) = T_L, \ T_{\text{exp}}(n, \text{BACKWARD}) = T_W, \) and \( T_{\text{exp}}(n, \text{BUCKET}) = T_B \).

The inputs of the following algorithms are \( n \) (the length of the sequence \( s \)) and \( s = (s_1, s_2, \ldots, s_n) \), a sequence of nonnegative integers with \( 1 \leq s_i \leq n \) for \( 1 \leq i \leq n \) in all cases. The output is always a logical variable \( g \) (its value is \( \text{TRUE} \), if the input sequence is good, and \( \text{FALSE} \) otherwise). The working variables are usually the cycle variables \( i \) and \( j \).

We use the pseudocode defined in [19].

3.1 Definition and running time of algorithm LINEAR

LINEAR writes zero into the elements of an \( n \) length vector \( v = (v_1, v_2, \ldots, v_n) \), then investigates the elements of the realization \( s \) and if \( v_{s_i} > 0 \) (signalising a repetition), then returns \( \text{FALSE} \), otherwise adds 1 to \( v_k \). If LINEAR does not find a repetition among the elements of \( s \) then it returns finally \( \text{TRUE} \).

\[
\text{LINEAR}(n, s)
\]
1 \( g \leftarrow \text{TRUE} \)
2 for \( i \leftarrow 1 \) to \( n \)
3 \( v_i \leftarrow 0 \)
Testing of sequences by simulation

4 for \( i \leftarrow 1 \) to \( n \)
5 \( \text{if } v_{s_i} > 0 \)
6 \( g \leftarrow \text{FALSE} \)
7 return \( g \)
8 else \( v_{s_i} \leftarrow v_{s_i} + 1 \)
9 return \( g \)

**Linear** needs assignments in lines 1, 3, and 8, and it needs comparisons in line 5. The number of assignments in lines 1 and 3 equals to \( n + 1 \) for arbitrary input and varies between 1 and \( n \) in line 8. The number of comparisons in line 8 also varies between 1 and \( n \). Therefore the running time of **Linear** is \( \Theta(n) \) in the best, worst and expected case too.

The following theorem gives the expected number of the comparisons of **Linear**.

**Theorem 9** The expected number of comparisons \( C_{\text{exp}}(n, \text{Linear}) = C_L \) of **Linear** is

\[
C_L = 1 - \frac{n!}{n^n} + R_1 = \sqrt{\frac{\pi n}{2}} + \frac{2}{3} + \kappa - \frac{n!}{n^n}. \tag{29}
\]

where

\[
\kappa = \frac{1}{3} - \sqrt{\frac{\pi n}{2}} + \sum_{k=1}^{n} \frac{n!k^2}{(n-k)!n^{k+1}}
\]

tends monotonically decreasing to zero when \( n \) tends to infinity.

**Proof.** Let

\[
y(n) = y = \max\{k : 1 \leq k \leq n \text{ and } s_1, s_2, \ldots, s_k \text{ are different}\} \tag{30}
\]

be a random variable characterising the maximal length of the prefix of \( s \) containing different elements. Then

\[
\Pr(y = k) = p_k \quad (k = 1, 2, \ldots, n),
\]

where \( p_k \) is the probability introduced in (11).

If \( y = k \) and \( 1 \leq k \leq n - 1 \), then **Linear** executes \( k + 1 \) comparisons, and only \( n \) comparisons, if \( y = n \), therefore

\[
C_L = \sum_{k=1}^{n-1} p_k(k + 1) + p_n n = \sum_{k=1}^{n} p_k(k + 1) - p_n = 1 - \frac{n!}{n^n} + \sum_{k=1}^{n} p_k k. \tag{31}
\]
from where using Lemma 7 we receive
\[ C_L = 1 - \frac{n!}{n^n} + R_1 = \sqrt{\frac{\pi n}{2}} + \frac{2}{3} - \frac{n!}{n^n} + \kappa. \] (32)

The monotonity of \( \kappa(n) \) was proved in the proof of Lemma 7. □

The next assertion gives the running time of LINEAR.

**Theorem 10** The expected running time \( T_{\text{exp}}(n, \text{LINEAR}) = T_L \) of LINEAR is
\[ T_L = n + \sqrt{2\pi n} + \frac{7}{3} + 2\kappa - 2\frac{n!}{n^n}, \]
where \( \kappa \) tends monotonically decreasing to zero when \( n \) tends to infinity.

**Proof.** LINEAR requires \( n+1 \) assignments in lines 01 and 03, plus assignments in line 08. The expected number of assignments in line 8 is the same as \( C_L \). Therefore
\[ T_L = n + 1 + 2C_L. \] (33)
Substitution of (32) into (33) results the required (29). □

We remark, that (32) is equivalent with
\[ C_L = 1 - \frac{n!}{n^n} + 1 + \frac{n - 1}{n} + \frac{n - 1}{n} \frac{n - 2}{n} + \cdots + \frac{n - 1}{n} \frac{n - 2}{n} \cdots \frac{1}{n}, \]
demonstrating the close connection with the function
\[ Q(n) = Q = C_L - 1 + \frac{n!}{n^n}, \] (34)
studied by several authors, e.g. in [12, 40, 51].

Table 1 shows the concrete values of the functions appearing in the analysis of \( C_L \) and \( T_L \) for \( 1 \leq n \leq 10 \), where \( C_L \) was calculated using (32), \( \kappa \) using (11), and \( \sigma \) using (3) (data in this and further tables are taken from [43]). We can observe in Table 1 that \( \delta(n) = \delta = \kappa - \frac{n!}{n^n} \) is increasing from \( n = 1 \) to \( n = 8 \), but for larger \( n \) is decreasing. Taking into account that for \( n > 8 \)
\[ \frac{n!}{n^n} = \left( \frac{n}{e} \right)^n \sqrt{2\pi n} \frac{e^n}{n^n} \sqrt{2\pi n} \frac{e^n}{e^n} e^{1/(12n)} < \frac{2.7\sqrt{n}}{e^n} < 0.012 \frac{n^2}{n^2} \]
holds, we can prove—using the same arguments as in the proof of Lemma 7—the following assertion.
Testing of sequences by simulation

Table 1: Values of $C_L$, $u = \sqrt{\pi n/2} + 2/3$, $n!/n^n$, $\kappa$, $\delta = \kappa - n!/n^n$, and $\sigma$ for $n = 1, 2, \ldots, 10$

| n  | $C_L$  | $u$    | $n!/n^n$ | $\kappa$ | $\delta$ | $\sigma$ |
|----|--------|--------|----------|----------|----------|----------|
| 1  | 1.000000 | 1.919981 | 1.000000 | 0.080019 | -0.919981 | 0.025808 |
| 2  | 2.000000 | 2.439121 | 0.500000 | 0.060879 | -0.439121 | 0.013931 |
| 3  | 2.666667 | 2.837470 | 0.222222 | 0.051418 | -0.170804 | 0.009504 |
| 4  | 3.125000 | 3.173295 | 0.093750 | 0.045455 | -0.048295 | 0.007205 |
| 5  | 3.472000 | 3.49162 | 0.041238 | 0.041238 | +0.002838 | 0.005799 |
| 6  | 3.759259 | 3.73647 | 0.015432 | 0.038045 | +0.022612 | 0.004852 |
| 7  | 4.012019 | 3.982624 | 0.006120 | 0.035515 | +0.029395 | 0.004170 |
| 8  | 4.242615 | 4.211574 | 0.006120 | 0.035515 | +0.029395 | 0.004170 |
| 9  | 4.457379 | 4.426609 | 0.006120 | 0.035515 | +0.029395 | 0.004170 |
| 10 | 4.659853 | 4.629994 | 0.006120 | 0.035515 | +0.029395 | 0.004170 |

**Theorem 11** The expected running time $T_{\text{exp}}(n, \text{LINEAR}) = T_L$ of LINEAR is

$$T_L = n + \sqrt{2\pi n} + \frac{7}{3} + \delta,$$

where $\delta(n) = \delta$ tends to zero when $n$ tends to infinity, further

$$\delta(n+1) > \delta(n) \text{ for } 1 \leq n \leq 7 \text{ and } \delta(n+1) < \delta(n) \text{ for } n \geq 8.$$

If we wish to prove only the existence of some threshold index $n_0$ having the property that $n \geq n_0$ implies $\delta(n+1) < \delta(n)$, then we can use the following shorter proof.

Using (29) and (34) we get

$$\kappa = C_L - \sqrt{\frac{\pi n}{2} - \frac{2}{3} - \frac{n!}{n^n}} = Q - \sqrt{\frac{\pi n}{2} + \frac{1}{3}}.$$  \hspace{1cm} (35)

Substituting the power series

$$Q = \sqrt{\frac{\pi n}{2} - \frac{1}{3} + \frac{1}{12} \frac{\pi}{2n} - \frac{14}{135n} + \frac{1}{288n^3} + O(n^{-2})}$$

cited by D. E. Knuth [51, Equation (25) on page 120] into (35) and using

$$\frac{1}{n^{k/2}} - \frac{1}{(n+1)^{k/2}} = \Theta\left(\frac{1}{n^{1+k/2}}\right)$$
for \( k = 1, 2, 3 \) and 4 we get
\[
\kappa(n) - \kappa(n+1) = \frac{\sqrt{\pi}}{12\sqrt{2}} \left( \frac{1}{\sqrt{n}} - \frac{1}{\sqrt{n+1}} \right) + O(n^{-2}),
\]
implying
\[
\kappa(n) - \kappa(n+1) = \frac{\sqrt{\pi}}{12\sqrt{2}} \sqrt{n/\sqrt{n+1}} + O(n^{-2}),
\]
guaranteeing the existence of the required \( n_0 \).

### 3.2 Running time of algorithm Backward

Backward compares the second \( (s_2) \), third \( (s_3) \), . . . , last \( (s_n) \) element of the realization with the previous elements until the first collision or until the last pair of elements.

Taking into account the number of the necessary comparisons in line 04 of Backward, we get: \( C_{\text{best}}(n, \text{Backward}) = 1 = \Theta(1) \), and \( C_{\text{worst}}(n, \text{Backward}) = B(n, 2) = \Theta(n^2) \). The number of assignments is 1 in the best case (in line 1) and is 2 in the worst case (in lines 1 and in line 5). The expected number of assignments is \( A_{\text{exp}}(n, \text{Backward}) = 1 + \frac{n!}{n^n} \), since only the good realizations require the second assignment.

**Backward**(n, s)
1 \( g \leftarrow \text{True} \)
2 for \( i \leftarrow 2 \) to \( n \)
3 \quad for \( j \leftarrow i - 1 \) down to 1
4 \quad \quad if \( s_i = s_j \)
5 \quad \quad \quad \( g \leftarrow \text{False} \)
6 \quad return \( g \)
7 return \( g \)

The next assertion gives the expected running time.

**Theorem 12** The expected number of comparisons \( C_{\text{exp}}(n, \text{Backward}) = C_W \) of the algorithm Backward is
\[
C_W = n - \sqrt{\frac{\pi n}{8}} + \frac{2}{3} - \frac{1}{2} \kappa - \frac{n! n + 1}{n^n} = \sqrt{\frac{\pi n}{8}} + \frac{2}{3} - \alpha,
\]
where \( \alpha(n) = \alpha = \frac{\kappa}{2} + \frac{n! n + 1}{2} \) monotonically decreasing tends to zero when \( n \) tends to \( \infty \).
Proof. Let \( y \) be as defined in (30), \( p_k \) as defined in (11), and let
\[
z = \{ q : 1 \leq q \leq k; s_1, s_2, \ldots, s_k \text{ are different}; s_{k+1} = s_q \mid y = k \}
\]
be a random variable characterising the index of the first repeated element of \( s \).

Let
\[
q_i(k, n) = q_i(k) = \Pr[z = i \mid y = k] \quad (k = 1, 2, \ldots, n; i = 1, 2, \ldots k).
\]

Backward executes \( B(k, 2) \) comparisons among the elements \( s_1, s_2, \ldots, s_k \), and \( s_{k+1} \) requires at least 1 and at most \( k \) comparisons (with exception of case \( k = n \) when additional comparisons are not necessary). Therefore using the theorem of the full probability we have
\[
C_W = \sum_{k=1}^{n-1} p_k \left( B(k, 2) + \sum_{i=1}^{k} i q_i(k) \right) + p_n B(n, 2),
\]
where
\[
q_i(k, n) = q_i(k) = \frac{1}{k} \quad (i = 1, 2, \ldots, k; k = 1, 2, \ldots, n). \tag{36}
\]

Adding a new member to the first sum we get
\[
C_W = \sum_{k=1}^{n} p_k \left( B(k, 2) + \sum_{i=1}^{k} i q_i(k) \right) - p_n \sum_{i=1}^{n} q_i(k) i. \tag{37}
\]

Using the uniform distribution (36) of \( z \) we can determine its contribution to \( C_W \):
\[
\sum_{i=1}^{k} q_i(k) i = \sum_{i=1}^{k} \frac{i}{k} = \frac{k+1}{2}. \tag{38}
\]

Substituting the contribution in (38) into (37), and taking into account Lemma 6 and Lemma 7 we have
\[
C_W = \frac{1}{2} R_2 - \frac{1}{2} R_3 - \frac{n!}{n^n} \frac{n + 1}{2}.
\]

Now Lemma 6 and Lemma 7 result
\[
C_W = n - \sqrt{\frac{\pi n}{8}} + \frac{2}{3} - \frac{1}{2} \kappa - \frac{n!}{n^n} \frac{n + 1}{2}. \tag{39}
\]

The known decreasing monotonicity of \( \kappa \) and \( \frac{n!}{n^n} \) imply the decreasing monotonicity of \( \alpha \).
Theorem 13 The expected running time $T_{\text{exp}}(n, \text{BACKWARD}) = T_W$ of the algorithm BACKWARD is

$$T_W = n - \sqrt{\frac{\pi n}{8}} + \frac{5}{3} - \alpha,$$

(40)

where $\alpha = \kappa/2 + (n!/n^n)((n + 1)/2)$ tends monotonically decreasing to zero when $n$ tends to $\infty$.

Proof. Taking into account (39) and $A_{\text{exp}}(n, \text{BACKWARD}) = 1 + \frac{n!}{n^n} - \frac{n!}{n^n} \frac{n+1}{2}$ we get (40).

Table 2 represents some concrete numerical results. It is worth to remark that $\frac{n!}{n^n} \frac{n+1}{2} = \Theta \left( \frac{n\sqrt{n}}{e^n} \right)$, while $\kappa = \Theta \left( \frac{1}{\sqrt{n}} \right)$, therefore $\kappa$ decreases much slower than the other expression.

### 3.3 Running time of algorithm Bucket

Bucket divides the interval $[1,n]$ into $m = \sqrt{n}$ subintervals $I_1$, $I_2$, …, $I_m$, where $I_j = [(j-1)m+1, jm]$ for $j = 1, 2, \ldots m$, and sequentially puts the elements of $s$ into the bucket $B_j$ (we use the word bucket due to some similarity to bucket sort [19]): if $[s_i/m] = j$, then $s_i$ belongs to $B_j$. Bucket works until the first repetition (stopping with $g = \text{FALSE}$), or up to the processing of the last element $s_n$ (stopping with $g = \text{TRUE}$).

| $n$ | $C_W$ | $n - \sqrt{\pi n/8} + 2/3$ | $t$ | $\kappa$ | $\alpha$ |
|---|---|---|---|---|---|
| 1 | 0.000000 | 1.040010 | 1.000000 | 0.080019 | 1.040010 |
| 2 | 1.000000 | 1.780440 | 0.750000 | 0.060879 | 0.780440 |
| 3 | 2.111111 | 2.581265 | 0.444444 | 0.051418 | 0.470154 |
| 4 | 3.156250 | 3.413353 | 0.234375 | 0.045455 | 0.257103 |
| 5 | 4.129600 | 4.265419 | 0.115200 | 0.041238 | 0.135819 |
| 6 | 5.058642 | 5.131677 | 0.054012 | 0.038045 | 0.073035 |
| 7 | 5.966451 | 6.008688 | 0.024480 | 0.035515 | 0.042237 |
| 8 | 6.866676 | 6.894213 | 0.010815 | 0.033444 | 0.027536 |
| 9 | 7.766159 | 7.786695 | 0.004683 | 0.031707 | 0.020537 |
| 10 | 8.667896 | 8.685003 | 0.001996 | 0.030222 | 0.017107 |
Bucket handles an array $Q[1:m, 1:m]$ (where $m = \lceil \sqrt{n} \rceil$) and puts the element $s_i$ into the $r$th row of $Q$, and it tests using linear search whether $s_j$ appeared earlier in the corresponding bucket. The elements of the vector $c = (c_1, c_2, \ldots, c_m)$ are counters, where $c_j$ $(1 \leq j \leq m)$ shows the actual number of elements in $B_j$.

Bucket $(n, s)$

1. $g \leftarrow$ True
2. $m \leftarrow \sqrt{n}$
3. for $j \leftarrow 1$ to $m$
   4. $c_j \leftarrow 1$
5. for $i \leftarrow 1$ to $n$
   6. $r \leftarrow \lceil s_i/m \rceil$
   7. for $j \leftarrow 1$ to $c_r - 1$
   8. if $s_i = Q_{r,j}$
      9. $g \leftarrow$ False
   10. return $g$
11. $Q_{r,c_r} \leftarrow s_i$
12. $c_r \leftarrow c_r + 1$
13. return $g$

For the simplicity let us suppose that $m$ is a positive integer and $n = m^2$.

In the best case $s_1 = s_2$. Then Bucket executes 1 comparisons in line 8, $m$ assignments in line 4, and 1 assignment in line 1, 1 in line 2, 2 in line 6, and 1 in line 8, 11 and 12, therefore $T_{\text{best}}(n, \text{Bucket}) = m + 7 = \Theta(\sqrt{n})$. The worst case appears, when the input is bad. Then each bucket requires $1 + 2 + \cdots + m - 1 = B(n - 1, 2)$ comparisons in line 8, further $3m$ assignments in lines 6, and 12, totally $\frac{m^2(m-1)}{2} + 3m^2$ operations. Lines 1, 2, and 9 require 1 assignment per line, and the assignment in line 4 is repeated $m$ times. So $T_{\text{worst}}(n, \text{Bucket}) = \frac{m^2(m-1)}{2} + 3m^2 + m + 3 = \Theta(n^{3/2})$.

In connection with the expected behaviour of Bucket at first we show that the expected number of elements in a bucket has a constant bound which is independent from $n$.

**Lemma 14** Let $b_j(n) = b_j$ $(j = 1, 2, \ldots, m)$ be a random variable characterising the number of elements in the bucket $B_j$ at the moment of the first repetition. Then

$$E[b_j] = \sqrt{\frac{\pi}{2}} - \mu \quad \text{for } j = 1, 2, \ldots, m, \quad (41)$$
where
\[ \mu(n) = \mu = \frac{1}{3\sqrt{n}} - \frac{\kappa}{\sqrt{n}}, \tag{42} \]
and \( \mu \) tends monotonically decreasing to zero when \( n \) tends to infinity.

**Proof.** Due to the symmetry of the buckets it is sufficient to prove (41) and (42) for \( j = 1 \).

Let \( m \) be a positive integer and \( n = m^2 \). Let \( y \) be the random variable defined in (28) and \( p_k \) be the probability defined in (11).

Let \( A_i(n) = A_i \ (i = 1, 2, \ldots, n) \) be the event that the number \( i \) appears in \( s \) before the first repetition and \( Y_i(n) = Y_i \) be the indicator of \( A_i \). Then using the theorem of the full probability we have
\[
E(b_1) = \sum_{i=1}^{m} Y_i = \sum_{i=1}^{m} \Pr(A_i) = m \Pr(A_1)
\]
and
\[
\Pr(A_1) = \Pr(1 \in \{s_1, s_2, \ldots, s_k\} | y = k) = \sum_{k=1}^{n} p_k \frac{k}{n} = \frac{1}{n} \sum_{k=1}^{n} p_k k = \frac{1}{n} R_1.
\]

Using Lemma 7, we get
\[
E(b_1) = m \frac{R_1}{n} = m \frac{\left( \frac{\sqrt{m}}{2} - \frac{1}{3} + \frac{\kappa}{2} \right)}{n},
\]
resulting (41) and (42).

We omit the proof of the monotony of \( \mu \), since it is similar to the corresponding part in the proof of Lemma 7. \( \square \)

Table 3 shows some concrete values.

**Lemma 15** Let \( f(n) = f \) be a random variable characterising the number of comparisons executed in connection with the first repeated element. Then
\[
E(f) = 1 + \sqrt{\frac{\pi}{8}} - \eta,
\]
where
\[
\eta(n) = \eta = \frac{1/6 + \sqrt{\pi/8} - \kappa/2}{\sqrt{n} + 1},
\]
and \( \eta \) tends monotonically decreasing to zero when \( n \) tends to infinity.
Testing of sequences by simulation

Table 3: Values of $E\{b_1\}$, $\sqrt{\pi/2}$, $1/(3\sqrt{n})$, $\kappa/\sqrt{n}$, and $\mu = 1/(3\sqrt{n}) - \kappa/\sqrt{n}$ for $n = 1, 2, \ldots, 10$

| $n$ | $E\{b_1\}$ | $\sqrt{\pi/2}$ | $1/(3\sqrt{n})$ | $\kappa/\sqrt{n}$ | $\mu$ |
|-----|-------------|-----------------|-----------------|-----------------|------|
| 1   | 1.000000    | 1.253314        | 0.333333        | 0.080019        | 0.253314 |
| 2   | 1.060660    | 1.253314        | 0.235702        | 0.043048        | 0.192654 |
| 3   | 1.090055    | 1.253314        | 0.192450        | 0.029686        | 0.162764 |
| 4   | 1.109375    | 1.253314        | 0.166667        | 0.022727        | 0.143940 |
| 5   | 1.122685    | 1.253314        | 0.149071        | 0.018442        | 0.130629 |
| 6   | 1.132763    | 1.253314        | 0.136083        | 0.015532        | 0.120627 |
| 7   | 1.140740    | 1.253314        | 0.125988        | 0.013423        | 0.112565 |
| 8   | 1.147287    | 1.253314        | 0.117851        | 0.011824        | 0.106027 |
| 9   | 1.152772    | 1.253314        | 0.111111        | 0.010569        | 0.100542 |
| 10  | 1.157462    | 1.253314        | 0.105409        | 0.009557        | 0.095852 |

Proof. Let $p(i, j, k, n) = p(i, k, n)$ be the probability of the event that there are $k$ different elements before the first repetition, and the repeated element belongs to $B_j$, and $B_j$ contains $i$ elements in the moment of the first repetition. Due to the symmetry $p(i, j, k, n)$ does not depend on $j$ and

$$p(i, j, k, n) = \binom{m}{i} \binom{n-m}{k-i} \frac{k!}{n^{k+1}},$$

since we investigate $n^{k+1}$ sequences, and if there are $k$ ($1 \leq k \leq n$) different elements before the repeated one, then we can choose $i$ elements for the $j$th bucket in $\binom{m}{i} \binom{n-m}{k-i}$ manner, we can permute them in $k!$ manner, and we can choose the repeated element in $i$ manner. Then

$$E(f) = \sum_{i,j,k,n} p(i, j, k, n) \frac{i + 1}{2} - mp_n$$

$$= \frac{m}{2n} \sum_{k=1}^{n} \frac{k!}{n^k} \sum_{i=1}^{k} \binom{m}{i} \binom{n-m}{k-i} i(i+1) - p_n \frac{n+1}{2}$$

The last member of the formula takes into account that if $k = n$, then additional comparisons with the elements of the bucket corresponding to the repeated element are not necessary.

Let

$$E'(f) = E(f) + p_n \frac{n + 1}{2}.$$
Then dividing the inner sum in (44) by \( \binom{n}{k} \) we get the expected value of the random variable \( \xi(\xi + 1) \), where \( \xi \) has hypergeometric distribution with parameters \( n, m, \) and \( k \). It is easy to compute that

\[
E'(\xi(\xi + 1)) = E'(\xi)E'(\xi + 1) + \text{Var}\{\xi\} = \frac{km(km - 1 + (2n - 1 - m))}{n(n - 1)},
\]

therefore

\[
E'(f) = \frac{m}{2n} \sum_{k=1}^{n} \frac{k! \binom{n}{k}}{k^m} \frac{km(km - 1 + (2n - 1 - m))}{n(n - 1)}
\]

\[= \frac{1}{2(n - 1)} \sum_{k=1}^{n} p_k[km - 1 + (2n - 1 - m)]
\]

\[= \frac{m - 1}{2(n - 1)} R_1 + \frac{2n - 1 - m}{2(n - 1)} = \frac{2m + 1 + R_1}{2m + 2}
\]

\[= 1 + \frac{\sqrt{\frac{\pi}{8}} - 1/6 + \sqrt{\frac{\pi}{8}} - \kappa/2}{\sqrt{n + 1}}.
\]

The convergence and monotonicity of \( \eta \) is the consequence of the properties of \( \kappa \). Taking into account the small value of \( p_n \) (see equation (11)) the difference \( E'(f) - E(f) \) has negligible influence on the limit of \( E(f) \). \( \square \)

**Theorem 16** The expected number of comparisons \( C_{\exp}(n, \text{Bucket}) = C_B \) of \( \text{Bucket} \) is

\[C_B = \sqrt{n} + \frac{1}{3} - \sqrt{\frac{\pi}{8}} + \rho,
\]

where

\[\rho(n) = \rho = \frac{5/6 - \sqrt{9\pi/8} - 3\kappa/2}{\sqrt{n + 1}}.
\]

and \( \rho \) tends monotonically decreasing to zero when \( n \) tends to infinity.

**Proof.** Let \( s = (s_1, s_2, \ldots, s_n) \) be the input sequence of the algorithm \( \text{Bucket} \). \( \text{Bucket} \) processes the input sequence using \( m = \sqrt{n} \) buckets \( B_1, B_2, \ldots, B_n \): it investigates the input elements sequentially and if the \( i \)-th input element \( s_i \) belongs to the interval \([ (r - 1)m + 1, (r - 1)m + 2, \ldots, rm \], then it sequentially compares \( s_i \) with the elements in the bucket \( B_r \) and finishes, if it finds a collision, or puts \( s_i \) into \( B_r \), if \( s_i \) differs from all elements in \( B_r \).
Let $y$ be the random variable, defined in (30), and $p_k$ the probability defined in (11). Let $b_i$ be the random variable defined in Lemma 14, and $c_j(n) = c_j$ ($j = 1, 2, \ldots, m$) be a random variable characterising the number of comparisons executed in $B_j$ before the processing of the first repeated element, and $c(n) = c$ a random variable characterising the number of necessary comparisons executed totally by BUCKET. Then due to the symmetry we have

$$C_B = E \left\{ \sum_{j=1}^{m} c_j \right\} + E\{f\} = mE\{c_1\} + E\{f\}.$$  \hspace{1cm} (50)

The probability of the event $A(i_1, i_2, k, n) = A(i_1, i_2, k)$ that the elements $i_1$ and $i_2$ ($1 \leq i_1, i_2 \leq m$) will be compared before the processing of the first repeated element at the condition that $y = k$ and $2 \leq k \leq n$ equals to

$$Pr(A(i_1, i_2, k)|y = k \text{ and } 2 \leq k \leq n) = \frac{(n-k-2)(n-k)}{(n-1)(n-2)} = \frac{k(k-1)}{n(n-1)},$$

Since there are $\binom{m}{n}$ possible comparisons among the elements of the interval $[1, m]$, we have

$$E\{c_1\} = \sum_{k=1}^{n} p_k \frac{k(k-1)}{n(n-1)} \left( \frac{m}{2} \right) = \frac{m(m-1)}{2n(n-1)} \left( \sum_{k=1}^{n} p_k k^2 - \sum_{k=1}^{n} p_k k \right),$$

from where using Lemma 7 and Lemma 8 we get

$$E\{c_1\} = \frac{n - \sqrt{n}}{2n^2 - 2n} (R_2 - R_1) = \frac{1}{2n + 2\sqrt{n}} \left[ 2n - 2 \left( \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \kappa \right) \right]. \hspace{1cm} (51)$$

This equality implies

$$E\{c_1\} = 1 - \frac{1}{\sqrt{n} + 1} \left( \sqrt{\frac{\pi}{8} + \frac{2}{3} - \kappa} \right). \hspace{1cm} (52)$$

From (50), taking into account (52), (45), and (47) we get

$$C_B = \sqrt{n} + \frac{1}{3} - \sqrt{\frac{\pi}{8} + \frac{9\pi/8 + 5/6 - 3\kappa/2}{\sqrt{n} + 1}}.$$  \hspace{1cm} (51)

Denoting the last fraction by $\rho$ we get the required (48). The monotony of $\rho$ is the consequence of the monotony of $\kappa$. $\square$
The expected running time $T_{\text{exp}}(n, \text{Bucket}) = T_B$ of Bucket is

$$T_B = \sqrt{n} \left( 3 + 3 \sqrt{\frac{\pi}{2}} \right) + \sqrt{\frac{25\pi}{8}} + \phi,$$

where

$$\phi(n) = \phi = 3\kappa - \rho - 3\eta - \frac{n!}{n^n} - \frac{3\sqrt{\pi/8} - 1/3 - 3\kappa/2}{\sqrt{n} + 1},$$

and $\phi$ tends to zero when $n$ tends to infinity.

**Proof.** Bucket requires 2 assignments in lines 1 and 2, $\sqrt{n}$ assignments in line 4, $R_1$ assignments in line 6, $C_B + \text{E}(f)$ assignments in line 8, $1 - p_n$ expected assignment in line 9 and $2R_1$ assignments in lines 11 and 12 before the first repeated element, and $2\text{E}(f) - 1$ assignments after the first repeated element.

Therefore the expected number $A_{\text{exp}}(n, \text{Bucket}) = A_B$ of assignments of Bucket is

$$A_B = 2 + \sqrt{n} + 3R_1 + C_B + 3\text{E}(f) - \frac{n!}{n^n}.$$

Substituting $R_1$, and $C_B$, and $\text{E}(f)$ we get

$$A_B = 2\sqrt{n} + \frac{13}{3} + 3 \sqrt{\frac{n!}{2}} + 3\kappa - \sqrt{\frac{\pi}{8}} + \rho + 3 \sqrt{\frac{\pi}{8}} - 3\eta - \frac{n!}{n^n},$$

implying

$$A_B = \sqrt{n} \left( 2 + 3 \sqrt{\frac{\pi}{2}} \right) + \frac{13}{3} + \sqrt{\frac{\pi}{2}} + 3\kappa + \rho - 3\eta - \frac{n!}{n^n}.$$

Summing up the expected number of comparisons in (48) and of assignments in (54) we get the final formula (53). \qed

### 3.4 Test of random arrays

**Matrix** is based on Bucket.

For the simplicity let us suppose that $n$ is a square.

Let $M$ be an $n \times n$ sized matrix, where $m_{ij} \in \{1, 2, \ldots, n\}$. The $i$th row of $M$ is denoted by $r_i$, and the $j$th column by $c_j$ for $1 \leq i, j \leq n$. The matrix $M$ is called *good*, if its all lines (rows and columns) contain a permutation of the elements $1, 2, \ldots, n$. 
Matrix(n, M)
1  g ← True
2  Bucket(n, r_1)
3  if g = False
4    return g
5  for i ← 2 to n
6    Bucket(n, r_i)
7    if g = False
8      return g
9  for j ← 1 to n
10   Bucket(n, c_j)
11  if g = False
12    return g
13  return g

**Theorem 18** The expected running time \( T_{\text{exp}}(n, \text{Matrix}) = T_M \) of \( \text{Matrix} \) is
\[
T_M = T_B + o(1).
\] (55)

**Proof.** According to Theorem 17 we have
\[
T_B = \sqrt{n} \left( 3 + 3 \sqrt{\frac{n}{2}} + \sqrt{\frac{25\pi}{8}} + o(1) \right).
\]
Since the rows of \( \mathcal{M} \) are independent, therefore the probability of the event \( G_k(n) = G_k \) (\( k = 1, 2, \ldots, n \)) that the first \( k \) rows are good is
\[
\Pr(G_k) = \left( \frac{n!}{n^n} \right)^k,
\]
so for the expected time \( T_{\text{exp}}(n, \text{Matrix}) = T_R \) of the testing of the rows we have
\[
T_R \leq T_B + T_B \sum_{k=1}^{n-1} \left( \frac{n!}{n^n} \right)^k = T_B + o(1).
\]
Since the columns are also independent, all the rows and the first \( k \) columns are good with the probability
\[
p = \left( \frac{n!}{n^n} \right)^{n+k},
\]
and so for the expected time of testing of the columns $T_{\text{exp}}(n, \text{Matrix}) = T_C$ holds

$$T_C \leq T_B \sum_{k=0}^{n-1} \left( \frac{n!}{n^k} \right) = o(1),$$

and so

$$T_M = T_R + T_C$$

implies (55).

## 4 Summary

Table 4 summarises the basic properties of the number of necessary comparisons of the investigated algorithms.

Table 5 summarises the basic properties of the running times of the investigated algorithms.

We used in our calculations the RAM computation model [19]. If the investigated algorithms run on real computers then we have to take into account also the limited capacity of the memory locations and the increasing execution time of the elementary arithmetical and logical operations.
Testing of sequences by simulation

Acknowledgements

Authors thank Tamás F. Móri [58] for proving Lemmas 14 and 15, Péter Burcsi [14] for useful information on references (both are teachers of Eötvös Loránd University) and the unknown referee for the useful corrections.

The European Union and the European Social Fund have provided financial support to the project under the grant agreement no. TAMOP 4.2.1/B-09/1/KMR-2010-0003.

References

[1] P. Adams, D. Bryant, M. Buchanan, Completing partial Latin squares with two filled rows and two filled columns, *Electron. J. Combin.* 15, 1 (2008), R56, 26 pages. ⇒ 99

[2] A. M. Alhakim, A simple combinatorial algorithm for de Bruijn sequences, *Amer. Math. Monthly* 117, 8 (2010) 728–732. ⇒ 99

[3] M.-C. Anisiu, Z. Blázsik, Z. Kása, Maximal complexity of finite words, *Pure Math. Appl.* 13, 1-2 (2002) 39–48. ⇒ 99

[4] M.-C. Anisiu, A. Iványi, Two-dimensional arrays with maximal complexity, *Pure Math. Appl.* (PU.M.A.) 17, 3-4 (2006) 197–204. ⇒ 99

[5] M.-C. Anisiu, Z. Kása, Complexity of words, in *Algorithms of Informatics, Vol. 3* (electronic book, ed. A. Iványi), AnTonCom, Budapest, 2011 (to appear). ⇒ 99

[6] C. Arcos, G. Brookfield, M. Krebs, Mini-Sudokus and groups. *Math. Mag.* 83, 2 (2010) 111–122. ⇒ 99

[7] R. A. Bailey, R. Cameron P. J., Connelly, Sudoku, gerechte designs, resolutions, affine space, spreads, reguli, and Hamming codes, *American Math. Monthly* 115, 5 (2008) 383–404. ⇒ 99

[8] W. U. Behrens, Feldversuchsanordnungen mit verbessertem Ausgleich der Bodenunterschiede, *Zeitschrift für Landwirtschaftliches Versuchs- und Untersuchungswesen* 2 (1956) 176–193. ⇒ 99
[9] E. Bokova, G. Tzaturjan, *Speed of computers with interleaved memory* (in Russian), Master thesis. Moscow State University, Moscow, 1985, 43 pages. ⇒ 105

[10] J. Bond, A. Iványi, Modelling of interconnection networks using de Bruijn graphs, *Third Conference of Program Designers* (ed. A. Iványi), Budapest, July 1–3, 1987, Eötvös Loránd University, Budapest, 1987, pp. 75–88. http://compalg.inf.elte.hu/~tony/Kutatas/Conferences-of-Program-Designers/Volume-4/ ⇒ 99

[11] S. Brett, G. Hurlbert, B. Jackson, Preface [Generalisations of de Bruijn cycles and Gray codes], *Discrete Math.*, 309, 17 (2009) 5255–5258. ⇒ 99

[12] R. Breusch, H. W. Gould, The truncated exponential series. *Amer. Math. Monthly* 75, 9 (1968) 1019–1021. ⇒ 108

[13] H. L. Buchanan, M. N. Ferencak, On completing Latin squares, *J. Combin. Math. Combin. Comput.* 34 (2000) 129–132. ⇒ 99

[14] P. Burcsi, Personal communication. Budapest, March 2009. ⇒ 121

[15] Ch.-Ch. Chang, P.-Y. Lin, Z.-H. Wang, M.-Ch. Li, A sudoku-based secret image sharing scheme with reversibility. *J. Commun.* 5, 1 (2010) 5–12. ⇒ 99

[16] Z. Chen, Heuristic reasoning on graph and game complexity of sudoku, 6 pag. arXiv:0903.1659v1, 2010. ⇒ 99

[17] T. Cirulis, A. Iványi, On the monotony of a "small function“, *Fourth Conference of Program Designers* (ed. A. A. Iványi), Eötvös Loránd University, Budapest, June 1–3, 1988. pp. 171–180. http://compalg.inf.elte.hu/~tony/Kutatas/Conferences-of-Program-Designers/Volume-4/ ⇒ 105

[18] J. Cooper, C. Heitsch, The discrepancy of the lex-least de Bruijn sequence, *Discrete Math.* 310, 6–7 (2010) 1152–1159. ⇒ 99

[19] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms*, Third edition, The MIT Press, 2009. ⇒ 100, 101, 106, 112, 120

[20] J. F. Crook, A pencil-and-paper algorithm for solving Sudoku puzzles, *Notices Amer. Math. Soc.* 56, (2009) 460–468. ⇒ 99
Testing of sequences by simulation

[21] G. Dahl, Permutation matrices related to Sudoku, Linear Algebra Appl. 430 (2009) 2457–2463. ⇒ 99

[22] J. Dénes, A. D. Keedwell, Latin Squares. New Developments in the Theory and Applications, North-Holland, Amsterdam, 1991. ⇒ 99

[23] T. Easton, R. G. Parker, On completing Latin squares, Discrete Appl. Math. 113, 2–3 (2001) 167–181. ⇒ 99

[24] E. P. Egorychev, Integral Representation and the Computation of Combinatorial Sums, American Mathematical Society, Providence, RI, Translations of Mathematical Monographs, 59. ⇒ 105

[25] G. P. Egorychev, A. Iványi, A. I. Makosiy, Analysis of two characterizing the speed of computers with interleaved memory (in Russian), Annales Univ. Sci. Budapest., Sectio Comput. 7 (1987) 19–32. ⇒ 105

[26] C. H. Elzinga, S. Rahmann, H. Wang, Algorithms for subsequence combinatorics, Theor. Comput. Sci. 409, 3 (2008) 394–404. ⇒ 99

[27] C. H. Elzinga, Complexity of categorial time series, Sociological Methods & Research 38, 3 (2010) 463–481. ⇒ 99

[28] M. Erickson, Pearls of discrete mathematics, Discrete Mathematics and its Applications. CRC Press, Boca Raton, 2010. ⇒ 99

[29] R. Euler, On the completability of incomplete Latin squares. European J. Combin. 31 (2010) 535–552. ⇒ 99

[30] S. Ferenczi, Z. Kása, Complexity for finite factors of infinite sequences, Theoret. Comput. Sci. 218, 1 (1999) 177–195. ⇒ 99

[31] A. F. Gabor, G. J. Woeginger, How *not* to solve a Sudoku. Operation Research Letters 38, 6 (2010) 582–584. ⇒ 99

[32] I. Hajirasouliha, H. Jowhari, R. Kumar, R. Sundaram, On completing Latin squares, Lecture Notes in Comput. Sci. 4393 (2007) 524–535, Springer, Berlin, 2007. ⇒ 99

[33] H. Hellerman, Digital Computer System Principles, McGraw Hill, New York, 1967. ⇒ 99, 102
[34] A. Heppes, P. Révész, A new generalization of the concept of Latin squares and orthogonal Latin squares and its application to the design of experiments (in Hungarian), *Magyar Tud. Akad. Mat. Int. Közl.*, 1 (1956) 379–390. ⇒ 99

[35] M. Horváth M., A. Iványi, Growing perfect cubes, *Discrete Math.* 308, 19 (2008) 4378–4388. ⇒ 99

[36] A. Iványi, On the $d$-complexity of words. *Ann. Univ. Sci. Budapest., Sect. Comput.* 8 (1987) 69–90. ⇒ 99

[37] A. Iványi, Construction of infinite de Bruijn arrays, *Discrete Appl. Math.* 22, 3 (1988/89), 289–293. ⇒ 99

[38] A. Iványi, Construction of three-dimensional perfect matrices, (Twelfth British Combinatorial Conference, Norwich, 1989), *Ars Combin.* 29C (1990) 33–40. ⇒ 99

[39] A. Iványi, Perfect arrays, in *Algorithms of Informatics, Vol. 3* (electronic book, ed. A. Iványi), AnTonCom, Budapest, 2011 (to appear). ⇒ 99

[40] A. Iványi, I. Kátai, Estimates for speed of computers with interleaved memory systems, *Annales Univ. Sci. Budapest., Sectio Math.* 19 (1976) 159–164. ⇒ 99, 105, 108

[41] A. Iványi, I. and Kátai, Processing of random sequences with priority. *Acta Cybernet.* 4, 1 (1978/79) 85–101. ⇒ 99

[42] A. Iványi, J. Madarász, Perfect hypercubes. *Electron. Notes Discrete Math.* (submitted). ⇒ 99

[43] A. Iványi, B. Novák, Testing of random sequences by simulation. *Acta Univ. Sapientiae, Inform.* 2, 2 (2010) 135–153. ⇒ 108

[44] A. Iványi, Z. Tóth, Existence of de Bruijn words, *Second Conference on Automata, Languages and Programming Systems* (Salgótarján, 1988), 165–172, DM, 88-4, Karl Marx Univ. Econom., Budapest, 1988. ⇒ 99

[45] I. Kanaana, B. Ravikumar, Row-filled completion problem for Sudoku, *Util. Math.* 81 (2010) 65–84. ⇒ 99

[46] Z. Kása, Computing the $d$-complexity of words by Fibonacci-like sequences. *Studia Univ. Babes-Bolyai Math.* 35, 3 (1990) 49–53. ⇒ 99
[47] Z. Kása, On the $d$-complexity of strings, Pure Math. Appl. 9, 1-2 (1998) 119–128. ⇒99

[48] Z. Kása, On arc-disjoint Hamiltonian cycles in De Bruijn graphs, arXiv 1003.1520 (submitted 7 March 2010). ⇒99

[49] Z. Kása, On scattered subword complexity of strings, Acta Univ. Sapientiae, Inform. 3, 1 (2011) 127–136. ⇒99

[50] A. D. Keedwell, Constructions of complete sets of orthogonal diagonal Sudoku squares, Australas. J. Combin. 47 (2010) 227–238. ⇒99

[51] D. E. Knuth, The Art of Computer Programming. Vol. 1. Fundamental Algorithms (third edition), Addison–Wesley, Upper Saddle River, NJ, 1997. ⇒100, 108, 109

[52] D. E. Knuth, The Art of Computer Programming. Vol. 4A. Combinatorial Algorithms, Addison–Wesley, Upper Saddle River, NJ, 2011. ⇒99

[53] J. S. Kuhl, T. Denley, On a generalization of the Evans conjecture, Discrete Math. 308, 20 (2008) 4763–4767. ⇒99

[54] S. R. Kumar, A. Russell, R. Sundaram, Approximating Latin square extensions, Algorithmica 24, 2 (1999) 128–138. ⇒99

[55] L. Lorch, Mutually orthogonal families of linear Sudoku solutions, J. Aust. Math. Soc. 87, 3 (2009) 409–420. ⇒99

[56] M. Matamala, E. Moreno, Minimum Eulerian circuits and minimum de Bruijn sequences, Discrete Math. 309, 17 (2009) 5298–5304. ⇒99

[57] T. K. Moon, J. H. Gunther, J. J. Kupin, Sinkhorn solves Sudoku, IEEE Trans. Inform. Theory, 55, 4 (2009) 1741–1746. ⇒99

[58] T. Móri, Personal communication, Budapest, March 2011. ⇒121

[59] L.-D. Öhman, A note on completing Latin squares, Australas. J. Combin. 45 (2009) 117–123. ⇒99

[60] R. M. Pedersen, T. L. Vis, Sets of mutually orthogonal Sudoku Latin squares, College Math. J. 40, 3 (2009) 174–180. ⇒99

[61] R. Penne, A note on certain de Bruijn sequences with forbidden subsequences, Discrete Math. 310, 4 (2010) 966–969. ⇒99
A. Iványi, I. Kátai

[62] J. S. Provan, Sudoku: strategy versus structure, *Amer. Math. Monthly* **116**, 8 (2009) 702–707. ⇒ 99

[63] S. Ramanujan, Question 294, *J. Indian Math. Society* **3** (1928) 128–128. ⇒ 101

[64] R. Rowley, B. Bose, *On the number of arc-disjoint Hamiltonian circuits in the De Bruijn graphs*, *Parallel Processing Letters* **3** 4 (1993) 375–382. ⇒ 99

[65] T. Sander, Sudoku graphs are integral, *Electron. J. Combin.* **16**, 1 (2009), N25, 7 pag. ⇒ 99

[66] M. J. Soottile, T. G. Mattson, and C. E. Rasmussen, *Introduction to Concurrency in Programming Languages*. Chapman & Hall/CRC Computational Science Series, CRC Press, Boca Raton, FL, 2010. ⇒ 99

[67] G. Szegő, Über einige von S. Ramanujan gestellte Aufgaben, *J. London Math. Society* **3** (1928) 225–232. See also in *Collected Papers of Gábor Szegő* (ed. by R. Askey), Birkhäuser, Boston, MA, 1982. Volume 2, 141–152. ⇒ 101

[68] O. G. Troyanskaya, O. Arbell, Y. Koren, G. M. Landau, A. Bolshoy, Sequence complexity profiles of prokaryotic genomic sequences: A fast algorithm for calculating linguistic complexity, *Bioinformatics* **18**, 5 (2002) 679–688. ⇒ 99

[69] E. R. Vaughan, The complexity of constructing gerechte designs, *Electron. J. Combin.* **16**, 1 (2009) R15, 8 pag. ⇒ 99

[70] X. Xu, Y. Cao, J.-M. Xu, Y. Wu, Feedback numbers of de Bruijn digraphs, *Comput. Math. Appl.* **59**, 4 (2010) 716–723. ⇒ 99

[71] C. Xu, W. Xu, The model and algorithm to estimate the difficulty levels of Sudoku puzzles. *J. Math. Res.* **11**, 2 (2009) 43–46. ⇒ 99

[72] W. Zhang, S. Liu, H. Huang, An efficient implementation algorithm for generating de Bruijn sequences. *Computer Standards & Interfaces* **31**, 6 (2009) 1190–1191. ⇒ 99

Received: January 11, 2011 • Revised: April 5, 2011