Acta Univ. Sapientiae, Informatica, **3,** 1 (2011) xx–yy

Testing of random matrices

Antal Iványi
Eötvös Loránd University
Department of Computer Algebra
H-1117, Budapest, Hungary
Pázmány sétány 1/C
email: tony@compalg.inf.elte.hu

Imre Kátai
Eötvös Loránd University
Department of Computer Algebra
H-1117, Budapest, Hungary
Pázmány sétány 1/C
email: katai@compalg.inf.elte.hu

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 $\mathcal{M} = [m_{ij}]$ 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, 9, 18, 22, 25, 43, 44, 49, 50], of Sudoku squares [4, 5, 11, 12, 15, 16, 17, 21, 24, 37, 45, 47, **?**, 50, 52, 54, **?**, 55, 57, 59, 61], of de Bruijn arrays [2, 3, 7, 13, 19, 20, 28, 31, 34, 42, 46, 51, 60, 62] and gerechte designs, connected with agricultural and industrial experiments [5, 6, 27] have to contain different elements. The one dimensional special case is also studied is several papers [23, 26, 29, **?**, 30, **?**, 33, 38, 39, 40].

The testing of these matrices raises the following problem.

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

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

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

A realization $\mathcal{M} = [m_{ij}]$ 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, while in Section 5 the pseudocodes of the investigated algorithms are presented.

2 Mathematical background

We start with the first step of the testing of $\mathcal{M}$: describe and analyse several algorithms testing the first row of $\mathcal{M}$. The inputs of these algorithms are n (the length of the first row of $\mathcal{M}$) and the elements of the first row $\mathbf{m} = (m_{11}, m_{12}, \ldots, m_{1n})$. For the simplicity we use the notation $\mathbf{s} = (s_1, s_2, \ldots, s_n)$. 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 $\lg n$ [14], and usually omit the argument n from the functions $\tau(n)$, $\sigma(n)$, $\kappa(n)$, $\kappa_1(n)$, $\kappa_2(n)$, $\gamma(n)$, $\lambda(n)$, $\delta(n)$, $\alpha(n)$, $\mu(n)$, $\eta(n)$, $\phi(n)$, $\rho(n)$, $\beta(n)$, $S_i(n)$, $R_i(n)$, $Q(n)$, $p_k(n)$, $y(n)$, $q_i(k, n)$, $A_i(n)$, $b_j(n)$, $f(n)$, $p(i, j, k, n)$, $c_j(n)$, $c(n)$, 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 Alg by $T_{worst}(n, \text{Alg})$, $T_{best}(n, \text{Alg})$ and $T_{exp}(n, \text{Alg})$ in the worst, best, resp. expected case. The numbers of the corresponding assignments and comparisons are denoted by A, resp. C. The notations O, Ω, Θ, o and ω are used according to [14, pages 43–52] and [41, 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 ([14]) *If* $n \geq 1$ *then*

$$n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n}e \ , \tag{2}$$

where

$$\frac{1}{12n+1} < \tau < \frac{1}{12n}, \tag{3}$$

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:

$$a_k = \frac{n^k}{k!} \quad (k = 0,\ 1,\ 2,\ \ldots), \tag{4}$$

$$S_i = \sum_{k=0}^{n-1} a_k k^i \quad (i = 0,\ 1,\ 2,\ \ldots). \tag{5}$$

If in (5) $k = i = 0$, then $k^i = 0$.

Solving a problem posed by S. Ramanujan [53] Gábor Szegő [56] proved the following connection between e^n and S_0.

Lemma 2 ([56]) *The function* $\sigma(n) = \sigma$, *defined by*

$$\frac{e^n}{2} = S_0 + \left(\frac{1}{3} + \sigma\right) a_n = \sum_{k=0}^{n-1} \frac{n^k}{k!} + \left(\frac{1}{3} + \sigma\right) a_n \quad (n = 1,\ 2,\ \ldots) \tag{6}$$

and

$$\sigma(0) = \frac{1}{6}, \tag{7}$$

tends monotonically decreasing to zero when n *tends to* ∞.

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*

$$S_i = n \sum_{k=0}^{i-1} B(i-1,k) S_k - n^{i-1} a_{n-1} \tag{8}$$

and

$$S_i = \Theta(e^n n^i). \tag{9}$$

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

$$S_i = \sum_{k=0}^{n-1} \frac{n^k}{k!} k^i = n \sum_{k=1}^{n-1} \frac{n^{k-1}}{(k-1)!} k^{i-1} = n \sum_{j=0}^{n-2} \frac{n^j}{j!} (j+1)^{i-1}. \tag{10}$$

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

$$S_i = n \sum_{j=0}^{n-1} \frac{n^j}{j!} (j+1)^{i-1} - n^i a_{n-1}. \tag{11}$$

Now the application of the binomial theorem results (8).

According to (9) $S_0 = \Theta(e^n)$, therefore using induction and (11) we get (9). □

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*

$$S_0 = \frac{e^n}{2} - \frac{n^n}{n!} \left(\frac{1}{3} + \sigma \right), \tag{12}$$

$$S_1 = nS_0 - na_{n-1}, \quad S_2 = S_0(n^2 + n) - 2n^2 a_n, \tag{13}$$

and

$$S_3 = S_0(n^3 + 3n^2 + n) - (3n^3 + 2n^2) a_n. \tag{14}$$

Proof. Expressing S_0 from (6), 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. □

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

$$R_i = \sum_{k=1}^{n} p_k(n) k^i \quad (i = 0,\ 1,\ 2,\ \ldots), \tag{15}$$

where $p_k(n) = p_k$ is the key probability of this paper, defined in [26] as

$$p_k = \frac{n}{n} \frac{n-1}{n} \cdots \frac{n-k+1}{n} \frac{k}{n} = \frac{n!k}{(n-k)!n^{k+1}} \quad (k = 1,\ 2,\ \ldots,\ n). \tag{16}$$

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*

$$R_i = \frac{n!}{n^{n+1}} \sum_{l=0}^{i+1} (-1)^l \binom{i+1}{l} n^{i+1-l} S_l. \tag{17}$$

Proof. Using (15) and (16) the substitution $n - k = j$ results

$$R_i = \sum_{k=1}^{n} \frac{n!k^{i+1}}{(n-k)!n^{k+1}} = \frac{n!}{n^{n+1}} \sum_{j=0}^{n-1} \frac{n^j (n-j)^{i+1}}{j!} \ . \tag{18}$$

From here using the binomial theorem we get (17). □

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

Lemma 6 *If* n *is a positive integer, then*

$$R_0 = 1, \quad R_1 = \frac{n!}{n^n} S_0, \tag{19}$$

and

$$R_2 = 2n - \frac{n!}{n^n} S_0. \tag{20}$$

Proof. $R_0 = 0$ follows from the definition of the probabilities p_k. Substituting $i = 1$ into (17) we get

$$R_1 = \frac{n!}{n^{n+1}} \left(n^2 \sum_{j=0}^{n-1} \frac{n^j}{j!} - 2n \sum_{j=0}^{n-1} \frac{n^j}{j!} j + \sum_{j=0}^{n-1} \frac{n^j}{j!} j^2 \right) . \tag{21}$$

From here using (5) we get

$$R_1 = \frac{n!}{n^{n+1}} (n^2 S_0 - 2n S_1 + S_2), \tag{22}$$

and using (11) the required formula for R_1.

Substituting $i = 2$ into (17) we get

$$R_2 = \frac{n!}{n^{n+1}} \left(n^3 \sum_{j=0}^{n-1} \frac{n^j}{j!} - 3n^2 \sum_{j=0}^{n-1} \frac{n^j}{j!} j + 3n \sum_{j=0}^{n-1} \frac{n^j}{j!} j^2 - \sum_{j=0}^{n-1} \frac{n^j}{j!} j^3 \right) . \tag{23}$$

From here using (5) we have

$$R_2 = \frac{n!}{n^{n+1}} (n^3 S_0 - 3n^2 S_1 + 3n S_2 - S_3), \tag{24}$$

and using (13) and (14) the required formula for R_2. □

The following lemmas give some further properties of R_1 and R_2.

Lemma 7 *If* n *is a positive integer, then*

$$R_1 = \frac{n!}{n^n} S_0 = \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \kappa, \tag{25}$$

where

$$\kappa(n) = \kappa = \sqrt{\frac{\pi n}{2}} \left(e \ \ - 1 - \frac{2\sigma e}{e^n} \right), \tag{26}$$

and κ *tends monotonically decreasing to zero when* n *tends to infinity.*

Proof. Substituting S_0 according to (12) in the formula (20) for R_1 we get

$$R_1 = \frac{n!}{n^n} \left[\frac{e^n}{2} - \frac{n^n}{n!} \left(\frac{1}{3} + \sigma \right) \right] = -\frac{1}{3} + \frac{n!}{n^n} \left(\frac{e^n}{2} - \frac{n^n}{n!} \sigma \right). \tag{27}$$

Substitution of $n!$ according to (2) (Stirling's formula) and writing $1+(e \ \ -1)$ instead of e results

$$R_1 = -\frac{1}{3} + \frac{1}{n^n} \left(\frac{n}{e} \right)^n \sqrt{2\pi n} \, [1 + (e \ \ - 1)] \left[\frac{e^n}{2} - \sigma \right]. \tag{28}$$

The product P of the expressions in the square brackets is

$$P = \frac{e^n}{2} + \frac{e^n}{2} (e \ \ - 1) - \sigma e \ \ , \tag{29}$$

therefore

$$R_1 = \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \frac{\sqrt{2\pi n}}{e^n} \left[\frac{e^n}{2} (e \ \ - 1) - \sigma e \ \ \right], \tag{30}$$

implying

$$R_1 = \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \sqrt{\frac{\pi n}{2}} (e \ \ - 1) - \sqrt{\frac{\pi n}{2}} \frac{2\sigma e \ \ }{e^n} \ . \tag{31}$$

Let

$$\kappa_1(n) = \kappa_1 = \sqrt{\frac{\pi n}{2}} (e \ \ - 1), \quad \kappa_2(n) = \kappa_2 = \sqrt{\frac{\pi n}{2}} \frac{2\sigma e \ \ }{e^n}, \quad \text{and } \kappa = \kappa_1 + \kappa_2, \tag{32}$$

and

$$\gamma(n) = \gamma = \frac{\kappa(n+1)}{\kappa(n)} = \frac{\kappa_1(n+1) - \kappa_2(n+1)}{\kappa_1(n) - \kappa_2(n)} \quad \text{for } n = 1, \ 2, \ \ldots \ . \tag{33}$$

Since all κ functions are positive for all positive integer n's, therefore $\gamma < 1$ for $n \geq 1$ implies the monotonity of κ. 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 (32). Since σ and τ are monotone decreasing functions, and $0 < \sigma(5) < 0.0058$, and $0 < e\ ^{(5)} < 1.02$, and $n^2 < e^n$ for $n \geq 10$, therefore

$$\frac{2\sigma e}{e^n} < \frac{2 \cdot 0.0058 \cdot 1.02}{e^n} < \frac{0.012}{n^2} \text{ for } n \geq 10. \tag{34}$$

Using (33), (34) and the Lagrange remainder of the Taylor series of the function e^x we have

$$\gamma < \frac{\sqrt{n+1}}{\sqrt{n}} \frac{\tau(n+1) + \frac{2(\ \ n+1)}{2}}{\tau(n) + \frac{2(\ \ n)}{2} - \frac{0.012}{n^2}}\ , \tag{35}$$

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

$$\gamma < \frac{\sqrt{n+1}}{\sqrt{n}} \frac{\frac{1}{12(n+1)+1}}{\frac{1}{12n} + \frac{1}{2}\left(\frac{1}{12n}\right)^2 - \frac{0.012}{n^2}}\ . \tag{36}$$

Now multiplication of the denominator and denominator of the right side of (36) by $(12n)^2$ results

$$\gamma = \frac{\sqrt{n+1}}{\sqrt{n}} \frac{\frac{12n \cdot 12n}{12n+13}}{12n + 0.5 - 1.584} = \frac{\sqrt{n+1}}{\sqrt{n}} \frac{12n}{(12n - 1.084)\left(1 + \frac{13}{12n}\right)}\ . \tag{37}$$

Since

$$(12n - 1.084)\left(1 + \frac{13}{12n}\right) > 12n + 10, \tag{38}$$

(37) and (38) imply

$$\gamma < \frac{\sqrt{144n^3 + 144n^2}}{\sqrt{144n^3 + 240n^2}} < 1, \tag{39}$$

finishing the proof of the monotonity of κ. □

We remark, that the monotonity of κ was published in [?] without proof.

Lemma 8 *If n is a positive integer, then*

$$R_2 = 2n - \frac{n!}{n^n} S_0 = 2n + \frac{1}{3} - \sqrt{\frac{\pi n}{2}} e\ \ - \lambda, \tag{40}$$

where

$$\lambda = \sqrt{\frac{\pi n}{2}}(e \quad - 1) + \sigma, \tag{41}$$

and λ *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. □

3 Running times of the algorithms

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

We derive exact formulas for the expected numbers of comparisons $C_{exp}(n,$ $\textsc{Linear}) = C_L$, $C_{exp}(n, \textsc{Backward}) = C_W$, and $C_{exp}(n, \textsc{Bucket}) = C_B$, further for the expected running times $T_{exp}(n, \textsc{Linear}) = T_L$, $T_{exp}(n, \textsc{Backward}) = T_W$, and $T_{exp}(n, \textsc{Bucket}) = T_B$.

3.1 Running time of algorithm Linear

Linear writes zero into the elements of an n length vector $\mathbf{v} = (v_1,\ v_2,\ \ldots,\ v_n)$, then investigates the elements of the realization $\mathbf{s}$ and if $v[s_k] > 0$ (signalising a repetition), then returns False, otherwise adds 1 to v_k. If Linear does not find a repetition among the elements of $\mathbf{s}$ then it returns finally True.

Linear needs assignments in lines 01, 03, and 08, and it needs comparisons in line 05. The number of assignments in lines 01 and 03 equals to $n + 1$ for arbitrary input and varies between 1 and n in line 08. The number of comparisons in line 08 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* $C_{exp}(n, \textsc{Linear}) = C_L$ *of* Linear *is*

$$C_L = 1 - \frac{n!}{n^n} + R_1 = \sqrt{\frac{\pi n}{2}} + \frac{2}{3} + \kappa - \frac{n!}{n^n}\ . \tag{42}$$

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

$$y(n) = y = \max\{k : 1 \le k \le n \text{ and } s_1,\ s_2,\ \ldots,\ s_k \text{ are different}\} \tag{43}$$

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

$$\Pr\{y = k\} = p_k \quad (k = 1,\ 2,\ \ldots,\ n), \tag{44}$$

where p_k is the probability introduced in (16).

If $y = k$ and $1 \le k \le n-1$, then Linear executes $k+1$ comparisons, and only n comparisons, if $y = n$, therefore

$$C_L = \sum_{k=1}^{n-1} p_k(k+1) + p_n n = \sum_{k=1}^{n} p_k(k+1) - p_n = 1 - \frac{n!}{n^n} + \sum_{k=1}^{n} p_k k, \tag{45}$$

from where using Lemma (7) we receive

$$C_L = 1 - \frac{n!}{n^n} + R_1 = \sqrt{\frac{\pi n}{2}} + \frac{2}{3} - \frac{n!}{n^n} + \kappa\ . \tag{46}$$

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*

$$T_L = n + \sqrt{2\pi n} + \frac{7}{3} + 2\kappa - 2\frac{n!}{n^n}, \tag{47}$$

where κ *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 08 is the same as C_L. Therefore

$$T_L = n + 1 + 2C_L. \tag{48}$$

Substitution of (46) into (48) results the required (42). □

We remark, that (46) is equivalent with

$$C_L = 1 - \frac{n!}{n^n} + 1 + \frac{n-1}{n} + \frac{n-1}{n}\frac{n-2}{n} + \cdots + \frac{n-1}{n}\frac{n-2}{n}\cdots\frac{1}{n}, \tag{49}$$

n	C_L	$\sqrt{\frac{n}{2}} + \frac{2}{3}$	$\frac{n!}{n^n}$	κ	$\delta = \kappa - \frac{n!}{n^n}$	σ
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	3.469162	0.038400	0.041238	+0.002838	0.005799
6	3.759259	3.736647	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	0.002403	0.033444	+0.031040	0.003656
9	4.457379	4.426609	0.000937	0.031707	+0.030770	0.003255
10	4.659853	4.629994	0.000363	0.030222	+0.029859	0.002933

Table 1: Values of C_L, $\sqrt{\pi n/2}+2/3$, $n!/n^n$, κ, δ, and σ for $n = 1,\ 2,\ \ldots,\ 10$

demonstrating the close connection with the function

$$Q(n) = Q = C_L - 1 + \frac{n!}{n^n}, \tag{50}$$

studied by several authors, e.g. in [8, ?, 41].

Table 1 shows the concrete values of the functions appearing in the analysis of C_L and T_L for $1 \le n \le 10$, where C_L was calculated using (46), κ using (16), and σ using (6) (data in this and further tables are taken from [?]). 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 $n > 8$

$$\frac{n!}{n^n} = \left(\frac{n}{e}\right)^n \sqrt{2\pi n}\frac{e}{n^n} < \frac{\sqrt{2\pi n}}{e^n}e^{1/(12n)} < \frac{2.7\sqrt{n}}{e^n} < \frac{0.012}{n^2} \tag{51}$$

holds, we can prove – using the same arguments as in the proof of Lemma 7 – the following assertion.

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

$$T_L = n + \sqrt{2\pi n} + \frac{7}{3} + \delta, \tag{52}$$

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

$$\delta(n+1) > \delta(n) \textit{ for } 1 \le n \le 7 \textit{ and } \delta(n+1) < \delta(n) \textit{ for } n \ge 8. \tag{53}$$

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 (42) and (50) we get

$$\kappa = C_L - \sqrt{\frac{\pi n}{2}} - \frac{2}{3} - \frac{n!}{n^n} = Q - \sqrt{\frac{\pi n}{2}} + \frac{1}{3}. \tag{54}$$

Substituting the power series

$$Q = \sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \frac{1}{12}\frac{\pi}{2n} - \frac{14}{135n} + \frac{1}{288}\frac{\pi}{2n^3} + O(n^{-2}) \tag{55}$$

cited by D. E. Knuth [41, Equation (25) on page 120] into (54) and using

$$\frac{1}{n^{k/2}} - \frac{1}{(n+1)^{k/2}} = \Theta\left(\frac{1}{n^{1+k/2}}\right) \tag{56}$$

for $k = 1,\ 2,\ 3$ and 4 we get

$$\kappa(n) - \kappa(n+1) = \frac{\sqrt{\pi}}{12\sqrt{2}}\left(\frac{1}{\sqrt{n}} - \frac{1}{\sqrt{n+1}}\right) + O(n^{-2}), \tag{57}$$

implying

$$\kappa(n) - \kappa(n+1) = \frac{\sqrt{\pi}}{12\sqrt{2}}\frac{1}{\sqrt{n}\sqrt{n+1}(\sqrt{n}+\sqrt{n+1})} + O(n^{-2}), \tag{58}$$

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 $C_{best}(n, \text{Backward}) = 1 = \Theta(1)$, and $C_{worst}(n, \text{Backward}) = B(n, 2) = \Theta(n^2)$. The number of assignments is 1 in the best case (in line 01) and is 2 in the worst case (in lines 01 and in line 05). The expected number of assignments is $A_{exp}(n, \text{Backward}) = 1 + \frac{n!}{n^n}$, since only the good realizations require the second assignment.

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*

$$C_W = n - \sqrt{\frac{\pi n}{8}} + \frac{2}{3} - \frac{1}{2}\kappa - \frac{n!}{n^n}\frac{n+1}{2} = \sqrt{\frac{\pi n}{8}} + \frac{2}{3} - \alpha, \tag{59}$$

where $\alpha(n) = \alpha = \frac{}{2} + \frac{n!}{n^n}\frac{n+1}{2}$ *monotonically decreasing tends to zero when* n *tends to* ∞*.*

Proof. Let y be as defined in (43), p_k as defined in (16), and let

$$z = \{q : 1 \leq q \leq k \text{ and } s_1, s_2, \ldots, s_k \text{ are different and } s_{k+1} = s_q \mid y = k\} \tag{60}$$

be a random variable characterising the index of the first repeated element of **s**.

Let

$$q_i(k, n) = q_i(k) = \Pr\{z = i | y = k\} \quad (k = 1,\ 2,\ \ldots,\ n;\ i = 1,\ 2,\ \ldots k). \tag{61}$$

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

$$C_W = \sum_{k=1}^{n-1} p_k \left(B(k, 2) + \sum_{i=1}^{k} i q_i(k) \right) + p_n B(n, 2), \tag{62}$$

where

$$q_i(k, n) = q_i(k) = \frac{1}{k} \quad (i = 1,\ 2,\ \ldots,\ k;\ k = 1,\ 2, \ldots,\ n). \tag{63}$$

Adding a new member to the first sum we get

$$C_W = \sum_{k=1}^{n} p_k \left(B(k, 2) + \sum_{i=1}^{k} q_i(k) i \right) - p_n \sum_{i=1}^{n} q_i(k) i. \tag{64}$$

Using the uniform distribution (63) of z we can determine its contribution to C_W:

$$\sum_{i=1}^{k} q_i(k) i = \sum_{i=1}^{k} \frac{i}{k} = \frac{k+1}{2}. \tag{65}$$

Substituting the contribution in (65) into (64), and taking into account Lemma 6 and Lemma 7 we have

$$C_W = \frac{1}{2}R_2 - \frac{1}{2}R_0 - \frac{n!}{n^n}\frac{n+1}{2}. \tag{66}$$

Now Lemma 6 and Lemma 7 result

$$C_W = n - \sqrt{\frac{\pi n}{8}} + \frac{2}{3} - \frac{1}{2}\kappa - \frac{n!}{n^n}\frac{n+1}{2}. \tag{67}$$

The known decreasing monotonity of κ and $\frac{n!}{n^n}$ imply the decreasing monotonity of α. □

Theorem 13 *The expected running time* $T_{exp}(n, \textsc{Backward}) = T_W$ *of the algorithm* Backward *is*

$$T_W = n - \sqrt{\frac{\pi n}{8}} + \frac{5}{3} - \alpha, \tag{68}$$

where $\alpha = \frac{}{2} + \frac{n!}{n^n}\frac{n+1}{2}$ *tends monotonically decreasing to zero when* n *tends to* ∞.

Proof. Taking into account (67) and $A_{exp}(n, \textsc{Backward}) = 1 + \frac{n!}{n^n} - \frac{n!}{n^n}\frac{n+1}{2}$ we get (68). □

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 κ 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 $I_1,\ I_2, \ldots,\ I_m$, where $I_j = [(j-1)m + 1, jm]$ for $j = 1,\ 2,\ \ldots\ j$, and sequentially puts the elements of $\mathbf{s}$ into the bucket B_j (we use the word bucket due to some similarity to bucket sort [?]): if $\lceil s_i/m \rceil = j$, then s_i belongs to B_j). Bucket works until the first repetition (stopping with $g = \textsc{False}$, or up to the processing of the last element s_n (stopping with $g = \textsc{True}$).

Bucket handles an array $Q[1 : m, 1 : m]$ (where $m = \lceil \sqrt{n} \rceil$ and puts the element s_i into the rth row of Q, and it tests using linear search whether s_j appeared earlier in the corresponding bucket. The elements of the vector $\mathbf{c} = (c_1, c_2, \ldots, c_m)$ are counters, where c_j $(1 \leq j \leq m))$ shows the actual number of elements in B_j.

n	C_W	$n - \sqrt{\frac{n}{8}} + \frac{2}{3}$	$\frac{n!}{n^n}\frac{n+1}{2}$	κ	$\alpha = \frac{}{2} + \frac{n!}{n^n}\frac{n+1}{2}$
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

Table 2: Values of C_W, $n - \sqrt{\pi n/8} + 2/3$, $\frac{n!}{n^n}\frac{n+1}{2}$, κ, and $\alpha = \frac{}{2} + \frac{n!}{n^n}\frac{n+1}{2}$ for $n = 1,\ 2,\ \dots,\ 10$

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 08, m assignments in line 04, and 1 assignment in line 01, 1 in line 02, 2 in line 06, and 1 in line 08, 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 08, further $3m$ assignments in lines 06, and 12, totally $\frac{m^2(m-1)}{2} + 3m^2$ operations. Lines 01, 02, and 09 require 1 assignment per line, and the assignment in line 04 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* $b_j(n) = b_j$ $(j = 1,\ 2,\ \dots,\ m)$ *be a random variable characterising the number of elements in the bucket* B_j *at the moment of the first repetition. Then*

$$E\{b_j\} = \sqrt{\frac{\pi}{2}} - \mu \quad \textit{for } j = 1,\ 2,\ \dots,\ m, \tag{69}$$

where

$$\mu(n) = \mu = \frac{1}{3\sqrt{n}} - \frac{\kappa}{\sqrt{n}}, \tag{70}$$

and μ *tends monotonically decreasing to zero when* n *tends to infinity.*

Proof. Due to the symmetry of the buckets it is sufficient to prove (69) and (70) for $j = 1$.

Let m be a positive integer and $n = m^2$. Let y be the random variable defined in (41) and p_k be the probability defined in (16).

Let $A_i(n) = A_i$ $(i = 1,\ 2,\ \dots,\ n)$ be the event that the number i appears in $\mathbf{s}$ 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

$$E\{b_1\} = \sum_{i=1}^{m} Y_i = \sum_{i=1}^{m} \Pr\{A_i\} = m\Pr\{A_1\} \tag{71}$$

and

$$\Pr\{A_1\} = \Pr\{1 \in \{s_1,\ s_2,\ \dots,\ s_k\} | y = k\} = \sum_{k=1}^{n} p_k \frac{k}{n} = \frac{1}{n} \sum_{k=1}^{n} p_k k = \frac{1}{n} R_1 . \tag{72}$$

Using Lemma 7 we get

$$E\{b_1\} = m\frac{1}{n}R_1 = \frac{m}{n}\left(\sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \kappa\right), \tag{73}$$

resulting (69) and (70).

We omit the proof of the monotonity of μ, since it is similar to the corresponding part in the proof of Lemma 7. □

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*

$$E\{f\} = 1 + \sqrt{\frac{\pi}{8}} - \eta, \tag{74}$$

where

$$\eta(n) = \eta = \frac{\frac{1}{6} + \sqrt{\frac{}{8}} - \frac{}{2}}{\sqrt{n} + 1}, \tag{75}$$

and η *tends monotonically decreasing to zero when* n *tends to infinity.*

Proof. Let $p(i, j, k, n) = p(i, j, k)$ be the probability of the event that there are k different elements before the first repetition, and the repeated element

n	$E\{b_1\}$	$\sqrt{\frac{}{2}}$	$\frac{1}{3\sqrt{n}}$	$\frac{}{\sqrt{n}}$	$\mu = \frac{1}{3\sqrt{n}} - \frac{}{\sqrt{n}}$
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	0.120551
7	1.147287	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

Table 3: Values of $E\{b_1\}$, $\sqrt{\frac{}{2}}$, $\frac{1}{3\sqrt{n}}$, $\frac{}{\sqrt{n}}$, and $\mu = \frac{1}{3\sqrt{n}} - \frac{}{\sqrt{n}}$ for $n = 1,\ 2,\ \ldots,\ 10$

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

$$p(i,j,k,n) = \binom{m}{i}\binom{n-m}{k-i}k!\frac{i}{n^{k+1}}\,, \tag{76}$$

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

$$E\{f\} = \sum_{i,j,k,n} p(i,j,k)\frac{i+1}{2} - mp_n \tag{77}$$

$$= \frac{m}{2n}\sum_{k=1}^{n}\frac{k!}{n^k}\sum_{i=1}^{k}\binom{m}{i}\binom{n-m}{k-i}i(i+1) - p_n\frac{n+1}{2} \tag{78}$$

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

$$E'\{f\} = E\{f\} + p_n\frac{n+1}{2}. \tag{79}$$

Then dividing the inner sum in (78) by $\binom{n}{k}$ we get the expected value of the random variable $\xi(\xi+1)$, where ξ has hypergeometric distribution with parameters n, m, and k. It is easy to compute that

$$\mathrm{E'}\{\xi(\xi{+}1)\} = \mathrm{E'}\{\xi\}(\mathrm{E'}\{\xi{+}1\}) + \mathrm{Var}\{\xi\} = \frac{km[k(m-1)+(2n-1-m)]}{n(n-1)}, \tag{80}$$

therefore

$$\mathrm{E'}\{f\} = \frac{m}{2n}\sum_{k=1}^{n}\frac{k!}{n^k}\binom{n}{k}\frac{km[k(m-1)+(2n-1-m)]}{n(n-1)} \tag{81}$$

$$= \frac{1}{2(n-1)}\sum_{k=1}^{n} p_k[k(m-1)+(2n-1-m)] \tag{82}$$

$$= \frac{m-1}{2(n-1)}R_1 + \frac{2n-1-m}{2(n-1)} = \frac{2m+1+R_1}{2m+2} \tag{83}$$

$$= 1 + \sqrt{\frac{\pi}{8}} - \frac{\frac{1}{6} + \sqrt{\frac{}{8}} - \frac{}{2}}{\sqrt{n}+1}. \tag{84}$$

The convergence and monotonicity of η is the consequence of the properties of κ. Taking into account the small value of p_n (see 16) the difference $E'\{f\} - E\{f\}$ has negligible influence on the limit of $E\{f\}$. □

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

$$C_B = \sqrt{n} + \frac{1}{3} - \sqrt{\frac{\pi}{8}} + \rho, \tag{85}$$

where

$$\rho(n) = \rho = \frac{\frac{5}{6} - \sqrt{\frac{9}{8}} - \frac{3}{2}}{\sqrt{n}+1}. \tag{86}$$

and ρ *tends monotonically decreasing to zero when* n *tends to infinity.*

Proof. Let $\mathbf{s} = (s_1,\ s_2,\ \ldots,\ s_n)$ be the input sequence of the algorithm BUCKET. 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 (43), and p_k the probability defined in (16). 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

$$C_B = \mathrm{E}\left\{\sum_{j=1}^{m} c_j\right\} + \mathrm{E}\{f\} = m\mathrm{E}\{c_1\} + \mathrm{E}\{f\}. \tag{87}$$

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

$$\Pr\{A(i_1, i_2, k) | y = k \text{ and } 2 \le k \le n\} = \frac{\binom{n-2}{k-2}}{\binom{n}{k}} = \frac{k(k-1)}{n(n-1)}\ , \tag{88}$$

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

$$\mathrm{E}\{c_1\} = \sum_{k=1}^{n} p_k \frac{k(k-1)}{n(n-1)} \binom{m}{2} = \frac{m(m-1)}{2n(n-1)} \left(\sum_{k=1}^{n} p_k k^2 - \sum_{k=1}^{n} p_k k\right), \tag{89}$$

from where using Lemma 7 and Lemma 8 we get

$$\mathrm{E}\{c_1\} = \frac{n - \sqrt{n}}{2n^2 - 2n}(R_2 - R_1) = \frac{1}{2n + 2\sqrt{n}} \left[2n - 2\left(\sqrt{\frac{\pi n}{2}} - \frac{1}{3} + \kappa\right)\right]. \tag{90}$$

This equality implies

$$\mathrm{E}\{c_1\} = 1 - \frac{1}{\sqrt{n} + 1} \left(\sqrt{\frac{\pi}{8}} + \frac{2}{3} - \kappa\right). \tag{91}$$

From (87) taking into account (91), (81), and (84) we get

$$C_B = \sqrt{n} + \frac{1}{3} - \sqrt{\frac{\pi}{8}} + \frac{\sqrt{\frac{9}{8}} + \frac{5}{6} - \frac{3}{2}}{\sqrt{n} + 1}\ . \tag{92}$$

Denoting the last fraction by ρ we get the required (85). The monotonity of ρ is the consequence of the monotonity of κ. □

Theorem 17 *The expected running time* $T_{\text{exp}}(n, \text{BUCKET}) = T_B$ *of* BUCKET *is*

$$T_B = \sqrt{n}\left(3 + 3\sqrt{\frac{\pi}{2}}\right) + \sqrt{\frac{25\pi}{8}} + \phi, \tag{93}$$

where

$$\phi(n) = \phi = 3\kappa - \rho - 3\eta - \frac{n!}{n^n} - \frac{3\sqrt{\frac{}{8}} - \frac{1}{3} - \frac{3}{2}}{\sqrt{n}+1}, \tag{94}$$

and ϕ *tends to zero when* n *tends to infinity.*

Proof. BUCKET requires 2 assignments in lines 01 and 02, $\sqrt{n}$ assignments in line 04, R_1 assignments in line 06, $C_B + E\{f\}$ assignments in line 08, $1 - p_n$ expected assignment in line 09 and $2R_1$ assignments in lines 11 and 12 before the first repeated element, and $2E\{f\} - 1$ assignments after the first repeated element.

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

$$A_B = 2 + \sqrt{n} + 3R_1 + C_B + 3E\{f\} - \frac{n!}{n^n}\ . \tag{95}$$

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

$$A_B = 2\sqrt{n} + \frac{13}{3} + 3\sqrt{\frac{\pi n}{2}} + 3\kappa - \sqrt{\frac{\pi}{8}} + \rho + 3\sqrt{\frac{\pi}{8}} - 3\eta - \frac{n!}{n^n}\ , \tag{96}$$

implying

$$A_B = \sqrt{n}\left(2 + 3\sqrt{\frac{\pi}{2}}\right) + \frac{13}{3} + \sqrt{\frac{\pi}{2}} + 3\kappa + \rho - 3\eta - \frac{n!}{n^n}\ . \tag{97}$$

Summing up the expected number of comparisons in (85) and of assignments in (96) we get the final formula (93). □

3.4 Test of random arrays

Now let $A = [1 : n, 1 : n]$ be a two-dimensional random array. The array A is called *good,* if its all lines (rows and columns) contain a permutation of the elements $1,\ 2,\ \ldots,\ n$.

Theorem 18 *The expected running time* $T_{\text{exp}}(n, \text{MATRIX}) = T_M$ *of* MATRIX *is*

$$T_M = T_B + o(1). \tag{98}$$

Proof. According to Theorem 17 we have

$$T_B = \sqrt{n}\left(3 + 3\sqrt{\frac{\pi}{2}}\right) + \sqrt{\frac{25\pi}{8}} + o(1). \tag{99}$$

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

$$\Pr\{G_k\} = \left(\frac{n!}{n^n}\right)^k, \tag{100}$$

so for the expected time $T_{exp}(n, \text{Matrix}) = T_R$ of the testing of the rows we have

$$T_R \le T_B + T_B \sum_{k=1}^{n-1} \left(\frac{n!}{n^n}\right)^k = T_B + o(1). \tag{101}$$

Since the columns are also independent, all the rows and the first k columns are good with the probability

$$p = \left(\frac{n!}{n^n}\right)^{n+k}, \tag{102}$$

and so for the expected time of testing of the columns $T_{exp}(n, \text{Matrix}) = T_C$ holds

$$T_C \le T_B \sum_{k=0}^{n-1} \left(\frac{n!}{n^n}\right) = o(1), \tag{103}$$

and so

$$T_M = T_R + T_C \tag{104}$$

implies (98). □

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 [?]. 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.

Index and Algorithm	$T_{best}(n)$	$T_{worst}(n)$	$T_{exp}(n)$
1. Linear	$\Theta(1)$	$\Theta(n)$	$\Theta(\sqrt{n})$
2. Backward	$\Theta(1)$	$\Theta(n^2)$	$\Theta(n)$
3. Bucket	$\Theta(1)$	$\Theta(n\sqrt{n})$	$\Theta(\sqrt{n})$
4. Matrix	$\Theta(1)$	$\Theta(n\sqrt{n})$	$\Theta(\sqrt{n})$

Table 4: The expected number of comparisons of the investigated algorithms in best, worst and expected cases

Index and Algorithm	$T_{best}(n)$	$T_{worst}(n)$	$T_{exp}(n)$
1. Linear	$\Theta(n)$	$\Theta(n)$	$n + \Theta(\sqrt{n})$
2. Backward	$\Theta(1)$	$\Theta(n^2)$	$\Theta(n)$
3. Bucket	$\Theta(\sqrt{n})$	$\Theta(n\sqrt{n})$	$\Theta(\sqrt{n})$
8. Matrix	$\Theta(1)$	$\Theta(n\sqrt{n})$	$\Theta(\sqrt{n})$

Table 5: The running times of the investigated algorithms in best, worst and expected cases

5 Pseudocodes of the algorithms

The inputs of the following seven algorithms are n (the length of the sequence $\mathbf{s}$) and $\mathbf{s} = (s_1, s_2, \ldots, s_n)$, 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 True, if the input sequence is good, and False otherwise).

The working variables are usually the cycle variables i and j.

We use the pseudocode defined in [**?**].

5.1 Definition of algorithm Linear

Linear writes zero into the elements of an n length vector $\mathbf{v} = (v_1,\ v_2,\ \ldots,\ v_n)$, then investigates the elements of the realization and if $v[s_i] > 0$ (signalising a repetition), then stops, otherwise adds 1 to v_k.

Linear$(n, \mathbf{s})$

```
$g \leftarrow$ True
for $i \leftarrow 1$ to $n$
     do $v[i] \leftarrow 0$
for $i \leftarrow 1$ to $n$
```

```
    do if $v[s[i]] > 0$
          then $g \leftarrow$ FALSE
               return $g$
          else $v[s[i]] \leftarrow v[s[i]] + 1$
return $g$
```

5.2 Definition of algorithm Backward

Backward compares the second (i_2), third (i_3), ..., last (i_n) element of the realization **s** with the previous elements until the first collision or until the last pair of elements.

Backward$(n, \mathbf{s})$

```
g ← TRUE
for $i \leftarrow 2$ to $n$
    do for $j \leftarrow i - 1$ downto $1$
           do if $s[i] = s[j]$
                 then $g \leftarrow$ FALSE
                      return $g$
return $g$
```

5.3 Definition of algorithm Bucket

Bucket handles the array $Q[1 : m, 1 : m]$ (where $m = \lceil\sqrt{n}\rceil$) and puts the element s_i into the jth row of Q, where $j = \lceil s_i/m \rceil$ and it tests using linear search whether s_i appeared earlier in the corresponding row. The elements of the vector $\mathbf{c} = (c_1, c_2, \ldots, c_m)$ are counters, where c_j $(1 \leq j \leq m)$ shows the number of elements of the ith row.

For the simplicity we suppose that n is a square.

Bucket$(n, \mathbf{s})$

```
g ← TRUE
$m \leftarrow \sqrt{n}$
for $j \leftarrow 1$ to $m$
    do $c[j] \leftarrow 1$
for $i \leftarrow 1$ to $n$
    do $r \leftarrow \lceil s[i]/m \rceil$
       for $j \leftarrow 1$ to $c[r] - 1$
           do if $s[i] = Q[r, j]$
                 then $g \leftarrow$ FALSE
                      return $g$
```

```
                  else $Q[r, c[r]] \leftarrow s[i]$
                       $c[r] \leftarrow c[r] + 1$
return $g$
```

5.4 Definition of algorithm Matrix

Matrix is based on Bucket.

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

Let $\mathcal{M}$ be an $n \times n$ sized matrix, where $m_{ij} \in \{1, 2, \ldots, n\}$. The ith row of $\mathcal{M}$ is denoted by r_i, and the jth column by c_j for $1 \leq i, j \leq n$.

```
Matrix($n, \mathcal{M}$)

$g \leftarrow$ True
Bucket($n, r_1$)
if $g =$ False
  then return $g$
for $i \leftarrow 2$ to $n$
     do Bucket($n, r_i$)
        if $g =$ False
          then return $g$
for $j \leftarrow 1$ to $n$
     do Bucket($n, c_j$)
        if $g =$ False
          then return $g$
return $g$
```

Acknowledgements. Authors thank Tamás F. Móri [48] for proving Lemmas 14 and 15, and Péter Burcsi [10] for useful information on references (both are teachers of Eötvös Loránd University).

The European Union and the European Social Fund have provided financial support to the project under the grant agreement no. TÁMOP 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), Research paper 56, 26 pages. ⇒1

[2] A. M. Alhakim, A simple combinatorial algorithm for de Bruijn sequences. *Amer. Math. Monthly* **117,** (8) (2010) 728–732, ⇒1

[3] M.-C. Anisiu, A. Iványi, Two-dimensional arrays with maximal complexity. *Pure Math. Appl.* (*PU.M.A.*) **17,** (3–4) (2006) 197–204. ⇒1

[4] C. Arcos, G. Brookfield, M. Krebs, Mini-Sudokus and groups. *Math. Mag.* **83,** (2) (2010) 111–122. ⇒1

[5] 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. ⇒1

[6] W. U. Behrens, Feldversuchsanordnungen mit verbessertem Ausgleich der Bodenunterschiede, *Zeitschrift für Landwirtschaftliches Versuchs- und Untersuchungswesen,* **2** (1956) 176–193. ⇒1

[7] S. Brett, G. Hurlbert, B. Jackson, Preface [Generalisations of de Bruijn cycles and Gray codes]. *Discrete Math.*, **309,** (17) (2009) 5255–5258. ⇒1

[8] R. Breusch, H. W. Gould, The truncated exponential series. *American Math. Monthly* **75,** (9) (1968) 1019–1021. ⇒10

[9] H. L. Buchanan, M. N. Ferencak, On completing Latin squares. *J. Combin. Math. Combin. Comput.*, **34** (2000) 129–132. ⇒1

[10] P. Burcsi, Personal communication. Budapest, March 2009. ⇒23

[11] 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. io-port 05765623 ⇒1

[12] Z. Chen, Heuristic reasoning on graph and game complexity of sudoku. 6 pages. ARXIV, 2010. ⇒1

[13] J. Cooper, C. Heitsch, The discrepancy of the lex-least de Bruijn sequence. *Discrete Math.,* **310,** (6–7) (2010) 1152–1159. ⇒1

[14] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms.* Third edition. The MIT Press, 2009. ⇒2, 3

[15] J. F. Crook, A pencil-and-paper algorithm for solving Sudoku puzzles. *Notices Amer. Math. Soc.,* **56,** (2009) 460–468. ⇒1

[16] G. Dahl, Permutation matrices related to Sudoku. *Linear Algebra Appl.,* **430** (2009) 2457–2463. ⇒1

[17] J. Dénes, A. D. Keedwell, *Latin Squares. New Developments in the Theory and Applications.* North-Holland, Amsterdam, 1991. ⇒1

[18] T. Easton, R. G. Parker, On completing Latin squares. *Discrete Appl. Math.*, **113,** (2–3) (2001) 167–181. ⇒1

[19] C. H. Elzinga, S. Rahmann, H. Wung, Algorithms for subsequence combinatorics, *Theor. Comput. Sci.*, **409,** (3) (2008) 394–404. ⇒1

[20] C. H. Elzinga, Complexity of categorial time series, *Sociological Methods & Research,* **38,** (3) (2010) 463–481. ⇒1

[21] M. Erickson, *Pearls of Discrete Mathematics.* Discrete Mathematics and its Applications. CRC Press, Boca Raton, 2010. Zbl 1182.05001. ⇒1

[22] R. Euler, On the completability of incomplete Latin squares. *European J. Combin.* **31** (2010) 535–552. ⇒1

[23] S. Ferenczi, Z. Kása, Complexity for finite factors of infinite sequences. *WORDS* (Rouen, 1997). *Theoret. Comput. Sci.* **218,** (1) (1999) 177–195. ⇒1

[24] A. F. Gabor, G. J. Woeginger, How *not* to solve a Sudoku. *Operation Research Letters* (accepted). ⇒1

[25] I. Hajirasouliha, H. Jowhari, R. Kumar, R. Sundaram, On completing Latin squares. *Lecture Notes in Comput. Sci.,* **4393** (STACS2007), 524–535. Springer, Berlin, 2007. MR2362490 (2008k:68122). ⇒1

[26] H. Hellerman, *Digital Computer System Principles.* Mc Graw Hill, New York, 1967. ⇒1, 4

[27] 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. ⇒1

[28] M. Horváth M., A. Iványi, Growing perfect cubes. *Discrete Math.,* **308,** (19) (2008) 4378–4388. MR2433864 (2009d:05036) ⇒1

[29] A. Iványi, On the **d**-complexity of words. *Ann. Univ. Sci. Budapest., Sect. Comput.* **8** (1987) 69–90 (1988). MR0974183 (90d:68063), http://compalg.inf.elte.hu/~tony/Kutatas/PerfectArrays/On-the-d-complexity.pdf. ⇒1

[30] A. Iványi, Construction of infinite de Bruijn arrays. *Discrete Appl. Math.* **22,** (3) (1988/89), 289–293. ⇒1

[31] A. Iványi, Construction of three-dimensional perfect matrices. (Twelfth British Combinatorial Conference, Norwich, 1989). *Ars Combin.* **29C** (1990) 33–40. ⇒1

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

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

[34] A. Iványi, J. Madarász, Perfect hypercubes. *Electron. Notes Discrete Math.* (submitted). ⇒1

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

[36] 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. ⇒

[37] I. Kanaana, B. Ravikumar, Row-filled completion problem for Sudoku. *Util. Math.* **81** (2010) 65–84. ⇒1

[38] Z. Kása, Computing the **d**-complexity of words by Fibonacci-like sequences. *Studia Univ. Babes-Bolyai Math.* **35,** (3) (1990) 49–53. ⇒1

[39] Z. Kása, On the **d**-complexity of strings. In *Modern applied analysis* (Ilieni, 1997). *Pure Math. Appl.* **9,** (1–2) (1998) 119–128. MR1677568 (2000c:68112) ⇒1

[40] Z. Kása, On scattered subword complexity of strings. *Acta Univ. Sapientiae, Informatica* **3,** (1) (2011) 111–120. ⇒1

[41] D. E. Knuth, *The Art of Computer Programming. Vol. 1. Fundamental Algorithms* (third edition). Addison–Wesley, 1997. ⇒2, 10, 11

[42] D. E. Knuth, *The Art of Computer Programming. Vol. 4A. Combinatorial Algorithms.* Addison–Wesley, 2011. ⇒1

[43] J. S. Kuhl, T. Denley, On a generalization of the Evans conjecture. *Discrete Math.* **308,** (20) (2008) 4763–4767. ⇒1

[44] S. R. Kumar S., A. Russell, R. Sundaram, Approximating Latin square extensions. *Algorithmica* **24,** (2) (1999) 128–138. ⇒1

[45] L. Lorch, Mutually orthogonal families of linear Sudoku solutions. *J. Aust. Math. Soc.,* **87,** (3) (2009) 409–420. ⇒1

[46] M. Matamala, F. Moreno, Minimum Eulerian circuits and minimum de Bruijn sequences. *Discrete Math.,* **309,** (17) (2009) 5298–5304. ⇒1

[47] T. K. Moon, J. H. Gunther, J. J. Kupin, Sinkhorn solves Sudoku. *IEEE Trans. Inform. Theory,* **55,** (4) (2009) 1741–1746. ⇒1

[48] T. Móri, Personal communication. Budapest, March 2011. ⇒23

[49] L.-D. Öhman, A note on completing Latin squares. *Australas. J. Combin.,* **45** (2009) 117–123. ⇒1

[50] R. M. Pedersen, T. L. Vis, Sets of mutually orthogonal Sudoku Latin squares. *College Math. J.,* **40,** (3) (2009) 174–180. ⇒1

[51] R. Penne, A note on certain de Bruijn sequences with forbidden subsequences. *Discrete Math.*, **310,** (4) (2010) 966–969. ⇒1

[52] J. S. Provan, Sudoku: strategy versus structure. *Amer. Math. Monthly*, **116,** (8) (2009) 702–707. ⇒1

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

[54] T. Sander, Sudoku graphs are integral. *Electron. J. Combin.,* **16,** (1) (2009), Note 25, 7 pages. ⇒1

[55] 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, 2010. ⇒1

[56] G. Szegő G., 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, 1982. Volume 2, 141–152. ⇒3

[57] D. Thom, SUDOKU ist NP-vollständig. PhD Dissertation. Stuttgart, 2007. ⇒1

[58] 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. ⇒

[59] E. R. Vaughan, The complexity of constructing gerechte designs. *Electron. J. Combin.*, **16,** (1) (2009), paper R15, pp. 8. MR2475538 (2009k:05041) ⇒1

[60] X. Xu, Y. Cao, J.-M. Xu, Y. Wu, Feedback numbers of de Bruijn digraphs. *Comput. Math. Appl.*, **59,** (4) (2010), 716–723. ⇒1

[61] C. Xu, W. Xu, The model and algorithm to estimate the difficulty levels of Sudoku puzzles. *J. Math. Res.* **11,** (2) (2009), 43–46. ⇒1

[62] W. Zhang, S. Liu, H. Huang, An efficient implementation algorithm for generating de Bruijn sequences. *Computer Standards & Interfaces,* **31,** (6) (2009) 1190–1191. ⇒1