CEJOR manuscript No.
(will be inserted by the editor)

Parallel Erdős-Gallai algorithm

Antal Iványi [1,3] · **Loránd Lucz** [2]

Received: **NOVEMBER 26, 2011** / Accepted: date

Abstract Havel in 1955 [15], Erdős and Gallai in 1960 [10], Hakimi in 1962 [13], Tripathi, Venugopalan and West in 2010 [36] proposed a method to decide, whether a sequence of nonnegative integers can be the degree sequence of a simple graph. The running time of their algorithms in worst case is $\Omega(n^2)$. In [19] the authors proposed a new algorithm called EGL (Erdős-Gallai Linear algorithm), whose worst running time is $\Theta(n)$. As an application of this linear time algorithm we describe Erdős-Gallai-Enumerative algorithm and using its parallel version enumerate the different degree sequences of simple graphs for 24, ..., 29 vertices (compare with [33]).

Keywords Degree sequences · Graphical sequences · Erdős-Gallai theorem · Parallel algorithms

Mathematics Subject Classification (2000) 05C20, 05C85, 68R10

1 Introduction

In the practice an often appearing problem is the ranking of different objects as economical decisions, hardware or software products, cars, persons etc. A typical method of the ranking is the pairwise comparison of the objects, assignment of points to the objects and sorting the objects according to the sums of the numbers of the received points.

For example Landau [25] references to biological, Hakimi [13] to chemical, Kim et al. [23], Newman and Barabási [29] to net-centric, Anholzer, Bozóki, Fülöp, Koczkodaj, Poesz, Rónyai and Temesi to economical [2,4–6,35], Liljeros et al. [26] to human applications, while Frank, Iványi, Lucz, Móri, Sótér and Pirzada [12,16–21] to applications in sports.

[1] Department of Computer Algebra, Eötvös Loránd University. Mail: 1117 Budapest, Pázmány Péter sétány 1/C, Hungary. E-mail: tony@compalg.inf.elte.hu
[2] Department of Computer Algebra, Eötvös Loránd University. Mail: 1117 Budapest, Pázmány Péter sétány 1/C, Hungary. E-mail: lorand.lucz@gmail.com
[3] Corresponding author

Let $n \geq 1$. We will denote the degrees of a simple graph as $d = (d_1, \ldots, d_n)$. We call a sequence $b = (b_1, \ldots, b_n)$ *n-bounded*, if $0 \leq b_i \leq n-1$ for $i = 1, \ldots, n$, and *n-regular*, if the conditions $n-1 \geq b_1 \geq \cdots \geq b_n \geq 0$ hold, and *n-even*, if the sum of the elements of b is even. If there exists a graph with n vertices which has the degree sequence b, then we say that b is *n-graphical*.

The main aim of this paper is to report on the parallel realization of the linear Erdős-Gallai algorithm. Although this problem is interesting in itself, for us the main motivation was our wish to answer the question formulated in the recent monograph [12, Research problem 2.3.1] András Frank: "*Decide if a sequence of n integers can be the final score of a football tournament of n teams.*" During testing and reconstructing of potential football sequences important subproblem is the handling of sequences of draws. And the problems "Is this sequence graphical?" and "Is this sequence a football draw sequence?" are equivalent, therefore the quick answer is vital for us.

The structure of the paper is as follows. After the introductory Section 1 in Section 2 we describe two classical quadratic testing algorithms, then in Section 3 we explain a new linear time algorithm. Section 4 contains the description of the enumerative version of the new linear algorithm, while in Section 5 its parallel implementation is explained. Finally in the last Section 6 we summarize the results.

2 Classical algorithms (Havel-Hakimi and Erdős-Gallai)

In this section we describe two classical testing algorithms due to Václav Havel and Louis Hakimi, resp. Paul Erdős and Tibor Gallai.

2.1 Havel-Hakimi algorithm (HH)

This algorithm was published by Václav Havel Czech mathematician in 1955 [15]. Later Louis Hakimi in 1962 [13] independently published the same result, therefore today the theorem usually is called *Havel-Hakimi theorem*.

Theorem 1 (Hakimi [13], Havel [15]) *If $n \geq 1$, then the n-regular sequence $b = (b_1, \ldots, b_n)$ is n-graphical if and only if the sequence*

$$b' = (b_2 - 1, b_3 - 1, \ldots, b_{b_1} - 1, b_{b_1+1} - 1, b_{b_1+2}, \ldots, b_{n-1}, b_n) \tag{1}$$

is $(n-1)$-graphical.

Proof See [13, 15].

Let $c = (c_1, \ldots c_n)$ and $d = (d_1, \ldots, d_n)$ be n-bounded sequences. In 1965 Hakimi [14] presented a necessary and sufficient condition for c and d to be the in-degree, resp. out-degree sequence of a loopless multigraph. In 2009 this theorem was extended for directed (a, b)-graphs [16, 17]. In 2010 the special case of directed $(0, 1)$-graphs was considered in [11, 24].

Since the direct implementation of the theorem sorts the investigated sequence in each round, it is slow: the worst running time is quadratic even in that case when we use a linear time sorting algorithm.

2.2 Erdős-Gallai algorithm (EG)

In 1960 Paul Erdős and Tibor Gallai [10] Hungarian mathematicians proposed a necessary and sufficient condition to decide, whether an n-regular sequence is n-graphical or not.

Theorem 2 (Erdős, Gallai [10]) *If $n \geq 1$, then the n-regular sequence $b = (b_1, \ldots, b_n)$ is n-graphical if and only if*

$$\sum_{i=1}^{n} b_i \quad \text{is even} \tag{2}$$

and

$$\sum_{i=1}^{j} b_i - j(j-1) \leq \sum_{k=j+1}^{n} \min(j, b_k) \ \ (j = 1, \ldots, n-1). \tag{3}$$

Proof See [7, 10].

In 2010 Tripathi, Venugopalanb and West [36] published a constructive proof, which not only tests the input sequence, but for graphical inputs also constructs a corresponding simple graph. The worst running time of the algorithm based on their proof is $\Theta(n^3)$.

Chungphaisan in 1974 [8], Özkan in 2011 [30] extended Erdős-Gallai theorem for $(0, r)$-graphs.

The following algorithm is based on Theorem 2. In this paper we use the pseudocode prescribed in [9].

Input. n: the length of the input sequence;
$b = (b_1, \ldots, b_n)$: n-regular input sequence.

Output. L : logical variable (L = FALSE shows, that b is not graphical, while if b is graphical, then the returned value is L = TRUE).

Working variable. t: the estimated capacity of the actual tail.

```
ERDŐS-GALLAI(n, b, L)
01 H_1 = b_1                    // lines 01–03: computation of the values of the vector H
02 for i = 2 to n
03     H_i = H_{i-1} + b_i
04 if H_n is odd                // lines 04–06: checking of the parity
05    L = FALSE                 // lines 05–06: refuse of the nongraphical sequences
06    return L
07 for i = 1 to n - 1           // lines 07–15: checking of the input
08     t = 0                    // line 08: initialization of t
09     for k = i + 1 to n       // lines 09–10: computation of the tail capacity
10         t = t + min(i, b_k)
11     if H_i - i(i - 1) > t    // line 11: check the necessary condition
12        L = FALSE             // lines 12–13: the input is nongraphical
13        return L
14 L = TRUE                     // lines 14–15: the input is graphical
15 return L
```

The memory requirement of the algorithm EG is $\Theta(n)$, he running time varies between the best $\Theta(n)$ and the worst $\Theta(n^2)$.

3 Linear Erdős-Gallai algorithm (EGL)

In 2011 we proposed a new algorithm Erdős-Gallai-Linear (EGL) based on the next theorem that even in worst case needs only $\Theta(n)$ time to decide whether the input sequence is graphical.

Theorem 3 (Iványi, Lucz [18], Iványi, Lucz, Móri, Sótér [19]) *If* $n \geq 1$, *then the* n*-regular sequence* $(b_1, \ldots, b_n)$ *is* n*-graphical if and only if*

$$H_n \quad \textit{is even} \tag{4}$$

$$H_i \leq i(Y_i - 1) + H_n - H_{Y_i}, \tag{5}$$

where

$$Y_i(b) = Y_i = \begin{cases} W_i, & \textit{if } i \leq W_i, \\ i, & \textit{if } i > W_i. \end{cases} \tag{6}$$

Proof See in [18, 19].

The following program is based on Theorem 3. It decides on arbitrary n-regular sequence whether it is n-graphical or not.

Input. n: number of vertices ($n \geq 1$);
$b = (b_1, \ldots, b_n)$: n-regular sequence.

Output. L: logical variable, whose value is True, if the input is graphical, and it is False, if the input is not graphical.

Work variables. i and j: cycle variables;
$H = (H_1, \ldots, H_n)$: H_i is the cumulated degree of the first i elements of the tested b;
w: the weight point of the actual b_i, that is the maximum of the indices of such elements of b, which are not smaller than i;
y: the cutting point of the actual b_i that is the maximum of i and w.

```
Erdős-Gallai-Linear(n, b, L)
01 H_1 = b_1                                        // line 01: initialization
02 for i = 2 to n             // line 02–03: computation of the elements of H
03      H_i = H_{i-1} + b_i                      // line 04–06: test of the parity
04 if H_n is odd
05     L = False
06     return
07 w = n                               // line 07: initialization of the weight point
08 for i = 1 to n                         // lines 08–14: test of the elements of b
09      while w > 0 and b_w < i
10            w = w - 1
11      y = max(i, w)
12      if H_i > i(y - i) + H_n - H_y
13          L = False                                // lines 13–14: rejection of b
14          return L
15 L = True                                          // lines 15–16: b is graphical
16 return L
```

The memory requirement of this algorithm is $\Theta(n)$, the time requirement changes between the best $\Theta(n)$ and the worst $\Theta(n^2)$.

It is worth to remark that using a linear sorting algorithm and Erdős-Gallai-Linear we can test even unsorted sequences in linear time.

4 Enumerating Erdős-Gallai algorithm (EGE)

A classical problem of the graph theory the enumeration of the degree sequences of different graphs– among others simple graphs. For example *The On-Line Encyclopedia of Integer Sequences* [33] contains only for $n = 1, \ldots, 23$ vertices the number of degree sequences of simple graphs (the last values for $n = 20, 21, 22$ were set in July of 2011).

Therefore we apply the new quick EGL to get these numbers for larger number of vertices.

Our starting point was to test all regular sequences and so enumerate the graphical ones. It is easy to see that there are

$$R(n) = \binom{2n-1}{n}. \tag{7}$$

In 1987 Ascher derived the following explicit formula for the number of n-even sequences $E(n)$.

Lemma 1 (Ascher [1], Sloane, Pfoffe [34]) *If $n \geq 1$, then the number of n-even sequences $E(n)$ is*

$$E(n) = \frac{1}{2}\left(\binom{2n-1}{n} + \binom{n-1}{\lfloor n \rfloor}\right). \tag{8}$$

Due to the following lemma and its consequence it is enough to test only the zerofree even sequences.

Lemma 2 (Iványi, Lucz, Móri, Sótér [19]) *If $n \geq 2$, then the number of n-graphical sequences $G(n)$ can be computed from the number of $(n-1)$-graphical sequences $G(n-1)$ and the number of n-graphical zero-free sequences $G_z(n)$:*

$$G(n) = G(n-1) + G_z(n). \tag{9}$$

This assertion has the following consequence.

Corollary 1 *If $n \geq 1$ then*

$$G(n) = 1 + \sum_{i=2}^{n} G_z(n). \tag{10}$$

Using the parallel version of EGE we computed G_n till $n = 29$. These numbers can be found in Table 1.

Due to Corollary 1 it is enough to check all of the n-regular zero-free sequences to compute the number of all n-graphical ones. This is a very important idea because only one fourth of the n-regular even sequences are zero-free [19], so we can safe one fourth of the work.

Lemma 3 *If $b = (b_1, \ldots, b_n)$ is a nongraphical sequence and $b_n \geq 3$, then the sequence $b' = (b_1, \ldots, b_{n-1}, c)$ is also nongraphical for every c with $0 \leq c \leq b_{n-1}$.*

n	$R(n)$	$E(n)$	$G(n)$
1	1	1	1
2	3	2	2
3	10	6	4
4	35	19	11
5	126	66	31
6	462	236	102
7	1 716	868	342
8	6 435	3 235	1 213
9	24 310	12 190	4 361
10	92 378	46 252	16 016
11	352 716	176 484	59 348
12	1 352 078	676 270	222 117
13	5 200 300	2 600 612	836 315
14	20 058 300	10 030 008	3 166 852
15	77 558 760	38 781 096	12 042 620
16	300 540 195	150 273 315	45 967 479
17	1 166 803 110	583 407 990	176 005 709
18	4 537 567 650	2 268 795 980	675 759 564
19	17 672 631 900	8 836 340 260	2 600 672 458
20	68 923 264 410	34 461 678 394	10 029 832 754
21	269 128 937 220	134 564 560 988	38 753 710 486
22	1 052 049 481 860	526 024 917 288	149 990 133 774
23	4 116 715 363 800	2 058 358 034 616	581 393 603 996
24	16 123 801 841 550	8 061 901 596 814	**2 256 710 139 346**
25	63 205 303 218 876	31 602 652 961 516	**8 770 547 818 956**
26	247 959 266 474 052	123 979 635 837 176	**34 125 389 919 850**
27	973 469 712 824 056	486 734 861 612 328	**132 919 443 189 544**
28	3 824 345 300 380 220	1 912 172 660 219 260	**518 232 001 761 434**
29	15 033 633 249 770 520	7 516 816 644 943 559	**2 022 337 118 015 338**

Table 1 Number of regular, even and graphical sequences.

Taking into account these results we have to test only one tenth of the regular sequences.

Two results due to Tripathi and Vijay [37, Lemma 6, Theorem 7,Corollary 17] allow to reduce the testing time with more than 50 percent.

The following algorithm Erdős-Gallai-Enumerating (EGE) is an enumerative version of EGL. In this algorithm we use some techniques to speed up the computing process. From one side we we check only zero-free, even sequences. And we used lexicographical order, so most of the operations could be saved by updating the old values. We were updating the following values.

- H_i values: most of the time the only thing that is changing is the last element of the sequence b, so it is enough to update the last H values, according to the changes of the values of b.
- C_i checkpoints: if we modify the ith element of the sequences then the values before that point remain the same so all of the checkpoint before that will be the same, so we update only the first one before the ith index and all of them after it.
- W_i weight points: every time the checking algorithm got a sequence to check we update the weight points we use, but we never start from 1 or n. We use the last value we used when we checked a sequence in that index. We have a distinct weight point for every i index and we just shift the value to left or right.

– Y_i cutting points:

We suppose that n, b, H, c, C, W and Y are global variables, therefore their **return** does not require additional time.

Important property of EGE is that it solves in $\Theta(1)$ average time

– the generation of one zerofree even sequence;
– the updating of the sequence of the cumulated degrees H;
– the updating of the sequence of the checking points C;
– the updating of the sequence of the weight points W.

The average running time of this algorithm for a sequence is $\Theta(1)$, so the total running time of the whole program is $\Theta(E(n))$.

Erdős-Gallai-Enumerating(n, G_z)

```
01 for i = 1 to n                                   // lines 01–09: initialization
02      b_i = n − 1
03      H_i = i(n − 1)
04      W_i = n
05      y_i = n − 1
06      C_i = 0
07 G_z = 1
08 c = 0
09 b_{n+1} = −1
10 while b_2 ≥ 2 or b_1 ≥ 3                         // line 10: last sequence was?
11         if b_n ≥ 3 New3(n, b, H, c, C, W)  // lines 11–13: generate the next sequence
12         if b_n = 2 New2(n, b, H, c, C, W)
13         if b_n = 1 New1(n, b, H, c, C, W)
14         Check(n, b, H, c, W)          // line 12: checks and updates the parameters
15         G_z = G_z + L                              // line 13: increasing of G_z
16 print G_z                                          // line 14: final result
```

This algorithm uses two procedures. New generates a new sequences and update the key parameters, while Check decides whether the actually investigated sequence is graphical or not.

In Check we use condition (2) of Erdős-Gallai theorem.

Check(n, b, H, c, C, W)

```
01 for i = 1 to c                            // lines 01–07: checking in checkpoints
02      y = max w, i             // line 02: computation of the actual cutting point
03      if H_i > i(y − 1) + H_n − H_y                     // line 03–05: EG checking
04         L = 0
05         return L
06 L = 1                                              // line 06–07: b is graphical
07 return L
```

New1(n, b, H, c, C)

01 **if** $b_n \geq 3$ // line 01–06: generation if $b_n = 3$
02 $b_n = b_n - 2$
03 $H_n = H_n - 2$
04 **if** $b_n == b_{n-1} - 3$
05 $c = c + 1$
06 $C_c = n - 1$
07 $W = ?$
08 **return** H, C, W

New2(n, b, H, c, C)
01 **if** $b_n == 2$ // line 08–16: generation if $b_n = 2$
02 **if** $b_n = b_{n-1} - 1$
03 $H_n = H_n - 1$
04 $c = c - 1$
05 **if** b_{n-1} is odd
06 **if** $b_{n-2} = b_{n-1} + 1$
07 $c = c - 1$
08 $W = ???$
09 **return** H, C, W

New3 is similar to New2 although more complicated.

5 Parallel Erdős-Gallai algorithm (EGP)

To use our new linear time algorithm on a bunch of series we need an algorithm that can work on a part of all series we want to check. From now we will use the notation $G(n)$ for the number of n-graphical series and $G_z(n)$ for the number of n-graphical zero-free series. The reason to use z_n values is the following lemma.

Using our *Parallel* algorithm we computed this number till $n = 29$. These number can be found in Table 1.

Originally we used a server-client application consisting of two parts: server and client. The server had all the information to distribute jobs between client machines and collect results from them and the client had the IP address and the PORT of the server to ask for a job. To compute $G_z(n)$ value on multiple computers first we need to decompose the whole problem to smaller parts so called *jobs*, that we could pass to them. This is one of the most critical part of the parallel algorithm: divide the problem into almost same sized jobs. The next equation helps us to count approximately how many sequences starts with a fixed head. By knowing these numbers we can generate jobs with limited size, so every job is smaller than that maximum.

Generate-Matrix($n, MaxSize$)
01 **for** $i = n$ **downto** 2 ▷ Lines 01–05: filling up the matrix
02 **for** $j = 1$ **to** $n - 1$
03 $M_{i,j} = \binom{i+j-2}{i-1}$
04 **end**
05 **end**
06 **for** $j = n - 1$ **downto** 1 ▷ Lines 06–08: fill up the first line in matrix
07 $M_{1,j} = 1$
08 **end**

09 **Generate-Sequences**$(M, n, n, 1, n-1, MaxSize, 0)$
▷ Line 09: job generation

This algorithm give us a matrix filled up with values computed with the equation. Now, we can generate the sequences by read out the last row from the matrix from left to right. If a value is too big and not fit into a job, then we can move one line above that value and read that line from the first column until the one that was too big and we can continue this technique until we got size of parts we need. The next (recursive) algorithms reads out the last row with this method.

In the *On-Line Encyclopedia of Integer Sequences* [33] you can find numbers of degree-vectors for simple graphs, that consist of n vertices, where n is from 1 to 23. Originally we used a server-client application consisting of two parts: server and client. The server had all the information to distribute jobs between client machines and collect results from them and the client had the IP address and the PORT of the server to ask for a job.

During the calculations we used more than two hundred computers and our theoretical maximal performance was over 6 TFLOPS based on the processors information we found on the manufacturers home pages.

6 Summary

Acknowledgements The authors are indebted to Zoltán Király (Eötvös Loránd University, Faculty of Science, Dept. of Operation Research) for the useful continuous consultation, Antal Sándor and his colleagues (Eötvös Loránd University, Faculty of Informatics), Ádám Mányoki (TFM World Kereskedelmi és Szolgáltató Kft.) and Zoltán Kása (Sapientia Hungarian University of Transylvania) for their help in running of our timeconsuming programs. 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. Ascher M (1987) Mu torere: an analysis of a Maori game, Math. Mag 60(2):90–100
2. Anholcer M, Babiy V, Bozóki S, Koczkodaj W. V (2011) A simplified implementation of the least squares solution for pairwise comparisons matrices. Central Eur J Oper Res 19(4):439–444
3. Barnes TM, Savage CD (1997) Efficient generation of graphical partitions. Discrete Appl Math 78:17–26
4. Bozóki S, Fülöp J, Koczkodaj WW (2011) An LP-based inconsistency monitoring of pairwise comparison matrices. Math Comp Modelling 54:789–793
5. Bozóki S, Fülöp J, Poesz A (2011) On pairwise comparison matrices that can be made consistent by the modification of a few elements. Central Eur J Oper Res 19(2):157–175
6. Bozóki S, Fülóp J, Rónyai L (2010) On optimal completion of incomplete pairwise comparison matrices. Math Comput Modelling 52:318–333
7. Choudum, SA (1986) A simple proof of the Erdős-Gallai theorem on graph sequences. Bull Austral Math Soc 33:67–70
8. Chungphaisan, V (1974) Conditions for sequences to be r-graphical. Discrete Math. 7:31–39
9. Cormen TH, Leiserson ChE, Rivest RL, Stein C (2009) Introduction to Algorithms (Third edition). The MIT Press/McGraw Hill, Cambridge/New York
10. Erdős P, Gallai T (1960) Graphs with prescribed degree sequence (Hungarian). Mat Lapok 11:264–274
11. Erdős P, Miklós I, Toroczkai Z (2010) A simple Havel-Hakimi type algorithm to realize graphical degree sequences of directed graphs. Electron J Combin 17:1 R66 (10 pages).

12. Frank A (2011) Connections in Combinatorial Optimization. Oxford University Press, Oxford
13. Hakimi SL (1962) On the realizability of a set of integers as degrees of the vertices of a simple graph. J SIAM Appl Math 10:496–506
14. Hakimi SL (1965) On the degrees of the vertices of a graph. F Franklin Institute 279(4)290–308
15. Havel V (1955) A remark on the existence of finite graphs (Czech). Časopis Pěst Mat 80:477–480
16. Iványi A (2009) Reconstruction of complete interval tournaments. Acta Univ Sapientiae, Inform 1(1):71–88
17. Iványi A (2010), Reconstruction of complete interval tournaments. II. Acta Univ Sapientiae, Math 2(1):47–71
18. Iványi A, Lucz L (2011) Erdős-Gallai test in linear time. Combinatorica (submitted).
19. Iványi A, Lucz L, Móri FT, Sótér P (2011) Quick Erdős-Gallai algorithms, Acta Univ Sapientiae, Inform 3(2):230-268
20. Iványi A, Lucz L, Móri FT (2011) Linear Erdős-Gallai test (Hungarian) Alk Mat Lapok (submitted)
21. Iványi A, Pirzada S (2011) Comparison based ranking. In: Algorithms of Informatics (electronic book, ed. by A. Iványi), AnTonCom, Budapest, 1410–1450
22. Kéri G. (2011) On qualitatively consistent, transitive and contradictory judgment matrices emerging from multiattribute decision procedures. Central Eur J Oper Res 19(2):215–224
23. Kim H, Toroczkai Z, Miklós I, Erdős PL, Székely LA (2009) Degree-based graph construction. J Physics: Math Theor A 42:392–401
24. LaMar MD (2010) Algorithms for realizing degree sequences of directed graphs. arXiv-0906:0343ve [math.CO], 7 June 2010
25. Landau HG (1953) On dominance relations and the structure of animal societies. III. The condition for a score sequence. Bull Math Biophys 15:143–148
26. Liljeros F, Edling CR, Amaral A, Stanley H, Åberg Y (2001) The web of human sexual contacts. Nature 411(6840):907–908
27. Lucz L, Iványi A (2012) Testing and enumeration of football sequences, Abstracts of MaCS 2012, ed. by Z. Csörnyei (Siófok, February 9–12, 2012).
28. Narayana TV, Bent DH (1964) Computation of the number of score sequences in round-robin tournaments. Canad Math Bull 7:133–136
29. Newman MEJ, Barabási AL (2006) The Structure and Dynamics of Networks. Princeton University Press, Princeton, NJ
30. Özkan S (2011) Generalization of the Erdős-Gallai inequality. Ars Combin 98:295–302
31. Pécsy G, Szűcs L (2000) Parallel verification and enumeration of tournaments. Studia Univ Babeş-Bolyai, Inform XLV(2):11–26
32. Ruskey F, Cohen R, Eades P, Scott A (1994) Alley CAT's in search of good homes. Congr Numer 102:97–110
33. Sloane NJA (2011) The number of degree-vectors for simple graphs. In: The On-Line Encyclopedia of the Integer Sequences (ed. by NJA Sloane). http://oeis.org/A004251
34. Sloane NJA, Plouffe S (1995) The Encyclopedia of Integer Sequences. Academic Press, 1995.
35. Temesi J (2011) Pairwise comparison matrices and the error-free property of the decision maker. Central Eur J Oper Res 19 (2) 239–249
36. Tripathi A, Venugopalanb S, West DB (2010) A short constructive proof of the Erdős-Gallai characterization of graphic lists. Discrete Math 310:833–834
37. Tripathi A, Vijay S (2003) A note on a theorem of Erdős & Gallai. Discrete Math 265:417–420