On Erdős-Gallai and Havel-Hakimi algorithms

Antal Iványi
Department of Computer Algebra, Eötvös Loránd University, Hungary
email: tony@inf.elte.hu

Loránd Lucz
Department of Computer Algebra, Eötvös Loránd University, Hungary
email: lorand.lucz@gmail.com

Tamás F. Móri
Department of Probability Theory and Statistics, Eötvös Loránd University, Hungary
email: moritamas@ludens.elte.hu

Péter Sótér
Department of Computer Algebra, Eötvös Loránd University, Hungary
email: mapoleon@freemail.hu

Abstract. Havel in 1955 [28], Erdős and Gallai in 1960 [21], Hakimi in 1962 [26], Ruskey, Cohen, Eades and Scott in 1994 [69], Barnes and Savage in 1997 [6], Kohnert in 2004 [49], Tripathi, Venugopalan and West in 2010 [83] 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 is $\Omega(n^2)$ in worst case. In this paper we propose a new algorithm called EGL (Erdős-Gallai Linear algorithm), whose worst running time is $\Theta(n)$. As an application of this quick algorithm we computed the number of the different degree sequences of simple graphs for 24, ..., 29 vertices (see [74]).

1 Introduction

In the practice an often appearing problem is the ranking of different objects as hardware or software products, cars, economical decisions, 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 [51] references to biological, Hakimi [26] to chemical, Kim et al. [45], Newman and Barabási [61] to net-centric, Bozóki, Fülop, Poesz, Kéri, Rónyai and Temesi to economical [1, 10, 11, 42, 80], Liljeros et al. [52] to human applications, while Iványi, Khan, Lucz, Pirzada, Sótér and Zhou [30, 31, 38, 65, 67] write on applications in sports.

From several popular possibilities we follow the terminology and notations used by Paul Erdős and Tibor Gallai [21].

Depending from the rules of the allocation of the points there are many problems. In this paper we deal only with the case when the comparisons have two possible results: either both objects get one point, or both objects get zero points. In this case the results of the comparisons can be represented using simple graphs and the number of points gathered by the given objects are the degrees of the corresponding vertices. The decreasing sequence of the degrees is denoted by \(b = (b_1, \ldots, b_n)\).

From the popular problems we investigate first of all the question, how can we quickly decide, whether for given \(b\) does exist there a simple graph \(G\) whose degree sequence is \(b\). In connection with this problem we remark that the main motivation for studying of this problem is the question: what is the complexity of deciding whether a sequence is the score sequence of some football tournament [24, 32, 35, 36, 43, 44, 54].

As a side effect we extended the popular data base On-line Encyclopedia of Integer Sequences [72] with the continuation of contained sequences.

In connection with the similar problems we remark, that in the last years a lot of papers and chapters were published on the undirected graphs (for example [8, 9, 12, 16, 19, 29, 37, 41, 55, 68, 81, 83, 84, 85]) and also on directed graphs (for example [7, 11, 14, 23, 24, 30, 31, 33, 38, 45, 48, 50, 57, 58, 63, 65, 64, 66]).

The majority of the investigated algorithms is sequential, but there are parallel results too [2, 18, 20, 36, 60, 62, 77].

Let \(l, u\) and \(m\) integers (\(m \geq 1\) and \(u \geq 1\)). A sequence of integer numbers \(b = (b_1, \ldots, b_m)\) is called \((l, u, m)\)-bounded, if \(l \leq b_i \leq u\) for \(i = 1, \ldots, m\). A \((l, u, m)\)-bounded sequence \(b\) is called \((l, u, m)\)-regular, if \(b_m \geq b_{m-1} \geq \cdots \geq b_1\). An \((l, u, m)\)-regular sequence is called \((l, u, m)\)-even, if the sum of its elements is even. A \((0, n-1, n)\)-regular sequence \(b\) is called \(n\)-graphical, if there exists a simple graph \(G\) whose degree sequence is \(b\). If \(l = 0\), \(u = n - 1\) and \(m = n\), then we use the terms \(n\)-bounded, \(n\)-regular, \(n\)-even, and \(n\)-
graphical (or simply bounded, regular, even, graphical).

In the following we deal first of all with regular sequences. In our definitions the bounds appear to save the testing algorithms from the checking of such sequences, which are obviously not graphical, therefore these bounds do not mean the restriction of the generality.

The paper consists of nine parts. After the introductory Section 1 in Section 2 we describe the classical algorithms of the testing and reconstruction of degree sequences of simple graphs. Section 3 introduces several linear testing algorithms, then Section 4 summarizes some properties of the approximate algorithms. Section 5 contains the description of new precise algorithms and in Section 6 the running times of the classical testing algorithms are presented. Section 7 contains enumerative results, in Section 8 we report on the application of the new algorithms for the computation of the number of score sequences of simple graphs. Finally Section 9 contains the summary of the results.

Our paper [37] written in Hungarian contains further algorithms and simulation results. [35] contains a short summary on the linear Erdős-Gallai algorithm while in [36] the details of the parallel implementation of enumerating Erdős-Gallai algorithm are presented.

2 Classical precise algorithms

For a given \( n \)-regular sequence \( b = (b_1, \ldots, b_n) \) the first \( i \) elements of the sequence we call the head of the sequence belonging to the index \( i \), while the last \( n - i \) elements of the sequence we call the tail of the sequence belonging to the index \( i \).

2.1 Havel-Hakimi algorithm

The first algorithm for the solution of the testing problem was proposed by Vaclav Havel Czech mathematician [28, 53]. In 1962 Louis Hakimi [26] published independently the same result, therefore the theorem is called today usually as Havel-Hakimi theorem, and the method of reconstruction is called Havel-Hakimi algorithm.

**Theorem 1** (Hakimi [26], Havel [28]). If \( n \geq 3 \), then the \( n \)-regular sequence \( b = (b_1, \ldots, b_n) \) is \( n \)-graphical if and only if the sequence \( b' = (b_2 - 1, b_3 - 1, \ldots, b_{b_1} - 1, b_{b_1+1} - 1, b_{b_1+2}, \ldots, b_n) \) is \((n - 1)\)-graphical.

**Proof.** See [9, 26, 28, 37].
If we write a recursive program based on this theorem, then according to the RAM model of computation its running time will be in worst case $\Omega(n^2)$, since the algorithm decreases the degrees by one, and e.g. if $b = ((n - 1)^n)$, then the sum of the elements of $b$ equals to $\Theta(n^2)$. It is worth to remark that the proof of the theorem is constructive, and the algorithm based on the proof not only tests the input in quadratic time, but also construct a corresponding simple graph (of course, only if it there exists).

It is worth to remark that the algorithm was extended to directed graphs in which any pair of the vertices is connected with at least $a \geq 0$ and at most $b \geq a$ edges [30, 31]. The special case $a = b = 1$ was reproved in [23].

In 1965 Hakimi [27] gave a necessary and sufficient condition for two sequences $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ to be the in-degree sequences and out-degree sequence of a directed multigraph without loops.

### 2.2 Erdős-Gallai algorithm

In chronological order the next result is the necessary and sufficient theorem published by Paul Erdős and Tibor Gallai [21].

For an $n$-regular sequence $b = (b_1, \ldots, b_n)$ let $H_i = b_1 + \cdots + b_i$. For given $i$ the elements $b_1, \ldots, b_i$ are called the head of $b$, belonging to $i$, while the elements $b_{i+1}, \ldots, b_n$ are called the tail of $b$ belonging to $i$.

When we investigate the realizability of a sequence, a natural observation is that the degree requirement $H_i$ of a head is covered partially with inner and partially with outer degrees (with edges among the vertices of the head, resp. with edges, connecting a vertex of the head and a vertex of the tail). This observation is formalized by the following Erdős-Gallai theorem.

**Theorem 2** (Erdős, Gallai [21]) Let $n \geq 3$. The $n$-regular sequence $b = (b_1, \ldots, b_n)$ is $n$-graphical if and only if

$$
\sum_{i=1}^{n} b_i \text{ even} \quad (1)
$$

and

$$
\sum_{i=1}^{j} b_i \leq j(j-1) + \sum_{k=j+1}^{n} \min(j,b_k) \quad (j = 1, \ldots, n-1). \quad (2)
$$

**Proof.** See [9, 15, 21, 70, 83].
Figure 1: Number of regular and even sequences, and the ratio of these numbers

Although this theorem does not solve the problem of reconstruction of graphical sequences, the systematic application of (2) requires in worst case
On Erdős-Gallai and Havel-Hakimi algorithms

(For example when the input sequence is graphical) $\Theta(n^2)$ time.

Recently Tripathi and Vijay [83] published a constructive proof of Erdős-Gallai theorem and proved that their construction requires $O(n^3)$ time.

Figure 1 shows the number of $n$-regular ($R(n)$) and $n$-even ($E(n)$) sequences and their ratio $(E(n)/R(n))$ for $n = 1, \ldots, 38$. According to (34) the sequence of these ratios tends to $\frac{1}{2}$ as $n$ tends to $\infty$. According to Figure 1 the convergence is quick: e.g. $E(20)/R(20) = 0.5000006701511$.

The pseudocode of Erdős-Gallai see in [37].

3 Testing algorithms

We are interested in the investigation of football sequences, where often appears the necessity of the testing of degree sequences of simple graphs.

A possible way to decrease the expected testing time is to use quick (linear) filtering algorithms which can state with a high probability, that the given input is not graphical, and so we need the slow precise algorithms only in the remaining cases.

Now we describe a parity checking, then a binomial, and finally a headsplitting filtering algorithm.

3.1 Parity test

Our first test is based on the first necessary condition of Erdős-Gallai theorem. This test is very effective, since according to Figure 1 and Corollary 14 about the half of the regular sequences is odd, and our test establishes in linear time, that these sequences are not graphical.

The following simple algorithm is based on (1).

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

Output. $L$: logical variable ($L = \text{False}$ shows, that $b$ is not graphical, while the meaning of the value $L = \text{True}$ is, that the test could not decide, whether $b$ is graphical or not).

Working variable. $i$: cycle variable;
$H = (H_1, \ldots, H_n)$: $H_i$ is the sum of the first $i$ elements of $b$.

Parity-Test($n, b, L$)

01 $H_1 = b_1$
02 for $i = 2$ to $n$
03 $H_i = H_{i-1} + b_i$
The running time of this algorithm is $\Theta(n)$ in all cases. Figure 1 characterizes the efficiency of \textsc{Parity-test}.

(1) is only a necessary condition, therefore \textsc{Parity-Test} is only an approximate (filtering) algorithm.

3.2 Binomial test

Our second test is based on the second necessary condition of Erdős-Gallai theorem. It received the given name since we estimate the number of the inner edges of the head of $b$ using a binomial coefficient. Let \( T_i = b_{i+1} + \cdots + b_n \) \((i = 1, \ldots, n)\).

**Lemma 3** If $n \geq 1$ and \( b \) is an $n$-graphical sequence, then

\[
H_i \leq i(i - 1) + T_i \quad (i = 1, \ldots, n - 1).
\]

**Proof.** The left side of (3) represents the degree requirement of the head of $b$. On the right side of (3) $i(i - 1)$ is an upper bound of the inner degree capacity of the head, while $T_i$ is an upper bound of the degree capacity of the tail, belonging to the index $i$. \hfill \square

The following program is based on Lemma 3.

**Input.** $n$: number of the vertices ($n \geq 1$);

\( b = (b_1, \ldots, b_n) \): an $n$-regular even sequence;

\( H = (H_1, \ldots, H_n) \): $H_i$ the sum of the first $i$ elements of $b$;

$H_0$: auxiliary variable, helping to compute the elements of $H$.

**Output.** $L$: logical variable ($L = \text{FALSE}$ signals, that $b$ is surely not graphical, while $L = \text{TRUE}$ shows, that the test could not decide, whether $b$ is graphical).

**Working variables.** $i$: cycle variable;

\( T = (T_1, \ldots, T_n) \): $T_i$ the sum of the last $n - i$ elements of $b$;

$T_0$: auxiliary variable, helping to compute the elements of $T$.

**Binomial-Test**($n, b, H, L$)

01 $T_0 = 0$
02 for $i = 1$ to $n - 1$
The running time of this algorithm is $\Theta(n)$ in worst case, while in best case is only $\Theta(1)$.

According to our simulation experiments Binomial-Test is an effective filtering test (see Figure 2 and Figure 3).

### 3.3 Splitting of the head

We can get a better estimation of the inner capacity of the head, than the binomial coefficient gives in (3), if we split the head into two parts. Let $[i/2] = h_i$, $p$ the number of positive elements of $b$. Then the sequence $(b_1,\ldots,b_{h_i})$ is called the beginning of the head belonging to index $i$ and the sequence $(b_{h_i+1},\ldots,b_i)$ the end of the head belonging to index $i$.

**Lemma 4** If $n \geq 1$ and $b$ is an $n$-graphical sequence, then

$$H_i \leq \min \left( \min(H_{h_i}, T_n - T_i, h_i(n - i)) + \min(H_i - H_{h_i}, T_n - T_i, (i - h_i)(n - i)), T_i \right) + \min(h_i(i - h_i) + \frac{h_i}{2} + \frac{(1 - h_i)}{2}) (i = 1,\ldots,n), \quad (4)$$

further

$$\min(H_{h_i}, T_n - T_i, h_i(n - i)) + \min(H_i - H_{h_i}, T_n - T_i, (i - h_i)(n - i)) \leq T_i. \quad (5)$$

**Proof.** Let $G$ be a simple graph whose degree sequence is $b$. Then we divide the set of the edges of the head belonging to index $i$ into five subsets: $(S_{i,1})$ contains the edges between the beginning of the head and the tail, $(S_{i,2})$ the edges between the end of the head and the tail, $S_{i,3}$ the edges between the parts of the head, $S_{i,4}$ the edges in the beginning of the head and $S_{i,5}$ the edges in the end of the head. Let us denote the number of edges in these subsets by $X_{i,1},\ldots,X_{i,5}$.

$X_{i,1}$ is at most the sum $H_{h_i}$ of the elements of the head, at most the sum $T_n - T_i$ of the elements of the tail, and at most the product $h_i(n - i)$ of the
elements of the pairs formed from the tail and from the beginning of the head, that is
\[ X_{i,1} \leq \min(H_{hi}, T_n - T_i, h_i(n - i)). \] (6)
A similar train of thought results
\[ X_{i,2} \leq \min(H_i - H_{hi}, T_n - T_i, (i - h_i)(n - i)). \] (7)
\[ X_{i,3} \leq \min(h_i(i - h_i), H_i). \] (8)
\[ X_{i,4} \leq \min\left(\frac{h_i}{2}, H_{hi}\right), \] (9)
while \[ X_{i,5} \leq \frac{i - h_i}{2}. \] (10)
A requirement is also, that the tail can overrun its capacity, that is
\[ X_{i,1} + X_{i,2} \leq T_i. \] (11)
Summing of (6), (7), (8), (9), and (10) results
\[ H_i \leq X_{i,1} + X_{i,2} + X_{i,3} + 2X_{i,4} + 2X_{i,5}. \] (12)
Substituting of (6), (7), (8), (9), and (10) into (12) results (4), while (11) is equivalent to (5).
□

The following algorithm executes the test based on Lemma 4.

Input. \( n \): the number of vertices \( n \geq 1 \);
\( b = (b_1, \ldots, b_n) \): an \( n \)-even sequence, accepted by BINOMIAL-TEST;
\( H = (H_1, \ldots, H_n) \): \( H_i \) the sum of the first \( i \) elements of \( b \);
\( T = (T_1, \ldots, T_n) \): \( T_i \) the sum of the last \( n - i \) elements of \( b \).

Output. \( L \): logical variable \( L = \text{FALSE} \) signals, that \( b \) is not graphical, while \( L = \text{TRUE} \) shows, that the test could nor decide, whether \( b \) is graphical).

Working variables. \( i \): cycle variable;
\( h_i \): the actual value of \( h_i \);
\( X = (X_1, X_2, X_3, X_4, X_5) \): \( X_j \) is the value of the actual \( X_{i,j} \).
Headsplitter-Test\((n, b, H, T, L)\)

01 for i = 2 to n − 1
02 h = ⌊i/2⌋
03 \(X_1 = \min(H_{h}, T_{n - T}, h(n - i))\)
04 \(X_2 = \min(H_{i} - H_{h}, T_{n} - T_{i}, (i - h)(n - i))\)
05 \(X_3 = \min(h(i - h))\)
06 \(X_4 = \binom{h}{2}\)
07 \(X_5 = \binom{i - h}{2}\)
08 if \(H_i > X_1 + X_2 + X_3 + 2X_4 + 2X_5\) or \(X_1 + X_2 > T_i\)
09 \(L = \text{FALSE}\)
10 return L
11 L = \text{TRUE}\)
12 return L

The running time of the algorithm is \(\Theta(1)\) in best, and \(\Theta(n)\) in worst case. It is a substantial circumstance that the use of Lemma 3 and Lemma 4 requires only linear time (while the earlier two theorems require quadratic time). But these improvements of Erdős-Gallai theorem decrease only the coefficient of the quadratic member in the formula of the running time, the order of growth remains unchanged.

Figure 2 contains the results of the running of Binomial-Test and Headsplitter-Test, further the values \(G(n)\) and \(\frac{G(n)}{G(n+1)}\) (the computation of the values of the function \(G(n)\) will be explained in Section 8).

Figure 3 shows the relative frequency of the zero-free regular, binomial, head-splitted and graphical sequences compared to the number of regular sequences.

## 3.4 Composite test

Composite-Test uses approximate algorithms in the following order: Parity-Test, Binomial-Test, Positive-Test, Headsplitter-Test.

Composite-Test\((n, b, L)\)

01 Parity-Test\((n, b, L)\)
02 if L == False
03 return L
04 Binomial-Test\((n, b, H, L)\)
05 if L == False
06 return L
07 Headsplitter-Test\((n, b, H, T, L)\)
The running time of this composite algorithm is in all cases $\Theta(n)$.

4 Properties of the approximate testing algorithms

We investigate the efficiency of the approximate algorithms testing the regular algorithms. Figure 1 contains the number $R(n)$ of regular, the number $E(n)$ of even, and the number $G(n)$ of graphical sequences for $n = 1, \ldots, 38$.

The relative efficiency of arbitrary testing algorithm $A$ for sequences of given length $n$ we define with the ratio of the number of accepted by $A$ sequences of length $n$ and the number of graphical sequences $G(n)$. This ratio as a function of $n$ will be noted by $X_A(n)$ and called the error function of $A$ [34].

We investigate the following approximate algorithms, which are the components of Composite-Test:

1) Parity-Test;
2) Binomial-Test;
3) Headsplitter-Test.

According to (23) there are $R(2) = 3$ 2-regular sequences: $(1,1)$, $(1,0)$ and $(0,0)$. According to (25) among these sequences there are $E(2) = 2$ even sequences. Binomial-Test accepts both even ones, therefore $B(2) = 2$. Both sequences are 2-graphical, therefore $G(2) = 2$ and so the efficiency of Parity-Test (PT) and Binomial-Test (BT) is $X_{PT}(2) = X_{BT}(2) = 2/2 = 1$, in this case both algorithms are optimal.

The number of 3-regular sequences is $R(3) = 10$. From these sequences $(2,2,2)$, $(2,2,0)$, $(2,1,1)$, $(2,0,0)$ $(1,1,0)$ and $(0,0,0)$ are even, so $E(3) = 6$. Binomial-Test excludes the sequences $(2,2,0)$ and $(2,0,0)$, so remains $B(3) = 4$. Since these sequences are 3-graphical, $G(3) = 4$ implies $X_{PT}(3) = \frac{3}{2}$ and $X_{BT}(3) = 1$.

The number of 4-regular sequences equals to $R(4) = 35$. From these sequences 16 is even, and the following 11 are 4-graphical: $(3,3,3,3)$, $(3,3,2,2)$, $(3,2,2,1)$, $(3,1,1,1)$, $(2,2,2,2)$, $(2,2,2,0)$, $(2,2,1,1)$, $(2,1,1,0)$, $(1,1,1,1)$, $(1,1,0,0)$ and $(0,0,0,0)$. From the 16 even sequences Binomial-Test also excludes the 5 sequences, so $B(4) = G(4) = 11$ and $X_{BT}(4) = 1$.

According to these data in the case of $n \leq 4$ Binomial-Test recognizes all nongraphical sequences. Figure 2 shows, that for $n \leq 5$ we have $B(n) = G(n)$,
On Erdős-Gallai and Havel-Hakimi algorithms

<table>
<thead>
<tr>
<th>n</th>
<th>$B_z(n)$</th>
<th>$F_z(n)$</th>
<th>$G(n)$</th>
<th>$G(n+1)/G(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2.000000</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2.000000</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2.750000</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>2.818182</td>
</tr>
<tr>
<td>5</td>
<td>31</td>
<td>31</td>
<td>31</td>
<td>3.290323</td>
</tr>
<tr>
<td>6</td>
<td>103</td>
<td>102</td>
<td>102</td>
<td>3.352941</td>
</tr>
<tr>
<td>7</td>
<td>349</td>
<td>344</td>
<td>342</td>
<td>3.546784</td>
</tr>
<tr>
<td>8</td>
<td>1256</td>
<td>1230</td>
<td>1213</td>
<td>3.595218</td>
</tr>
<tr>
<td>9</td>
<td>4577</td>
<td>4468</td>
<td>4361</td>
<td>3.672552</td>
</tr>
<tr>
<td>10</td>
<td>17040</td>
<td>16582</td>
<td>16016</td>
<td>3.705544</td>
</tr>
<tr>
<td>11</td>
<td>63944</td>
<td>62070</td>
<td>59348</td>
<td>3.742620</td>
</tr>
<tr>
<td>12</td>
<td>242218</td>
<td>234596</td>
<td>222117</td>
<td>3.765200</td>
</tr>
<tr>
<td>13</td>
<td>922369</td>
<td>891852</td>
<td>836315</td>
<td>3.786674</td>
</tr>
<tr>
<td>14</td>
<td>3530534</td>
<td>3409109</td>
<td>3166852</td>
<td>3.802710</td>
</tr>
<tr>
<td>15</td>
<td>13563764</td>
<td>13082900</td>
<td>12042620</td>
<td>3.817067</td>
</tr>
<tr>
<td>16</td>
<td>52283429</td>
<td>50380684</td>
<td>45967479</td>
<td>3.828918</td>
</tr>
<tr>
<td>17</td>
<td>202075949</td>
<td>194550002</td>
<td>17605709</td>
<td>3.839418</td>
</tr>
<tr>
<td>18</td>
<td>782879161</td>
<td>753107537</td>
<td>675759564</td>
<td>3.848517</td>
</tr>
<tr>
<td>19</td>
<td>3039168331</td>
<td>2921395019</td>
<td>260067258</td>
<td>3.856630</td>
</tr>
<tr>
<td>20</td>
<td>11819351967</td>
<td>11353359464</td>
<td>10029832754</td>
<td>3.863844</td>
</tr>
<tr>
<td>21</td>
<td>43875371046</td>
<td>41999333774</td>
<td>39870343</td>
<td>3.870343</td>
</tr>
<tr>
<td>22</td>
<td>149990133774</td>
<td>13583603996</td>
<td>3.876212</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>581393603996</td>
<td>54540089372</td>
<td>3.881553</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>2256710139346</td>
<td>203914013284</td>
<td>3.886431</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>8770547818956</td>
<td>827083991956</td>
<td>3.890907</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>34125389919850</td>
<td>313977990990</td>
<td>3.895031</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>132919443189544</td>
<td>12059443189544</td>
<td>3.897978</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>51823200761434</td>
<td>48423100621434</td>
<td>3.898843</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>2022337118015338</td>
<td>1908137116015338</td>
<td>3.899843</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2: Number of zerofree binomial, zerofree headsplitted and graphical sequences, further the ratio of the numbers of graphical sequences for neighbouring values of \( n \)

That is Binomial-Test accepts the same number of sequences as the precise algorithms. If \( n > 5 \), then the error function of Binomial-Test is increasing: while \( X_{BT}(6) = \frac{102}{102} \) (BT accepts one nongraphical sequence), \( X_{BT}(7) = \frac{349}{349} \) (BT accepts 7 nongraphical sequences) etc.

Figure 4 presents the average running time of the testing algorithms BT
Figure 3: The number of zero-free even sequences, further the ratio of the numbers binomial/regular, headsplit/regular and graphical/regular sequences and HT in seconds and in number of operations. The data contain the time and operations necessary for the generation of the sequences too.
5 New precise algorithms

In this section the zerofree algorithms, the shifting Havel-Hakimi, the parity checking Havel-Hakimi, the shortened Erdős-Gallai, the jumping Erdős-Gallai, the linear Erdős-Gallai and the quick Erdős-Gallai algorithms are presented.

5.1 Zerofree algorithms

Since the zeros at the and of the input sequences correspond to isolated vertices, so they have no influence on the quality of the sequence. This observation is exploited in the following assertion, in which \( p \) means the number of the positive elements of the input sequence.

**Corollary 5** If \( n \geq 1 \), the \((b_1, \ldots, b_n)\) \( n \)-regular sequence is \( n \)-graphical if and only if \((b_1, \ldots, b_p)\) is \( p \)-graphical.

**Proof.** If all elements of \( b \) are positive (that is \( p = n \)), then the assertion is equivalent with Erdős-Gallai theorem. If \( b \) contains zero element (that is

<table>
<thead>
<tr>
<th>( n )</th>
<th>BT, s</th>
<th>BT, operation</th>
<th>HT, s</th>
<th>HT, operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>14</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>41</td>
<td>0</td>
<td>43</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>180</td>
<td>0</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>716</td>
<td>0</td>
<td>815</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>2 918</td>
<td>0</td>
<td>3 321</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>11 918</td>
<td>0</td>
<td>13 675</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>48 952</td>
<td>0</td>
<td>56 299</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>201 734</td>
<td>0</td>
<td>233 182</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>831 374</td>
<td>0</td>
<td>964 121</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>3 426 742</td>
<td>0</td>
<td>3 988 542</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>14 107 824</td>
<td>0</td>
<td>16 469 036</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>58 028 152</td>
<td>0</td>
<td>67 929 342</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>238 379 872</td>
<td>0</td>
<td>279 722 127</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>978 194 400</td>
<td>1</td>
<td>1 150 355 240</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>4 009 507 932</td>
<td>3</td>
<td>4 724 364 716</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>16 417 793 698</td>
<td>13</td>
<td>19 379 236 737</td>
</tr>
<tr>
<td>17</td>
<td>26</td>
<td>67 160 771 570</td>
<td>51</td>
<td>79 402 358 497</td>
</tr>
<tr>
<td>18</td>
<td>106</td>
<td>274 490 902 862</td>
<td>196</td>
<td>324 997 910 595</td>
</tr>
<tr>
<td>19</td>
<td>423</td>
<td>1 120 923 466 932</td>
<td>798</td>
<td>1 328 948 863 507</td>
</tr>
<tr>
<td>20</td>
<td>1 627</td>
<td>4 573 895 421 484</td>
<td>3 201</td>
<td>5 429 385 115 097</td>
</tr>
</tbody>
</table>
p < n), then the assertion is the consequence of Havel-Hakimi and Erdős-
Gallai algorithms, since the zero elements do not help in the pairing of the
positive elements, but from the other side they have no own requirement. □

The algorithms based on this corollary are called HAVEL-HAKIMI-ZEROFREE
(HHZ), resp. ERDŐS-GALLAI-ZEROFREE (EGZ).

5.2 Shifting Havel-Hakimi algorithm

The natural algorithmic equivalent of the original Havel-Hakimi theorem is
called HAVEL-HAKIMI SORTING (HHSo), since it requires the sorting of the
reduced input sequence in every round.

But it is possible to design such implementation, in which the reduction of
the degrees is executed saving the monotonity of the sequence. Then we get
HAVEL-HAKIMI-SHIFTING (HHSh) algorithm.

For the pseudocode of this algorithms see [37].

5.3 Parity checking Havel-Hakimi algorithm

It is an interesting idea the join the application of the conditions of Erdős-
Gallai and Havel-Hakimi theorems in such a manner, that we start with the
parity checking of the input sequence, and only then use the recursive Havel-
Hakimi method.

For the pseudocode of the algorithm HAVEL-HAKIMI-PARITY (HHP) see
[37].

5.4 Shortened Erdős-Gallai algorithm (EGSh)

In the case of a regular sequence the maximal value of $H_i$ is $n(n - 1)$, therefore
the inequality (2) certainly holds for $i = n$, therefore it is unnecessary to check.

Even more useful observation is contained in the following assertion due to
Tripathi and Vijay.

Lemma 6 (Tripathi, Vijay [82]) If $n \geq 1$, then an $n$-regular sequence $b =
(b_1 \ldots, b_n)$ is $n$-graphical if and only if

$$H_n \text{ even}$$  \hspace{1cm} (13)

and

$$H_i \leq \min(H_i, i(i - 1)) + \sum_{k=i+1}^{n} \min(i, b_k) \quad (i = 1, 2, \ldots, r),$$  \hspace{1cm} (14)
where
\[ r = \max_{1 \leq s \leq n} (s \mid s(s - 1) < H_s) \] (15)

**Proof.** If \( i(i - 1) \geq H_i \), then the left side of (2) is nonpositive, therefore the inequality holds, so the checking of the inequality is nonnecessary. \( \square \)

The algorithm based on this assertion is called **Erdős-Gallai-Shortened**. For example if the input sequence is \( b = \langle 5, 100 \rangle \), then Erdős-Gallai computes the right side of (2) 99 times, while Erdős-Gallai-Shortened only 6 times.

### 5.5 Jumping Erdős-Gallai algorithm

Contracting the repeated elements a regular sequence \( (b_1, \ldots, b_n) \) can be written in the form \( (b_1^{e_1}, \ldots, b_q^{e_q}) \), where \( b_1 < \cdots < b_q, \ e_1, \ldots, \ e_q \geq 1 \) and \( e_1 + \cdots + e_q = n \). Let \( g_j = e_1 + \cdots + e_j \ (j = 1, \ldots, q) \). The element \( b_i \) is called the **checking points** of the sequence \( b \), if \( i = n \) or \( 1 \leq i \leq n - 1 \) and \( b_i > b_{i+1} \). Then the checking points are \( b_{g_1}, \ldots, b_{g_q} \).

**Theorem 7** (Tripathi, Vijay [82]) An \( n \)-regular sequence \( b = (b_1, \ldots, b_n) \) is \( n \)-graphical if and only if
\[ H_n \text{ even} \] (16)
and
\[ H_{g_i} - g_i(g_i - 1) \leq \sum_{k=g_i+1}^{n} \min(i, b_k) \ (i = 1, \ldots, q). \] (17)

**Proof.** See [82]. \( \square \)

Later in algorithm **Erdős-Gallai-Enumerating** we will exploit, that in the inequality (17) \( g_q \) is always \( n \), therefore it is enough to check the inequality only up to \( i = q - 1 \).

The next program implements a quick version of Erdős-Gallai algorithm, exploiting Corollary 5, Lemma 6 and Lemma 7. In this paper we use the pseudocode style proposed in [17].

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

**Output.** \( L \): logical variable \((L = \text{FALSE} \text{ signalizes, that, } b \text{ is not graphical, while } L = \text{TRUE} \text{ shows, that } b \text{ is graphical})\).

**Working variables.** \( i \) and \( j \): cycle variables;
\( H = (H_0, H_1, \ldots, H_n) \): \( H_i \) is the sum of the first \( i \) elements of \( b \);
C: the degree capacity of the actual tail;
b_{n+1}: auxiliary variable helping to decide, whether b_n is a jumping element.

Erdős-Gallai-Jumping(n, b, H, L)

01 H_1 = b_1                      // lines 01–06: test of parity
02 for i = 2 to n
03     H_i = H_{i-1} + b_i
04 if H_n odd
05     L = False
06     return L
07 b_{n+1} = -1                  // lines 07–20: test of the request of the head
08 i = 1
09 while i ≤ n and i(i - 1) < H_i
10     while b_i == b_{i+1}
11         i = i + 1
12     C = 0
13     for j = i + 1 to n
14         C = C + min(j, b_j)
15     if H_i > i(i - 1) + C
16         L = False
17     return L
18 i = i + 1
19 L = True
20 return L

The running time of EGJ varies between the best Θ(1) and the worst Θ(n^2).

5.6 Linear Erdős-Gallai algorithm

Recently we could improve Erdős-Gallai algorithm [35, 37]. The new algorithm Erdős-Gallai-Linear exploits, that b is monotone. It determines the capacities C_i in constant time. The base of the quick computation is the sequence w(b) containing the weight points w_i of the elements of the input sequence b.

For given sequence b let w(b) = (w_1, \ldots, w_{n-1}), where w_i gives the index of b_k having the maximal index among such elements of b which are greater or equal to i.

Theorem 8 (Iványi, Lucz [35], Iványi, Lucz, Móri, Sötér [37]) If n ≥ 1, then
the $n$-regular sequence $(b_1,\ldots,b_n)$ is $n$-graphical if and only if

$$H_n \text{ is even}$$

(18)

and if $i > w_i$, then

$$H_i \leq i(i - 1) + H_n - H_i,$$

further if $i \leq w_i$, then

$$H_i \leq i(i - 1) + i(w_i - i) + H_n - H_{w_i}.$$

**Proof.** (18) is the same as (1).

During the testing of the elements of $b$ by ERDŐS-GALLAI-LINEAR there are two cases:

- if $i > w_i$, then the contribution $C_i = \sum_{k=i+1}^{n} \min(i,b_k)$ of the tail of $b$ equals to $H_n - H_i$, since the contribution $c_k$ of the element $b_k$ is only $b_k$.

- if $i \leq w_1$, then the contribution of the tail of $b$ consists of contributions of two types: $c_{i+1},\ldots,c_{w_i}$ are equal to $i$, while $c_j = b_j$ for $j = w_i + 1,\ldots,n$.

Therefore in the case $n - 1 \geq i > w_i$ we have

$$C_i = H_n - H_i,$$

(19)

and in the case $1 \leq i \leq w_i$

$$C_i = i(w_i - i) + H_n - H_{w_i}.$$

(20)

□

The following program is based on Theorem 8. 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 sum of the first $i$ elements of the tested $b$;

$b_0$: auxiliary element of the vector $b$

$w = (w_1,\ldots,w_{n-1})$: $w_i$ is the weight point of $b_i$, that is the maximum of the indices of such elements of $b$, which are not smaller than $i$;

$H_0 = 0$: help variable to compute the other elements of the sequence $H$;

$b_0 = n - 1$: help variable to compute the elements of the sequence $w$. 


Theorem 9 (Iványi, Lucz [35], Iványi, Lucz, Móri, Sótér [37]) Algorithm Erdős-Gallai-Linear decides in $\Theta(n)$ time, whether an $n$-regular sequence $b = (b_1, \ldots, b_n)$ is graphical or not.

Proof. Line 1 requires $O(1)$ time, lines 2–3 $\Theta(n)$ time, lines 4–6 $O(1)$ time, line 07 $O(1)$ time, lines 08–12 $O(1)$ time, lines 13–14 $O(n)$ time, lines 15–23 $O(n)$ time and lines 24–25 $O(1)$ time, therefore the total time requirement of the algorithm is $\Theta(n)$.

Since in the case of a graphical sequence all elements of the investigated sequence are to be tested, in the case of RAM model of computations [17] Erdős-Gallai-Linear is asymptotically optimal.
We tested the precise algorithms determining their total running time for all the even sequences. The set of the even sequences is the smallest such set of sequences, whose the cardinality we know exact and explicit formula. The number of \( n \)-bounded sequences \( K(n) \) is also known, but this function grows too quickly when \( n \) grows.

If we would know the average running time of the bounded sequences we would take into account that is sufficient to weight the running times of the regular sequences with the corresponding frequencies. For example a homogeneous sequence consisting of identical elements would get a unit weight since it corresponds to only one bounded sequence, while a rainbow sequence consisting is \( n \) different elements as e.g. the sequence \( n, n-1, \ldots, 1 \) corresponds to \( n! \) different bounded sequences and therefore would get a corresponding weight equal to \( n! \).

We follow two ways of the decreasing of the expected running time of the precise algorithms. The first way is the decreasing of the number of the executable operations. The second way is, that we try to use quick (linear time) preprocessing algorithms for the filtering of the sequences in order to decrease of the part of sequences requiring the relative slow precise algorithms.

For the first type of decrease of the expected running time is the shortening of the sequences and the application of the checking points, while for the the second type are examples the completion of HH algorithm with the parity checking or the completion of the EG algorithm with the binomial and headsplit algorithms.

In this section we investigate the following precise algorithms:
1) \textsc{Havel-Hakimi-Shorting} (HHSo).
2) \textsc{Havel-Hakimi-Shifting} (HHSh).
3) \textsc{Erdős-Gallai} algorithm (EG).
4) \textsc{Erdős-Gallai-Jumping} algorithm (EGJ).
5) \textsc{Erdős-Gallai-Linear} algorithm (EGL).

Figure 5 contains the total number of operations of the algorithms HHSo, HHSh, EG, and EGL required for the testing of all even sequences of length \( n = 1, \ldots, 15 \). The operations necessary to generate the sequences are included.

Comparison of the first two columns shows that algorithm HHSh is much quicker than HHSo, especially if \( n \) increases. Comparison of the third and fourth columns shows that we get substantial decrease of the running time if we have to test the input sequence only in the check points. Finally the comparison of the third and fifth columns demonstrates the advantages of a
Figure 5: Total number of operations as the function of \(n\) for precise algorithms HHSo, HHSh, EG, EGJ, and EGL.

<table>
<thead>
<tr>
<th>(n)</th>
<th>HHSo</th>
<th>HHSh</th>
<th>EG</th>
<th>EGJ</th>
<th>EGL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>15</td>
<td>87</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>61</td>
<td>119</td>
<td>12</td>
<td>37</td>
</tr>
<tr>
<td>3</td>
<td>231</td>
<td>236</td>
<td>267</td>
<td>116</td>
<td>148</td>
</tr>
<tr>
<td>4</td>
<td>1 170</td>
<td>1 052</td>
<td>946</td>
<td>551</td>
<td>585</td>
</tr>
<tr>
<td>5</td>
<td>5 969</td>
<td>4 477</td>
<td>4 000</td>
<td>2 677</td>
<td>2 339</td>
</tr>
<tr>
<td>6</td>
<td>31 121</td>
<td>20 153</td>
<td>18 206</td>
<td>12 068</td>
<td>9 539</td>
</tr>
<tr>
<td>7</td>
<td>157 345</td>
<td>88 548</td>
<td>82 154</td>
<td>54 184</td>
<td>38 984</td>
</tr>
<tr>
<td>8</td>
<td>784 341</td>
<td>393 361</td>
<td>372 363</td>
<td>238 813</td>
<td>160 126</td>
</tr>
<tr>
<td>9</td>
<td>3 628 914</td>
<td>1 726 484</td>
<td>1 666 167</td>
<td>1 666 167</td>
<td>656 575</td>
</tr>
<tr>
<td>10</td>
<td>17 345 700</td>
<td>7 564 112</td>
<td>7 418 447</td>
<td>4 552 276</td>
<td>2 692 240</td>
</tr>
<tr>
<td>11</td>
<td>80 815 538</td>
<td>32 895 244</td>
<td>32 737 155</td>
<td>19 680 986</td>
<td>11 018 710</td>
</tr>
<tr>
<td>12</td>
<td>385 546 527</td>
<td>142 460 352</td>
<td>143 621 072</td>
<td>84 608 529</td>
<td>45 049 862</td>
</tr>
<tr>
<td>13</td>
<td>1 740 003 588</td>
<td>613 739 913</td>
<td>626 050 861</td>
<td>362 141 061</td>
<td>183 917 288</td>
</tr>
<tr>
<td>14</td>
<td>8 066 861 973</td>
<td>2 633 446 908</td>
<td>2 715 026 827</td>
<td>1 543 745 902</td>
<td>750 029 671</td>
</tr>
<tr>
<td>15</td>
<td>36 630 285 216</td>
<td>11 254 655 388</td>
<td>11 717 017 238</td>
<td>6 557 902 712</td>
<td>3 055 289 271</td>
</tr>
</tbody>
</table>

Figure 6: Total and amortized running time of \(Erdős-Gallai-Linear\) in secundum, resp. in the number of executed operations

<table>
<thead>
<tr>
<th>(n)</th>
<th>(E(n))</th>
<th>(T(n)), s</th>
<th>(Op(n))</th>
<th>(T(n)/E(n))/(n), s</th>
<th>(Op(n)/E(n))/(n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>37</td>
<td>0</td>
<td>9.25000000000</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>0</td>
<td>148</td>
<td>0</td>
<td>8.22222222222</td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>0</td>
<td>585</td>
<td>0</td>
<td>7.69736842105</td>
</tr>
<tr>
<td>5</td>
<td>66</td>
<td>0</td>
<td>2 339</td>
<td>0</td>
<td>7.08787878788</td>
</tr>
<tr>
<td>6</td>
<td>236</td>
<td>0</td>
<td>9 539</td>
<td>0</td>
<td>6.73658192090</td>
</tr>
<tr>
<td>7</td>
<td>868</td>
<td>0</td>
<td>38 984</td>
<td>0</td>
<td>6.41606319947</td>
</tr>
<tr>
<td>8</td>
<td>3 235</td>
<td>0</td>
<td>160 126</td>
<td>0</td>
<td>6.18724884080</td>
</tr>
<tr>
<td>9</td>
<td>12 190</td>
<td>0</td>
<td>656 575</td>
<td>0</td>
<td>5.98461432714</td>
</tr>
<tr>
<td>10</td>
<td>46 252</td>
<td>0</td>
<td>2 692 240</td>
<td>0</td>
<td>5.82087748858</td>
</tr>
<tr>
<td>11</td>
<td>176 484</td>
<td>0</td>
<td>11 018 710</td>
<td>0</td>
<td>5.67587385111</td>
</tr>
<tr>
<td>12</td>
<td>676 270</td>
<td>0</td>
<td>45 049 862</td>
<td>0</td>
<td>5.55126752434</td>
</tr>
<tr>
<td>13</td>
<td>2 600 612</td>
<td>0</td>
<td>183 917 288</td>
<td>0</td>
<td>5.44005937537</td>
</tr>
<tr>
<td>14</td>
<td>10 030 008</td>
<td>1</td>
<td>750 029 671</td>
<td>0.000000007121487</td>
<td>5.34132654018</td>
</tr>
<tr>
<td>15</td>
<td>38 781 096</td>
<td>5</td>
<td>3 055 280 271</td>
<td>0.00000008595253</td>
<td>5.25219687963</td>
</tr>
<tr>
<td>16</td>
<td>150 273 315</td>
<td>23</td>
<td>12 434 367 770</td>
<td>0.0000009565903</td>
<td>5.17156346504</td>
</tr>
<tr>
<td>17</td>
<td>583 407 990</td>
<td>79</td>
<td>50 561 399 261</td>
<td>0.0000007965367</td>
<td>5.0979604337</td>
</tr>
<tr>
<td>18</td>
<td>2 268 795 980</td>
<td>297</td>
<td>205 439 740 365</td>
<td>0.0000000727258</td>
<td>5.03056209298</td>
</tr>
</tbody>
</table>

Figure 6: Total and amortized running time of \(Erdős-Gallai-Linear\) in secundum, resp. in the number of executed operations.
On Erdős-Gallai and Havel-Hakimi algorithms

Figure 7: Distribution of the even nongraphical sequences according to the number of tests made by Erdős-Gallai-Jumping to exclude the given sequence for \( n = 3, \ldots, 15 \) vertices.

<table>
<thead>
<tr>
<th>( E(n) - G(n) )</th>
<th>( n/i )</th>
<th>( f_1 )</th>
<th>( f_2 )</th>
<th>( f_3 )</th>
<th>( f_4 )</th>
<th>( f_5 )</th>
<th>( f_6 )</th>
<th>( f_7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>5</td>
<td>33</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>134</td>
<td>6</td>
<td>122</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>526</td>
<td>7</td>
<td>459</td>
<td>65</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2022</td>
<td>8</td>
<td>1709</td>
<td>289</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7829</td>
<td>9</td>
<td>6421</td>
<td>1228</td>
<td>176</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30236</td>
<td>10</td>
<td>24205</td>
<td>4951</td>
<td>1013</td>
<td>67</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>115136</td>
<td>11</td>
<td>91786</td>
<td>19603</td>
<td>5126</td>
<td>610</td>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>454153</td>
<td>12</td>
<td>349502</td>
<td>76414</td>
<td>23755</td>
<td>4274</td>
<td>208</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1764297</td>
<td>13</td>
<td>1336491</td>
<td>296036</td>
<td>104171</td>
<td>25293</td>
<td>2277</td>
<td>29</td>
<td>0</td>
</tr>
<tr>
<td>6863156</td>
<td>14</td>
<td>5128246</td>
<td>1142470</td>
<td>439155</td>
<td>133946</td>
<td>18673</td>
<td>666</td>
<td>0</td>
</tr>
<tr>
<td>26738476</td>
<td>15</td>
<td>19739076</td>
<td>4404813</td>
<td>1803496</td>
<td>655291</td>
<td>127116</td>
<td>8603</td>
<td>81</td>
</tr>
</tbody>
</table>

Figure 6 shows the running time of Erdős-Gallai-Linear in secundum and operation, and also the amortized number of operation/even sequence.

The most interesting data of Figure 6 are in the last column: they show that the number of operations/investigated sequence/length of the investigated sequence is monotone decreasing (see [69]).

Figure 7 shows the distribution of the \( E(n) - G(n) \) even nongraphical sequences according to the number of tests made by Erdős-Gallai-Jumping to exclude the given sequence for \( n = 3, \ldots, 15 \) vertices. \( f_1(n) = f_1 \) gives the frequency of even nongraphical sequences of length \( n \), which required exactly \( i \) round of the test.

These data show, that the maximal number of tests is about \( \frac{n}{2} \) in all lines.

Figure 8 shows the average number of required rounds for the nongraphical, graphical and all even sequences. The data of the column belonging to \( G(n) \) are computed using Lemma 17. It is remarkable that the sequences of the coefficients are monotone decreasing in the last three columns.

Figure 9 presents the distribution of the graphical sequences according to their first element. These data help at the design of the algorithm Erdős-Gallai-Enumerating which computes the new values of \( G(n) \) (in the slicing of the computations belonging to a given value of \( n \)).
A. Iványi, L. Lucz, T. F. Móri, P. Sótér

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
n & E(n) & G(n) & E(n) - G(n) & \text{average of } E(n) - G(n) & \text{average of } G(n) \\
\hline
3 & 6 & 4 & 2 & 0.3333n & 0.8000n \\
4 & 19 & 11 & 8 & 0.3125n & 0.5714n \\
5 & 66 & 31 & 35 & 0.2114n & 0.5555n \\
6 & 236 & 102 & 134 & 0.1967n & 0.5455n \\
7 & 868 & 342 & 526 & 0.1649n & 0.5385n \\
8 & 3233 & 1213 & 2020 & 0.1458n & 0.5333n \\
9 & 12190 & 4363 & 7827 & 0.1337n & 0.5294n \\
10 & 46232 & 16016 & 30216 & 0.1249n & 0.5263n \\
11 & 174484 & 59348 & 115136 & 0.1175n & 0.5238n \\
12 & 676270 & 222117 & 454153 & 0.1085n & 0.5217n \\
13 & 2603612 & 836313 & 1767299 & 0.1035n & 0.5185n \\
14 & 10030008 & 3166852 & 6863156 & 0.0960n & 0.5153n \\
15 & 38761096 & 12042620 & 26718476 & 0.0934n & 0.5127n \\
\hline
\end{array}
\]

Figure 8: Weighted average number of tests made by Erdős-Gallai-Jumping while investigating the even sequences for \( n = 3, \ldots, 15 \)

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
n/b_1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\hline
1 & 1 & 1 & 2 & 4 & 4 & 4 & 4 & 4 & 4 \\
2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
3 & 1 & 1 & 2 & 4 & 4 & 4 & 4 & 4 & 4 \\
4 & 1 & 1 & 1 & 2 & 4 & 4 & 4 & 4 & 4 \\
5 & 1 & 2 & 7 & 10 & 11 & 11 & 11 & 11 & 11 \\
6 & 1 & 3 & 10 & 22 & 35 & 31 & 31 & 31 & 31 \\
7 & 1 & 4 & 14 & 33 & 78 & 110 & 102 & 102 & 102 \\
8 & 1 & 4 & 18 & 54 & 138 & 267 & 389 & 342 & 342 \\
9 & 1 & 4 & 24 & 74 & 223 & 503 & 968 & 1352 & 1213 \\
10 & 1 & 5 & 28 & 104 & 333 & 866 & 1927 & 3496 & 4895 \\
11 & 1 & 5 & 34 & 134 & 479 & 1356 & 3471 & 7221 & 12892 \\
12 & 1 & 6 & 40 & 176 & 661 & 2049 & 5591 & 13270 & 27449 \\
\hline
\end{array}
\]

Figure 9: The distribution of the graphical sequences according to \( b_1 \) for \( n = 1, \ldots, 12 \)

We see in Figure 9 that from \( n = 6 \) the multiplicities increase up to \( n - 2 \), and the last positive value is smaller than the last but one element.
7 Enumerative results

Until now for example Avis and Fukuda [4], Barnes and Savage [5, 6], Burns [13], Erdős and Moser [59], Erdős and Richmond [22], Frank, Savage and Sellers [25], Kleitman and Winston [47], Kleitman and Wang [46], Metropolis and Stein [56], Rodseth et al. [68], Ruskey et al. [69], Stanley [78], Simion [71] and Winston and Kleitman [86] published results connected with the enumeration of degree sequences. Results connected with the number of sequences investigated by us can be found in the books of Sloane és Ploffe [76], further Stanley [79] and in the free online database On-line Encyclopedia of Integer Sequences [73, 74, 75]

It is easy to show, that if \( l, u \) and \( m \) are integers, further \( u \geq l, m \geq 1 \), and \( l \leq b_i \leq u \) for \( i = 1, \ldots, m \), then the number of \((l, u, m)\)-bounded sequences \( a = (a_1, \ldots, a_m) \) of integer numbers \( K(l, u, m) \) is

\[
K(l, u, m) = (u - l + 1)^m.
\]  

(21)

It is known (e.g. see [39, page 65]), that if \( l, u \) and \( m \) are integers, further \( u \geq l \) and \( m \geq 1 \), and \( u \geq b_1 \geq \cdots \geq b_n \geq l \), then the number of \((l, u, m)\)-regular sequences of integer numbers \( R(l, u, m) \) is

\[
R(l, u, m) = \binom{u - l + m}{m}.
\]  

(22)

The following two special cases of (22) are useful in the design of the algorithm Erdős-Gallai-Enumerating.

If \( n \geq 1 \) is an integer, then the number of \( R(0, n-1, n)\)-regular sequences is

\[
R(0, n-1, n) = R(n) = \binom{2n-1}{n}.
\]  

(23)

If \( n \geq 1 \) is an integer, then the number of \( R(1, n-1, n)\)-regular sequences is

\[
R(1, n-1, n) = R_z(n) = \binom{2n-2}{n}.
\]  

(24)

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

Lemma 10 (Ascher [3], Sloane, Ploffe [76]) If

\[\text{Lemma 10 (Ascher [3], Sloane, Ploffe [76]) If}\]
Lemma 11 lemma-En n ≥ 1, then the number of n-even sequences E(n) is
\[ E(n) = \frac{1}{2} \left( \binom{2n-1}{n} + \binom{n-1}{\lfloor n/2 \rfloor} \right). \]  
(25)

**Proof.** See [3, 76]. \[ \Box \]

At the designing and analysis of the results of the simulation experiments is useful, if we know some features of the functions R(n) and E(n).

Lemma 12 If n ≥ 1, then
\[ \frac{R(n+2)}{R(n+1)} > \frac{R(n+1)}{R(n)}, \]  
(26)

\[ \lim_{n \to \infty} \frac{R(n+1)}{R(n)} = 4, \]  
(27)

further
\[ \frac{4^n n \sqrt{\pi n}}{\sqrt{4\pi n}} \left( 1 - \frac{1}{2n} \right) < E(n) < \frac{4^n n \sqrt{\pi n}}{\sqrt{4\pi n}} \left( 1 - \frac{1}{8n + 8} \right). \]  
(28)

**Proof.** On the base of (23) we have
\[ \frac{R(n+2)}{R(n+1)} = \frac{(2n+3)! (n+1)!}{(n+2)! (n+1)! (2n+1)!} = \frac{4n+6}{n+2} = 4 - \frac{2}{n+2}, \]  
(29)

from where we get directly (26) and (27). \[ \Box \]

Using Lemma 13 we can give the precise asymptotic order of growth of E(n).

Lemma 13 If n ≥ 1, then
\[ \frac{E(n+2)}{E(n+1)} > \frac{E(n+1)}{E(n)}, \]  
(30)

\[ \lim_{n \to \infty} \frac{E(n+1)}{E(n)} = 4, \]  
(31)

further
\[ \frac{4^n}{\sqrt{\pi n}} (1 - D_3(n)) < E(n) < \frac{4^n}{\sqrt{\pi n}} (1 - D_4(n)), \]  
(32)

where D_3(n) and D_4(n) are monotone decreasing functions tending to zero.
Proof. The proof is similar to the proof of Lemma 12.

Comparison of (23) and Lemma 13 shows, that the order of growth of numbers of even and odd sequences is the same, but there are more even sequences than odd. Figure 1 contains the values of $R(n)$, $E(n)$ and $E(n)/R(n)$ for $n = 1, \ldots, 37$.

As the next assertion and Figure 1 show, the sequence of the ratios $E(n)/R(n)$ is monotone decreasing and tends to $\frac{1}{2}$.

Corollary 14 If $n \geq 1$, then

$$\frac{E(n+1)}{R(n+1)} < \frac{E(n)}{R(n)}$$  \hspace{1cm} (33)

and

$$\lim_{n \to \infty} \frac{E(n)}{R(n)} = \frac{1}{2}.$$  \hspace{1cm} (34)

Proof. This assertion is a direct consequence of (23) and (25).

The expected value of the number of jumping elements has a substantial influence on the running time of algorithms using the jumping elements. Therefore the following two assertions are useful.

The number of different elements in an $n$-bounded sequence $b$ is called the rainbow number of the sequence, and it will be denoted by $r_n(b)$.

Lemma 15 Let $b$ be a random $n$-bounded sequence. Then the expectation and variance of its rainbow number are as follows.

$$E[r_n(b)] = n \left[ 1 - \left( 1 - \frac{1}{n} \right)^n \right] = n \left( 1 - \frac{1}{e} \right) + O(1),$$  \hspace{1cm} (35)

$$\text{Var}[r_n(b)] = n \left( 1 - \frac{1}{n} \right)^n \left[ 1 - \left( 1 - \frac{1}{n} \right)^n \right]$$

$$+ n(n - 1) \left[ \left( 1 - \frac{2}{n} \right)^n - \left( 1 - \frac{1}{n} \right)^{2n} \right]$$

$$= \frac{n}{e} \left( 1 - \frac{2}{e} \right) + O(1).$$  \hspace{1cm} (36)

Proof. Let $\xi_i$ denote the indicator of the event that number $i$ is not contained in a random $n$-bounded sequence. Then the rainbow number of a random
sequence is \( n - \sum_{i=0}^{n-1} \xi_i \), hence its expectation equals \( n - \sum_{i=0}^{n-1} E[\xi_i] \). Clearly,

\[
E[\xi_i] = \left( 1 - \frac{1}{n} \right)^n
\]

holds independently of \( i \), thus

\[
E[r_n(b)] = n \left[ 1 - \left( 1 - \frac{1}{n} \right)^n \right].
\]

On the other hand,

\[
\text{Var}[r_n(b)] = \text{Var} \left[ \sum_{i=0}^{n-1} \xi_i \right] = \sum_{i=0}^{n-1} \text{Var}[\xi_i] + 2 \sum_{0 \leq i < j \leq n-1} \text{cov}[\xi_i, \xi_j].
\]

Here

\[
\text{Var}[\xi_i] = \left( 1 - \frac{1}{n} \right)^n \left[ 1 - \left( 1 - \frac{1}{n} \right)^n \right],
\]

and

\[
\text{cov}[\xi_i, \xi_j] = E[\xi_i \xi_j] - E[\xi_i]E[\xi_j] = \left( 1 - \frac{2}{n} \right)^n - \left( 1 - \frac{1}{n} \right)^2^n,
\]

implying (36).

We remark that this lemma answers a question of Imre Kátai [40] posed in connection with the speed of computers having interleaved memory and with checking algorithms of some puzzles (e.g. sudoku).

**Lemma 16** The number of \((0, n-1, m)\)-regular sequences composed from \(k\) distinct numbers is

\[
\binom{n}{k} \binom{m-1}{k}, \quad k = 1, \ldots, n.
\]

In other words, the distribution of the rainbow number \( r_n(b) \) of a random \((0, n-1, m)\)-regular sequence \( b \) is hypergeometric with parameters \( n + m - 1, n, \) and \( m \).

**Proof.** The \(k\)-set of distinct elements of the sequence can be selected from \(\{0, 1, \ldots, n - 1\} \) in \( \binom{n}{k} \) ways. Having this values selected we can tell their multiplicities in \( \binom{m-1}{k-1} \) ways. Let us consider the \(k\) blocks of identical elements. The first one starts with \(b_1\), and the starting position of the other \(k-1\) blocks can be selected in \( \binom{m-1}{k-1} \) ways.

From this the expectation and the variance of a random \(n\)-regular sequence follow immediately.
Corollary 17 Let $b$ be a random $n$-regular sequence. Then the expectation and the variance of its rainbow number $r_n(b)$ are as follows:

\[
E[r_n(b)] = \frac{n^2}{2n-1} = \frac{n}{2} + \frac{n}{4n-2} = \frac{n}{2} + O(1), \tag{43}
\]

\[
\text{Var}[r_n(b)] = \frac{n^2(n-1)}{2(2n-1)^2} = \frac{n}{8} + \frac{n}{128n^2 - 128n + 32} = \frac{n}{8} + O(1). \tag{44}
\]

Lemma 18 Let $b$ be a random $n$-regular sequence. Let us write it in the form $b = (b_1^{e_1}, \ldots, b_r^{e_r})$. Then the expected value of the exponents $e_j$ is

\[
E[e_j | r(b) \geq j] = 4 + o(1). \tag{45}
\]

**Proof.** Let $c(n,j)$ denote the number of $n$-regular sequences with rainbow number not less than $j$. By Lemma 16,

\[
c(n,j) = \sum_{k=j}^{n} \binom{n}{k} \binom{n-1}{k-1}. \tag{46}
\]

Let us turn to the number of $n$-regular sequences with rainbow number not less than $j$ and $e_j = \ell$. This is equal to the number of $(0, n-1, n-\ell+1)$-regular sequences containing at least $j$ different numbers, that is,

\[
\sum_{k=j}^{n} \binom{n}{k} \binom{n-\ell}{k-1}. \tag{47}
\]

From this the sum of $e_j$ over all $n$-regular sequences with $e_j > 0$ is equal to

\[
\sum_{\ell=1}^{n-j+1} \ell \sum_{k=j}^{n} \binom{n}{k} \binom{n-\ell}{k-1} = \sum_{k=j}^{n} \binom{n}{k} \sum_{\ell=1}^{n-j+1} \binom{\ell}{1} \binom{n-\ell}{k-1} = \sum_{k=j}^{n} \binom{n}{k} \binom{n+1}{k+1} = c(n+1,j+1). \tag{48}
\]

This can also be seen in a more direct way. Consider an arbitrary $n$-regular sequence with at least $j+1$ blocks, then substitute the elements of the $j+1$st block with the number in the $j$th block (that is, concatenate this two adjacent blocks) and delete one element from the united block; finally, decrease by 1 all elements in the subsequent blocks. In this way one obtains an $n$-regular
sequence with at least $j$ blocks, and it is easy to see that every such sequence is obtained exactly $e_j$ times.

Thus the expectation to be computed is just

$$c(n+1,j+1) \over c(n,j).$$

(49)

Clearly, $c(n,1) = R(0, n-1, n) = \binom{2n-1}{n}$, hence

$$c(n,j) = \binom{2n-1}{n} - \sum_{k=1}^{j-1} \binom{n}{k} \binom{n-1}{k-1} = \binom{2n-1}{n} + O\left(n^{2j-3}\right), \quad (50)$$

as $n \to \infty$. This is asymptotically equal to

$$\binom{2n+1}{n+1} \over \binom{2n-1}{n} = 4n + 2 \over n + 1 = 4 - 2 \over n+1 = 4 + o(1). \quad (51)$$

It is interesting to observe that by (43) the average block length in a random $n$-regular sequence is

$$\frac{1}{r} \sum_{j=1}^{r} e_j = \frac{n}{r(b)} \approx 2 \quad (52)$$

approximately, as $n \to \infty$. This fact could be interpreted as if blocks in the beginning of the sequence were significantly longer. However, fixing $r_n(b) = r$ we find that the lengths of the $r$ blocks are exchangeable random variables with equal expectation $n/r$. At first sight this two facts seem to be in contradiction. The explanation is that exchangeability only holds conditionally. Blocks in the beginning do exist even for smaller rainbow numbers, when the average block length is big, while blocks with large index can only appear when there are many short blocks in the sequence.

The following assertion gives the number of zerofree sequences and the ratio of the numbers of zerofree and regular sequences.

**Corollary 19** The number of the zerofree $n$-regular sequences $R_z(n)$ is

$$R_z(n) = \binom{2n-2}{n-1} \quad (53)$$
and

$$\lim_{n \to \infty} \frac{R_z(n)}{R(n)} = \frac{1}{2}.$$  \hspace{1cm} (54)

Proof. (53) identical with (22), (54) is a direct consequence of (22) and (23).

As the experimental data in Figure 3 show, \( \frac{R_z(n)}{R(n)} \approx \frac{1}{4} \).

The following lemma allows that the algorithm \( \text{ERDŐS-GALLAI ENUMERATING} \) tests only the zero-free even sequences instead of the even sequences.

Lemma 20 If \( n \geq 2 \), then the number of the \( n \)-graphical sequences \( G(n) \) is

$$G(n) = G(n-1) + G_z(n).$$  \hspace{1cm} (55)

Proof. If an \( n \)-graphical sequence \( b \) contains at least one zero, that is \( b_n = 0 \), then \( b' = (b_1, \ldots, b_{n-1}) \) is \((n-1)\)-graphical or not. If \( a = (a_1, \ldots, a_{n-1}) \) is an \((n-1)\)-graphical sequence, then \( a' = (a_1, \ldots, a_{n-1}, 0) \) is \( n \)-graphical.

The set of the \( n \)-graphical sequences \( S \) consists of two subsets: set of zero-free sequences \( S_1 \) and the set of sequences \( S_2 \) containing at least one zero. There is a bijection between the set of the \((n-1)\)-graphical sequences and such \( n \)-graphical sequences, which contain at least one zero. Therefore \( |S| = |S_1| + |S_2| = G_z(n) + G(n-1) \). \hfill \Box

Corollary 21 If \( n \geq 1 \), then

$$G(n) = 1 + \sum_{i=2}^{n} G_z(n).$$  \hspace{1cm} (56)

Proof. Concrete calculation gives \( G(1) = 1 \). Then using (55) and induction we get (56). \hfill \Box

A promising direction of researches connected with the characterization of the function \( G(n) \) is the decomposition of the even integers into members and the investigation, which decompositions represent a graphical sequence [5, 6, 13]. Using this approach Burns proved the following asymptotic bounds in 2007.

Theorem 22 (Burns [13]) There exist such positive constants \( c \) and \( C \), that the following bounds of the function \( G(n) \) is true:

$$\frac{4^n}{cn} < G(n) < \frac{4^n}{(\log n)^c \sqrt{n}}.$$  \hspace{1cm} (57)
Proof. See [13].

This result implies that the asymptotic density of the graphical sequences is zero among the even sequences.

Corollary 23 If \( n \geq 1 \), then there exists a positive constant \( C \) such that

\[
\frac{G(n)}{E(n)} < \frac{1}{(\log_2 n)^C}
\]

and

\[
\lim_{n \to \infty} \frac{G(n)}{E(n)} = 0.
\]

Proof. (58) is a direct consequence of (25) and (58), and (58) implies (59).

As Figure 1 and Figure 3 show, the convergence of the ratio \( G(n)/E(n) \) is relative slow.

8 Number of graphical sequences

Erdős-Gallai-Enumerating algorithm (EGE) [37] generates and tests for given \( n \) every zerofree even sequence. Its input is \( n \) and output is the number of corresponding zerofree graphical sequences \( G_z(n) \).

The algorithm is based on Erdős-Gallai-Linear algorithm. It generates and tests only the zerofree even sequences, that is according to Corollary 5 and Figure 3 about the 25 percent of the \( n \)-regular sequences.

EGE tests the input sequences only in the checking points. Corollary 17 shows that about the half of the elements of the input sequences are check points.

Figure 3 contains data showing that EGE investigates even less than the half of the elements of the input sequences.

Important property of EGE is that it solves in \( O(1) \) expected time

1. the generation of one input sequence;
2. the updating of the vector \( H \);
3. the updating of the vector of checking points;
4. the updating of the vector of the weight points.

We implemented the parallel version of EGE (EGEP). It was run on about 200 PC’s containing about 700 cores. The total running time of EGEP is contained in Figure 10.

The pseudocode of the algorithm see in [37]. The amortized running time of this algorithm for a sequence is \( \Theta(1) \), so the total running time of the whole program is \( O(E(n)) \).
On Erdős-Gallai and Havel-Hakimi algorithms

<table>
<thead>
<tr>
<th>n</th>
<th>running time (in days)</th>
<th>number of slices</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>7</td>
<td>415</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>415</td>
</tr>
<tr>
<td>26</td>
<td>70</td>
<td>435</td>
</tr>
<tr>
<td>27</td>
<td>316</td>
<td>435</td>
</tr>
<tr>
<td>28</td>
<td>1130</td>
<td>2001</td>
</tr>
<tr>
<td>29</td>
<td>6733</td>
<td>15119</td>
</tr>
</tbody>
</table>

Figure 10: The running time of EGEP for $n = 24, \ldots, 29$

9 Summary

In Figure 1 the values of $R(n)$ up to $n = 24$ are the elements of the sequence A001700 of OEIS [73], the values of $E(n)$ up to $n = 23$ are the elements of the sequence A005654 [75] of the OEIS, and in Figure 2 the values of $G(n)$ are up to $n = 23$ are the elements of sequence A0004251-es [74] of OEIS. The remaining values are new [37, 36].

Figure 2 contains the number of graphical sequences $G(n)$ for $n = 1, \ldots, 29$, and also $G(n + 1)/G(n)$ for $n = 1, \ldots, 28$.

The referenced manuscripts, programs and further simulation results can be found at the homepage of the authors, among others at http://compalg.inf.elte.hu/~tony/Kutatas/EGHH/

Acknowledgements

The authors thank Zoltán Király (Eötvös Loránd University, Faculty of Science, Dept. of Computer Science) for his advice concerning the weight points, Antal Sándor and his colleagues (Eötvös Loránd University, Faculty of Informatics), further Ádám Mányoki (TFM World Kereskedelmi és Szolgáltató Kft.) for their help in running of our timeconsuming programs 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. TÁMOP 4.2.1/B-09/1/KMR-2010-0003.
References


On Erdős-Gallai and Havel-Hakimi algorithms


[40] I. Kátaí, Personal communication, Budapest, 2010. ⇒ 256


[57] I. Miklós, Graphs with prescribed degree sequences (Hungarian), Lecture in Alfréd Rényi Institute of Mathematics, 16 November 2009. ⇒231


Received: September 30, 2011 • Revised: November 10, 2011