Acta Univ. Sapientiae, Informatica, **3,** 2 (2011) xx–yy

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@caesar.elte.hu

Tamás 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 [23], Erdős and Gallai in 1960 [16], Hakimi in 1962 [21], Ruskey, Cohen, Eades and Scott 1994-ben [63], Barnes and Savage in 1997 [4], Tripathi, Venugopalan and West in 2010 [76] proposed a method to decide, whether a sequence of nonnegative integers can be the degree sequence of a simple graph (such sequences are called graphical). The running time of their algorithms in worst case is $\Omega(n^2)$. In this paper we propose a new algorithm called EGL (Erdős-Gallai Linear algorithm), whose worst running time is $O(n)$. 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 EGL is asymptotically optimal. As an application of this quick algorithm we could enumerate the different degree sequences of simple graphs for 24, ..., 29 vertices [69].

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

Computing Classification System 1998: G.2.2. [Graph Theory]: Subtopic - Network problems.
Mathematics Subject Classification 2010: 05C85, 68R10
Key words and phrases: simple graphs, prescribed degree sequences, Erdős-Gallai theorem, Havel-Hakimi theorem

typical method of the ranking is the pairwise comparison of the objects, assignment of points to the objests and sorting the objects according to the sums of the numbers of the received points.

For example Landau [46] references to biological, Hakimi [21] to chemical, Kim et al. [39], Newman and Barabási [55] to net-centric, Bozóki, Fülöp, Poesz and Rónyai to economical [8, 9], Liljeros et al. [47] to human applications, while Iványi and Pirzada [26, 27, 33, 59, 61] to applications in sports.

From several popular possibilities we follow the terminology and notations used by Pál Erdős and Tibor Gallai [16].

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 objectos get zero points. In this case the results of the comparisons can be represented using simple graphs.

A pontozás módjától függően sokféle feladat és eredmény van. Ebben a cikkben gráfon olyan véges, irányítás nélküli gráfot értünk, amelyben nem fordulnak elő hurokélek és bármely két csúcsot legfeljebb egy él köt össze – azaz *egyszerű gráfokról* lesz szó. Ez a gráf annak a pontozásnak felel meg, amikor az összehasonlításoknak két lehetséges eredménye van: vagy mindkét objektum egy, vagy pedig mindkettő nulla pontot kap. Az összehasonlítások eredményét ábrázoló gráfban két csúcs pontosan akkor van összekötve, amikor a nekik megfelelő objektumok összehasonlítása során mindkét objektum egy pontot kapott.

A sok népszerű feladat közül azt vizsgáljuk, hogyan dönthető el gyorsan, hogy nemnegatív egészek $n \geq 1$ hosszúságú $b = (b_1, \ldots, b_n)$ sorozatához létezik-e olyan egyszerű gráf, amelynek b a foksorozata. In connection with this problem we remark that the main motivation for studying of this problem is the questin: what is the complexity of deciding whether a sequence is the score sequence of some football tournament [19, 25, 49]

A hasonló feladatokkal kapcsolatban megjegyezzük, hogy mind az irányítatlan, mind pedig az irányított gráfokkal kapcsolatban az utóbbi néhány évben is számos cikk és könyvfejezet jelent meg (például [6, 7, 10, 15, ?, 24, 31, 35, 50, 62, 76, 77, 78], illetve [5, 9, 12, 18, 19, 26, 27, 29, ?, 33, 39, 42, ?, 53, 57, 59, 60]).

Legyen $n \geq m \geq 11$ és $b \geq$. Egész számok $b = (b_1, \ldots, b_n m$ sorozatát (m, n)-*korlátosnak* (röviden: korlátosnak) nevezzük, ha $0 \leq b_i \leq n - 1$ minden $1 \leq i \leq m$ indexre. A $b = (b_1, \ldots, b_m)$ n-korlátos sorozatot (m, n)-*szabályosnak* (röviden: szabályosnak) mondjuk, ha $n - 1 \geq b_1 \geq \cdots \geq b_m \geq 0$. Egy (m, n)-szabályos sorozatot n-*megvalósíthatónak* (röviden: *jónak*) nevezünk, ha létezik olyan G egyszerű gráf, hogy G foksorozata $(b_1, b_2, \ldots, b_m, 0^{n-m})$, ahol 0^{n-m} $n - m$ darab nulla elemet jelent. Ha egy korlátos vagy szabályos sorozat esetén $m = n$, akkor az n paramétert csak egyszer írjuk ki.

A továbbiakban főleg szabályos sorozatokkal foglalkozunk. A definíciókban az alsó és felső korlátok azért szerepelnek, hogy ellenőrző algoritmusainkat megkíméljük a nyilvánvalóan nem jó sorozatok ellenőrzésétől, ezért ezek a megszorítások nem jelentik az általánosság megszorítását.

Cikkünk fő célkitűzése, hogy minél kisebb várható futási idejű algoritmust találjunk annak eldöntésére, hogy előírt b szabályos sorozathoz található-e olyan G egyszerű gráf, melynek foksorozata b. A probléma önmagában is érdekes, azonban számunkra külön motivációt jelent, hogy a feladat részfeladatként jelentkezik, amikor minél kisebb várható futási idejű algoritmust keresünk annak eldöntésére, hogy előírt sorozat lehet-e egy labdarúgó bajnokság pontsorozata [19, 25, 28, **?**, 38, 37, 44, 28].

Melléktermékként bővítettük az EIS [67] adatbázist: egyrészt a benne lévő ismert sorozatok új elemeivel, másrészt konkrét alkalmazásokkal kapcsolatos új sorozatok elemeinek megadásával.

A cikkben egyrészt a vizsgált feladatot pontosan megoldó klasszikus és általunk javasolt új algoritmusokat, másrészt a feladat közelítő megoldását végző, új közelítő algoritmusokat vizsgálunk.

A cikk felépítése a következő. A bevezető első rész után a témakör klasszikus algoritmusait foglaljuk össze. A harmadik rész az új tesztelő algoritmusokat ismerteti, az negyedik részben a közelítő algoritmusok hibáját vizsgáljuk, míg az ötödikben a pontos algoritmusok futási idejét elemezzük.

A cikk eredményeinek egy része a magyar nyelvű [**?**] cikkből származik.

2 Classical precise algorithms

Given sequence $b = (b_1, \dots, b_n)$ of nonegative integers 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

A feladat megoldására az első módszert Vaclav Havel cseh matematikus javasolta 1955-ben [23, 48].

Theorem 1 (Havel [23]) *If* $n \geq 3$*, then the* n*-regular sequence* $b = (b_1, \dots, b_n)$ *is* n*-good if and only if the sequence* $b' = (b_2 - 1, b_3 - 1, \dots, b_{b_1} - 1, b_{b_1+1} -$

$1, b_{b_1+2}, \ldots, b_n)$ *is* $(n-1)$*-good.*

Proof. See [23]. □

Ha ezen tétel alapján írunk egy rekurzív algoritmust, akkor annak futási ideje a RAM számítási modell [14] szerint legrosszabb esetben $\Theta(n^2)$. Érdemes megjegyezni, hogy a tétel bizonyítása konstruktív, és a bizonyításon alapuló algoritmus négyzetes idő alatt nem csak ellenőriz, hanem egy megfelelő gráfot is előállít (feltéve persze, hogy létezik megfelelő egyszerű gráf). Mivel például a teljes gráfnak $\binom{n}{2}$ éle van, a megvalósítást is biztosító algoritmus futási ideje nem lehet kisebb, mint négyzetes.

2.2 Erdős-Gallai algorithm

In 1962 Louis Hakimi [21] published independently the same result, therefore the theoren is called today usually as *Havel-Hakimi theorem*, and the method of reconstruction is called *Havel-Hakimi algorithm*.

Időrendben a következő eredmény Erdős Pál és Gallai Tibor alábbi szükséges és elégséges feltétele [16] volt.

Theorem 2 (Erdős, Gallai, [16]) *Let* $n \geq 3$*. The* n*-regular sequence* $b = (b_1 \ldots, b_n)$ *is* n*-graphical if and only if*

$$\sum_{i=1}^{n} b_i \quad \textit{even} \tag{1}$$

and

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

Proof. See [13, 16, 64, 76]. □

Bár ez a tétel csak ellenőriz, (2) módszeres alkalmazása legrosszabb esetben (például jó sorozatok esetén) $\Theta(n^2)$ időt igényel. A közelmúltban Tripathi et al. [76] publikáltak a tételre konstruktív bizonyítást, amely megvalósítható bemenet esetén $O(n^3)$ idő alatt egy megoldást is előállít.

Erdős-Gallai

3 New precise theorems

3.1 Zerofree algorithms

Mivel a sorozatok végén lévő nullák izolált csúcsokat jelentenek, így azok nem befolyásolják, hogy az adott sorozat jó-e. Ezt a megfigyelést hasznosítja a következő állítás, amelyben p a b sorozat pozitív elemeinek a számát jelöli.

Corollary 3 *If $n \geq 1$, the $(b_1, \ldots, b_n)$ (n, n)-regular sequence is n-graphical if and only if $b_n = 0$ or the sequence $(b_1, \ldots, b_p)$ is (p, p)-graphical.*

Proof. Ha a sorozatnak van pozitív eleme, akkor az állítás a Havel-Hakimi, illetve az Erdős-Gallai következménye, de közvetlenül is adódik: a nullák ugyanis nem segítenek a pozitív fokszámok párosításánál, ugyanakkor nem okoznak önálló igényt sem. □

Az ezen a tulajdonságon alapuló megvalósítást nullamentes Havel-Hakimi (HHN), illetve nullamentes Erdős-Gallai (EGN) algoritmusnak nevezzük.

3.2 Sorting and shifting Havel-Hakimi algorithms

Havel és Hakimi eredeti tételének természetes algoritmikus megfelelőjét HHSonak (Havel-Hakimi Sorting) nevezzük, mert a tétel természetes alkalmazása minden menetben igényli a redukált bemenet rendezését.

A tétel alapján olyan megvalósítás is lehetséges, hogy a fokszámok redukálását a sorozat monotonitását megőrizve végezzük. Ekkor a Havel-Hakimi Shifting algoritmust (HHSh) kapjuk.

The pseudocode of the algithm see in [**?**].

3.3 Parity checking Havel-Hakimi algorithm

Érdekes gondolat az Erdős-Gallai és a Havel-Hakimi feltételek együttes alkalmazása úgy, hogy először b paritását vizsgáljuk, és csak a páros bemenetekre alkalmazzuk a rendszerint négyzetes futási idejű rekurzív ellenőrzést. Ezzel ugyan elveszítjük a rövidített Havel-Hakimi azon jó tulajdonságát, hogy legjobb esetben konstans idő alatt lefut, viszont cserébe megkapjuk azt, hogy a várható futási idő közel 50 százalékkal csökken.

The pseudocode of the algithm see in [**?**].

3.4 Shortened Erdős-Gallai algorithm (EGSh)

H_i maximális értéke szabályos sorozat esetén $n(n-1)$, ezért a 2. Erdős-Gallai tételben szereplő (2) egyenlőtlenség $i = n$ esetén biztosan teljesül, így felesleges ellenőrizni.

Ennél is hasznosabb észrevételt tartalmaz a következő 4. lemma.

Tripathi és Vijay 2003-as cikkében [75] szerepel az az észrevétel, hogy az Erdős-Gallai tételben a (2) egyenlőtlenséget elég csak addig ellenőrizni, amíg $H_i > i(i-1)$ teljesül.

Lemma 4 (Tripathi, Vijay [75]) *Ha* $n \geq 1$, *az* n*-szabályos* $b = (b_1 \dots, b_n)$ *sorozat akkor és csak akkor* n*-jó, ha*

$$H_n \quad \textit{even} \tag{3}$$

and

$$H_i - \min(H_i, i(i-1)) \leq \sum_{k=l+1}^{n} \min(i, b_k) \quad (i = 1, 2, \dots, g), \tag{4}$$

where

$$g = \max_{1 \leq k \leq n} (k \mid k(k-1) < H_k) \tag{5}$$

Proof. Ha $i(i-1) \geq H_i$, akkor (2) bal oldala nempozitív, ezért az egyenlőtlenség biztosan teljesül, így felesleges ellenőrizni. □

Például a $b = (5^{100})$ sorozat esetén (2) jobb oldalát az Erdős-Gallai algoritmus szerint 99-szer, míg a rövidített Erdős-Gallai algoritmus szerint csak 6-szor kell kiszámítani. A javításnak a várható futási időre gyakorolt hatását a **??**. részben vizsgáljuk.

A 4. lemmán alapuló algoritmust rövidített Erdős-Gallai algoritmusnak (EGR) nevezzük.

3.5 Jumping Erdős-Gallai algorithm

Az ismétlődő elemeket összevonva egy szabályos $(b_1, \dots, b_n)$ sorozat $(b_{i_1}^{e_1}, \dots, b_{i_q}^{e_q})$ alakban is felírható, ahol $b_{i_1} < \dots < b_{i_q}$, $e_1, \dots, e_q \geq 1$ és $e_1 + \dots + e_q = n$. Legyen $\sigma_j = e_{e_1} + \dots e_{i_j}$ $(j = 1, \ \dots, q)$.

A b_i elemet a b sorozat *ugró* elemének nevezzük, ha $i = n$ vagy $1 \leq i \leq n-1$ és $b_i > b_{i+1}$. Ekkor az ugró elemek a $b_{\ 1}, b_{\ 2}, \dots, b_{\ q}$ elemek.

Theorem 5 (Tripathi, Vijay [75]) *Az* n*-szabályos* $b = (b_1 \ldots, b_n)$ *sorozat akkor és csak akkor* n*-jó, ha*

$$H_n \quad \textit{páros} \tag{6}$$

and

$$H_{\ i} - \sigma_i(\sigma_i - 1) \le \sum_{k=\ i+1}^{n} \min(i, b_k) \quad (i = 1, \ldots, q). \tag{7}$$

Proof. Lásd [75]. □

Később majd az EGL-All algoritmusban kihasználjuk, hogy a (7) egyenlőtlenségben σ_q mindig n, ezért elég az egyenlőtlenséget $(i = q - 1)$-ig ellenőrizni.

A következő program az Erdős-Gallai algoritmusnak az **??**. és 4. lemma, valamint a 5. tétel alapján gyorsított változatát mutatja be.

Input. n: a csúcsok száma $(n \ge 1)$;
$b = (b_1, b_2, \ldots, b_n)$: n-szabályos sorozat.

Output. L: logikai változó ($L =$ False azt jelzi, hogy b biztosan nem jó, míg $L =$ True azt mutatja, hogy b jó).

Munkaváltozók. i és j: ciklusváltozók;
$H = (H_0, H_1, \ldots, H_n)$: H_i b első i elemének az összege;
p: b pozitív elemeinek a száma;
b_{p+1}: segédváltozó annak eldöntéséhez, hogy b_p ugró elem-e.

Erdős-Gallai-Jumping(n, b, L, γ)

01 $p = n$ ▷Lines 01–03: computing of the number of positive elements
02 **while** $b_p = 0$
03 $\quad$ $p = p - 1$
04 $H_1 = b_1$ ▷ Lines 04–09: test of parity
05 **for** $i = 2$ **to** p
06 $\quad$ $H_i = H_{i-1} + b_i$
07 **if** H_p odd
08 $\quad$ $L =$ False
09 $\quad$ **return** L
10 $b_{p+1} = 0$ ▷ Lines 11–23: test of the request of the head
11 $i = 1$
12 **while** $i \le p$ és $(i - 1) \ge H_i$
13 $\quad$ **if** $b_i == b_{i+1}$
14 $\quad\quad$ $i = i + 1$
15 $\quad\quad$ $E = 0$
16 $\quad\quad$ **for** $j = i + 1$ **to** p
17 $\quad\quad$ $E = E + \min(j, b_j)$

```
        if H_i > i(i − 1) + E
            L = FALSE
            return L
L = TRUE
return L
```

function L = EGJ(n, b) p = n; while b(p) == 0 p = p - 1; end H(1) = b(1); for i = 2:p H(i) = H(i-1) + b(i); end if mod(H(p), 2) == 1 L = false; return end b(p+1) = 0; i = 1; while i¡=p && (i-1) ¡= H(i) if b(i) = b(i+1) E = 0; for j=i+1:p E = E + min(i,b(j)) ; end if H(i) ¿ i*(i-1) + E L = false; return; end end i = i + 1; end L = true; return; end

Input argument ”n” is undefined.

Error in ==¿ EGJ at 3 p = n;

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

3.6 Linear Erdős-Gallai algorithm

Recently we could improve this theorem [?, ?]. The algorithm Erdős-Gallai-Linear exploits, that q is monoton. It determines the C_i capacities in constant time. The base of the quick computation is the sequence $m(q)$ containing indices.

For given sequence q let $m(q) = (m_1, \ldots, m_{n-1})$, where m_i points to the element of q_k having the maximal index among such elements of q which are greater or equal with i.

Theorem 6 (Iványi, Lucz [?], Iványi, Lucz, Sótér [31]) *If $n \geq 1$, the n-regular sequence $(q_1 \ldots, q_n)$ is graphical if and only if*

$$H_n \quad \textit{is even} \tag{8}$$

and if $i > m_i$, then

$$H_i \leq i(i-1) + H_n - H_i, \tag{9}$$

further if $i \leq m_i$, then

$$H_i \leq i(i-1) + i(m_i - i) + H_n - H_{m_i}, \tag{10}$$

Proof. (8) is the same as (1).

During the testing of the elements of q by Erdős-Gallai-Linear there are two cases:

- if $i > m_i$, then the contribution of the tail of q equals to $H_n - H_i$, since the contribution C_k of the element q_k is only q_k.
- if $i \leq m_1$, then the contribution of the tail of q consists of two parts: $C_{i+1}, \ldots, C_{m_i}$ equal to i, while $C_j = b_j$ for $j = m_i + 1, \ldots, n$.

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

$$C_i = i(i-1) + H_n - H_i, \tag{11}$$

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

$$C_i = i(i-1) + i(m_i - i) + H_n - H_{m_i}. \tag{12}$$

□

The following program is based on Theorem **??**. It decides on arbitrary n-regular sequence whether it is graphicakl or not.

Input. n: number of vertices ($n \geq 1$);
$q = (q_1, \ldots, q_n)$: 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 q;
$m = (m_1, \ldots, m_{n-1})$: m_i is the maximum of the indices of such elements of q, which are not smaller than i; $H_0 = 0$: help variable to compute of the other elenments of the sequence H;
$q_0 = n - 1$: help variable to compute the elements of the sequence m.

ERDŐS-GALLAI-LINEAR(n, b)

01 $H_0 = 0$ ▷ Line 01: initialization
02 **for** $i = 1$ **to** n ▷ Lines 02–03: computation of the elemnts of H
03 $\quad H_i = H_{i-1} + b_i$
04 **if** H_n odd ▷ Lines 04–06: test of the parity
05 $\quad L =$ FALSE
06 $\quad$ **return** L
07 $b_0 = n - 1$ ▷ Line 07: initialization of a working variable
08 **for** $i = 1$ **to** n ▷ Lines 08–12: computation of the indices
09 $\quad$ **if** $b_i < b_{i-1}$
10 $\qquad$ **for** $j = b_{i-1}$ **downto** $b_i + 1$

```
11          m_j = i − 1
12       m_{b_i} = i
13 for j = b_n downto 1                        ▷ Lines 13–14: large indices
14    m_j = n
15 for i = 1 to n                  ▷ Lines 15–23: test of the elements of b
16    if i ≤ m_i                ▷ Lines 16–19: test of indices for large m_i's
17       if H_i > i(i − 1) + i(m_i − i) + H_n − H_{m_i}
18          L = FALSE
19          return L
20    if i > m_i                ▷ Lines 20–23: test of indices for small m_i's
21       if H_i > i(i − 1) + H_n − H_i
22          L = FALSE
23          return L
24 L = TRUE            ▷ Lines 24–25: the program ends with the value TRUE
25    return L
```

Theorem 7 (Iványi, Lucz [?], Iványi, Lucz, Sótér [?]) *Algorithm* Erdős-Gallai-Linear *decides in* $O(n)$ *time, whether an* n*-regular sequence* $q = (q_1, \ldots, q_n)$ *is graphical or not.*

Proof. Line 1 requires $O(1)$ time, lines 2–3 $O(n)$ time, lines 4–6 $O(1)$ time, line 07 $O(1)$ time, line 08–09 $O(1)$ time, lines 10–18 $O(n)$ time, lines 19–25 $O(n)$ time and lines 26–27 $O(1)$ time, therefore the total time requirement of the algorithm is $O(n)$. □

Until now the number of good sequences was known only for $n = 1, 2, \ldots, 23$, see *Online Encyclopedia of Integer Sequences* [67]. Using the new and quick algorithms we determined $\gamma(n)$ for $n = 24, 25, \ldots, 28$ too [?, 31]. Figure 1 contains the number of good sequences $\gamma(n)$ for $n = 1, \ldots, 28$, and also $\gamma(n+1)/\gamma(n)$ for $n = 1, \ldots, 28$.

4 Quick Erdős-Gallai algorithm

A következő, a **??** Erdős-Gallai-Quick algoritmus minden n-páros sorozatot megvizsgál, és kimenetként megadja a $\gamma(n)$ értéket. Az algoritmus kihasználja, hogy a a páros sorozatok lexikografikusan csökkenső sorozatában szomszédos sorozatok több lényeges paramétere hasonló, ezért ezek a paraméterek a vizsgált b' sorozatot megelőző b sorozat adott paraméteréből gyorsan meghatározhatóak.

n	$\beta(n)$	$\varphi(n)$	$\gamma(n)$	$\gamma(n+1)/\gamma(n)$
1	**1**	**0**	**1**	2.000000
2	**2**	**2**	**2**	2.000000
3	**4**	**4**	**4**	2.750000
4	**4**	**11**	**11**	2.818182
5	**31**	**31**	**31**	3.290323
6	103	**102**	**102**	3.352941
7	349	344	**342**	3.546784
8	1256	1230	1213	3.595218
9	4577	4468	4361	3.672552
10	17040	16582	16016	3.705544
11	63944	62070	59348	3.742620
12	242218	234596	222117	3.765200
13	922369	891852	836315	3.786674
14	3530534	3409109	3166852	3.802710
15	13563764	13082900	12042620	3.817067
16	52283429	50380684	45967479	3.828918
17	202075949	194550002	176005709	3.839418
18	782879161	753107537	675759564	3.848517
19	3039168331	2921395019	2600672458	3.856630
20	11819351967	11353359464	10029832754	3.863844
21	???	???	38753710486	3.870343
22	???	???	149990133774	3.876212
23	???	???	581393603996	3.881553
24	???	???	2256710139346	3.886431
25	???	??	8770547818956	3.890907
26	???	???	34125389919850	3.895031
27	???	???	132919443189544	3.897978
28	???	???	518232001761434	???

Figure 1: Number of binomial, head halfing and good sequences, further the ratio of the numbers of good sequences for neighbouring values of n.

Például a $h(b')$ metszéspont vagy megegyezik $h(b)$-vel, vagy annál eggyel kisebb.

Ellenőrző pontoknak nevezzük a n-nél kisebb ugró pontokat. Az ellenőrző pontok $C(b')$ listája rendszerint megegyezik a $C(b)$ listával, és legfeljebb a végén változik egy vagy két elem.

Input. n: the number of vertices ($n \geq 4$); (we bound n from below to free the program from the investigation of the special properties of the short programs).

Output. γ: the number of the n-graphical sequences.

Working variables. i and j: cycle variables;
$b = (b_1, \ldots, b_n)$: b_i the ith element of the actually tested even sequence;
$H = (H_0, H_1, \ldots, H_n)$: H_i the sum of the first i elements of the tested sequence;
h: the intersection point of the tested sequence b;
$C(b) = C = (c_1, c_2, \ldots, c_{q-1})$: list of maximal length of the control points;
c: number of control points in the actual list C;
$f = (f_1, \ldots, f_{q-1}$: $f_i = \max(h, \sigma_i)$, the i-th control point.

Erdős-Gallai-Quick(n, γ)

```
01 H_0 = c = 0                              ▷ Lines 01–04: initialization
02 γ = 1
03 h = n − 2
04 f = n − 1
05 for i = 1 to n        ▷ Lines 05–06: generation of the first tested sequence
06      b_i = n − 1
07      H_i = H_i + b_i                        ▷ Line 07: computation of H_i's
08 while b_1 > 0                        ▷ Line 08: is there more sequence?
09         if b_h < h                           ▷ Lines 09–10: updating of h
10            h = h − 1
11         for i = 1 to c            ▷ Lines 11–14: test of the actual sequence
12             f = max(h, σ_i)
13             if H_i > f(f − i) + H_n − H_f
14                 go to 18                 ▷ continuation with a new sequence
15         γ = γ + 1           ▷ Line 15: rise of γ after a graphical sequence
16         go to 18                         ▷ continuation with a new sequence
17 return γ                                    ▷ Line 17: end of the program
18 if        b_n ≥ 2          ▷ Lines 18–69: generation of a new sequence
19           b_n = b_n − 2                          ▷ Lines 18–27: b_n = 2
20           H_n = H_n − 2
21           if   c = 0
22                c = 1
23                C_1 = n − 1
24           else if C_c = n − 1                              ▷ here c > 1
25                   c = c + 1
26                   C_{c+1} = n − 1
27           go to 08
28 else if b_n == 1                                 ▷ Lines 28–43: b_n = 1
29          b_{n−1} = b_{n−1} − 1
```

30 $b_n = b_{n-1}$
31 $H_{n-1} = H_{n-1} - 1$
32 **if** b_n is odd
33 $b_n = b_n - 1$
34 $H_n = H_{n-1} + b_n$
35 **if** $b_n = 0$ és $b_{n-1} > 0$
36 **if** $b_{n-2} = b_{n-1} + 1$
37 **else** $C_c = n - 2$
38 $c = c + 1$
39 $C_c = n - 1$
40 **elseif** $b_{n-2} = 1$
41 $c = c + 1$
42 $C_c = n - 2$
43 **go to** 08
44 **else if** $b_n == 0$ ▷ Lines 44–69: $b_n = 0$
45 $j = n - 1$
46 **while** $b_j == 0$
47 $j = j - 1$
48 **if** $b_j \geq 2$
49 $b_j = b_j - 1$
50 $H_j = H_j - 1$
51 **for** $k = j + 1$ **to** $n - 1$
52 $b_k = b_j$
53 $H_k = H_j + (k - j)b_j$
54 **if** $(n - j)b_j$ is even
55 $b_n = b_n - 1$
56 $H_n = H_{n-1} + b_n$
57 **else** $b_{j-1} = b_{j-1} - 1$ ▷ here $b_j = 1$
58 $H_{j-1} = H_{j-1} - 1$
59 **for** $k = j$ **to** $n - 1$
60 $b_k = b_{j-1}$
61 $H_k = H_{j-1} + (k - j + 1)b_j$
62 **if** $(n - j)b_n$ is odd
63 $b_n = b_n - 1$
64 $H_n = H_{n-1} + b_n$
65 **if** $b_j > 0$ **if** $b_{n-2} = b_{n-1} > b_n$
66 $C_c = n - 1$
67 **else** **else** $b_{n-2} > b_{n-1} = b_n$
68 $C_c = n - 2$

69 **go to** 08

5 Enumerative results

Eddig például Burns [11], Erdős és Moser [54], Kleitman és Winston [41], Winston és Kleitman [79], Stanley [73], Ruskey et al. [63], Barnes és Savage [3, 4], Simion [66], Frank, Savage and Selers [20], Rødseth et al. [62] publikáltak foksorozatok leszámlálására vonatkozó eredményeket. Az általunk vizsgált sorozatok számával kapcsolatos eredmények találhatók Sloane és Ploffe [71], valamint Stanley [**?**] könyvében és az *On-line Encyclopedia of Integer Sequences* című honlapon [68, 69, 70] is.

Lemma 8 *Ha* $n \geq 1$, *akkor az* n*-korlátos sorozatok* $\kappa(n)$*száma*

$$\kappa(n) = n^n, \tag{13}$$

az n*-szabályos sorozatoké pedig*

$$\rho(n) = \binom{2n-1}{n}. \tag{14}$$

Proof. A korlátos sorozatok számára vonatkozó képlet közvetlen adódik abból, hogy mind az n elem n lehetséges értéket vehet fel.

Egy $b = (b_1, \ldots, b_n)$ n-szabályos sorozat esetén legyen $b' = (b'_1, \ldots, b'_n)$, ahol $b'_i = b_i + i - 1$. A lehetséges b és b' sorozatok halmazai között kölcsönösen egyértelmű kapcsolat áll fenn. A különböző b' sorozatok száma pedig annyi, ahány féleképpen a különböző 0, 1, $\ldots$, $2n-2$ számok közül n számot ki tudunk választani. □

A szimulációs vizsgálatok elemzésénél (is) hasznos a szabályos és a páros sorozatok számait megadó függvények tulajdonságainak ismerete.

Lemma 9 *Ha* $n \geq 1$, *akkor*

$$\frac{\rho(n+2)}{\rho(n+1)} > \frac{\rho(n+1)}{\rho(n)}, \tag{15}$$

$$\lim \rho_{n\to\infty} = 4, \tag{16}$$

továbbá

$$\frac{4^n}{\sqrt{4\pi n}}(1 - \frac{1}{2n}) < \rho(n) < \frac{4^n}{\sqrt{4\pi n}}(1 - \frac{1}{8n+8}). \tag{17}$$

Proof. A (14) egyenlőség alapján

$$\frac{\rho(n+2)}{\rho(n+1)} = \frac{(2n+3)!(n+1)n!}{(n+2)!(n+1)!(2n+1)!} = \frac{4n+6}{n+2} = 4 - \frac{2}{n+2}, \tag{18}$$

ahonnan (15 és (21 is közvetlenül adódik. □

Lemma 10 (Ascher, Sloane, Pfoffe [1, 71]) *Ha* $n \geq 1$, *akkor az* n- *páros sorozatok* $\epsilon(n)$ *száma*

$$\epsilon(n) = \frac{1}{2}\left(\binom{2n-1}{n} + \binom{n-1}{\lfloor n \rfloor}\right). \tag{19}$$

Proof. Lásd [71]. □

A 8. és a 11. lemmák egybe vetése mutatja, hogy a páros és páratlan sorozatok számának nagyságrendje megegyezik, azonban több a páros sorozat, mint a páratlan. A 11. lemma alapján pontosan meg tudjuk adni $\epsilon(n)$ aszimptotikus nagyságrendjét.

Lemma 11 *Ha* $n \geq 1$, *akkor*

$$\frac{\epsilon(n+2)}{\epsilon(n+1)} > \frac{\epsilon(n+1)}{\epsilon(n)}, \tag{20}$$

$$\lim \epsilon_{n\to\infty} = 4, \tag{21}$$

továbbá

$$\frac{4^n}{\sqrt{\pi n}}(1-\delta_3(n)) < \epsilon(n) < \frac{4^n}{\sqrt{\pi n}}(1-\delta_4(n)), \tag{22}$$

ahol $\delta_3(n)$ *és* $\delta_4(n)$ *monoton csökkenve nullához tartó sorozatok.*

Proof. A bizonyítás hasonló a 9. lemma bizonyításához. □

Amint azt a következő állítás és a **??**. ábra is mutatja, az $\epsilon(n)/\rho(n)$ hányadosok sorozata monoton csökkenve $\frac{1}{2}$-hez tart.

Corollary 12 *Ha* $n \geq 1$, *akkor*

$$\frac{\epsilon(n+1)}{\rho(n+1)} < \frac{\epsilon(n)}{\rho(n)} \tag{23}$$

és

$$\lim_{n \quad \infty} \frac{\epsilon(n)}{\rho(n)} = \frac{1}{2}. \tag{24}$$

Proof.

□

Az ugró elemek számának várható értéke lényegesen befolyásolja a velük kapcsolatos algoritmusok futási idejét. Ezért hasznos a következő két állítás.

Egy n-korlátos sorozat *szivárvány száma* a sorozatban lévő különböző elemek száma.

Lemma 13 *Ha* $n \geq 1$, *akkor az* n*-korlátos sorozatok átlagos szivárvány száma*

$$\frac{n^2}{2n-1}. \tag{25}$$

Proof. □

Megjegyezzük, hogy ez a lemma választ ad egy Kátai Imre által [34] az átfedéses memóriűjú számítógépekkel és a rejtvények megoldásait ellenőrző algoritmusokkal kapcsolatban felvetett kérdésre.

Lemma 14 *Ha* $n \geq 1$, *akkor az* n*-szabályos sorozatok átlagos szivárvány száma*

$$\frac{n}{2} + \frac{1}{4} + \frac{3}{8n-4}. \tag{26}$$

Proof. Egy n-szabályos sorozatban $1, 2, \ldots, n$ különböző elem lehet. n különböző elem közül k elemet $\binom{n}{k}$ féleképpen választhatunk ki. Ha már kiválasztottunk k elemet, akkor ezeket $\binom{n-1}{k-1}$ módon rendezhetjük, hiszen az azonos elemekből álló részsorozatokat elválasztó $k-1$ helyet a sorozat n eleme közti $n-1$ hely közül kell kiválasztanunk. Figyelembe véve, hogy a **??**. lemma szerint az n-szabályos sorozatok száma $\rho(n) = \binom{2n-1}{n}$, az n-szabályos sorozatok átlagos szivárványszáma

$$\frac{\sum_{k=1}^{n} k\binom{n}{k}\binom{n-1}{k-1}}{\binom{2n-1}{n}}. \tag{27}$$

Ezt a kifejezést átalakítva és felhasználva a hipergeometrikus sorozat várható értékére vonatkozó képletet, az adódik, hogy a keresett várható érték

$$\frac{n^2}{2n-1} = \frac{n}{2} + \frac{1}{4} - \frac{3}{8n-4}. \tag{28}$$

□

Lemma 15 *Ha* $n \geq 1$, *akkor a* b n*-szabályos sorozatot* $b = (b_1^{e_1}, \ldots, b_m^{e_m})$ *alakban felírva az* e_j *kitevők ("futamhosszak") várható értéke konstans ?* $\log n$ *???*

Proof. ???? □

A jó sorozatok $\gamma(n)$ számának jellemzésével kapcsolatos kutatások ígéretes iránya a páros számok pozitív összeadandókra való felbontása, és annak vizsgálata, hogy az ilyen felbontások közül melyek jók [3, 4, 11]. Ezek segítségével sikerült a jó sorozatok számára vonatkozó alábbi aszimptotikus korlátokat bizonyítani.

Lemma 16 (Burns [11]) *Léteznek olyan pozitív* c *és* C *állandók, hogy a jó sorozatok* $\gamma(n)$ *száma a következő korlátok közé esik:*

$$\frac{4^n}{cn} < \gamma(n) < \frac{4^n}{(\log n)^C \sqrt{n}}. \tag{29}$$

6 Testing algorithms

Let $H_i = b_1 + \cdots + b_i$ and $T_i = b_{i+1} + \cdots + b_n$ $(i = 1, \ldots, n)$. Adott i esetén a b sorozat i-hez tartozó *fejének* a $b_1, \ldots, b_i$ elemeket, míg i-hez tartozó *farkának* a $b_{i+1}, \ldots, b_n$ elemeket nevezzük. A sorozatok megvalósíthatóságának vizsgálata során természetes észrevétel, hogy a b fejéhez tartozó fokszámigényt részben belső, részben pedig külső fokszámokkal elégítjük ki.

Először egy "paritásos", majd egy "binomiális", egy "haszontalan", egy "pozitív", egy "fejfelező" és végül egy "farokfelező" szűrő algoritmust mutatunk be.

6.1 Parity test

Első tesztünk az Erdős-Gallai tétel első szükséges feltételén alapul. Nagyon hatékony teszt, mivel mind a korlátos, mind a szabályos sorozatoknak körülbelül fele páratlan sorozat, és a teszt ezekről lineáris idő alatt megállapítja, hogy biztosan nem jó sorozatok.

Lemma 17 *If* $n \geq 1$ *and* b *is an* n*-good sequence, then*

$$H_i \quad \textit{even} \quad (i = 1, \ldots, n). \tag{30}$$

Proof. Every edge of a simple graph adds two to the sum of the degrees. □

A 17. lemmában javasolt egyszerű tesztet a következő algoritmus végzi.

Input. n: a csúcsok száma ($n \geq 1$);
$b = (b_1, b_2, \ldots, b_n)$: szabályos sorozat.

Output. L: logikai változó ($L =$ FALSE azt jelzi, hogy b biztosan nem jó, míg $L =$ TRUE azt mutatja, hogy a teszt *nem tudta eldönteni,* hogy b jó-e).

Working variable:. i: ciklusváltozó;

$S = (S_1, \ldots, S_n)$: S_i az első i fejelem összege. Az S sorozatot globális változónak tekintjük, ezért a további algoritmusoknál *nem* tüntetjük fel bemenő adatként.

```
Parity-Test(n, b, L)
H_1 = b_1
for i = 2 to n
    H_i = H_{i-1} + b_i
if H_n páratlan
   L = False
   return L
L = True
return L
```

Ennek az algoritmusnak a lépésszáma minden esetben $\Theta(n)$.

6.2 Binomial test

Második tesztünk az Erdős-Gallai tétel másik szükséges feltételén alapul. Nevét arról kapta, hogy a fej belső éleinek a számát egy binomiális együttható segítségével becsüljük.

Lemma 18 *If $n \geq 1$ and b is an n-good sequence, then*

$$H_j - j(j-1) \leq T_j \quad (j = 1, \ldots, n-1). \tag{31}$$

Proof. A (31) egyenlőtlenség bal oldalán szereplő összeg a b sorozat fejének fokszámigénye. Ebből levonjuk az igény belső párosításokkal elérhető csökkenésére adható felső korlátot. A jobb oldalon pedig a felhasználható külső fokszámok felső korlátja szerepel. □

A lemmában javasolt tesztet végzi el a következő program.

Input. n: a csúcsok száma ($n \geq 2$);
$b = (b_1, b_2, \ldots, b_n)$: szabályos sorozat.

Output. L: logikai változó ($L = \textsc{False}$ azt jelzi, hogy b biztosan nem jó, míg $L = \textsc{True}$ azt mutatja, hogy a teszt *nem tudta eldönteni,* hogy b jó-e).

Working variables. i, j: ciklusváltozók;

$H = (H_1, \ldots, H_n)$: H_i az első i fejelem összege;

$T = (T_1, \ldots, T_n)$: T_i a b sorozat i-nél nagyobb indexű elemeinek összege.

```
Binomial-Test(n, b, L)
B_1 = 1
for j = 1 to n − 1
    R_j = S_n − S_j
    if  S_j − j(j − 1) > R_j
        L = False
        return L
L = True
return ℒ
```

Ennek az algoritmusnak a futási ideje legrosszabb esetben $\Theta(n)$, míg legjobb esetben csak $\Theta(1)$. Átlagos futási idejének meghatározása további vizsgálatokat igényel.

Mivel (31) jobb oldalán a (2) egyenlet jobb oldalán lévőnél gyengébb feltétel szerepel, ez a lemma csak a vizsgált sorozatok szűrésére alkalmas közelítő algoritmus alapjául szolgálhat.

Az eddigi szimulációs vizsgálatok szerint nagyon hatékony szűrő algoritmus. Aszimptotikus hatékonysága kulcs fontosságú az optimális tesztelő algoritmus szempontjából.

6.3 Unuseful test

A fejen belül megvalósítható párosításnak nem csak a fejben lévő elemek száma, hanem a hozzájuk tartozó fokszámok összege is korlátja. Ezt fejezi ki a következő állítás.

Lemma 19 *If b is an n-good sequence, then*

$$H_i - \min\left(j(j-1), H_i\right) \leq T_i \quad (j = 1, \ldots, n-1). \tag{32}$$

Proof. Az egyenlőtlenség bal oldalán megjelent új követelmény abból adódik, hogy a fej belső döntetlenjeihez legfeljebb annyi szabad fokot tudunk felhasználni, mint a fej elemeinek összege. $\square$

Az új követelmény csak látszólag javítja a tesztelés hatékonyságát, mert ha a fejelemek összege kisebb vagy egyenlő a mellette lévő binomiális együtthatónál, akkor az egyenlőtlenség bal oldala nemnegatív, ami természetesen kisebb vagy egyenlő lesz a farokelemek összegénél. Ezért ennek a feltételnek az ellenőrzését nem is végezzük el.

Ugyanakkor ez a szükséges feltétel egyrészt jó ötletet ad az Erdős-Gallai algoritmus gyorsítására (lásd majd az **??**. részben a **??**. tételt), másrészt további közelítő algoritmusoknak lesz a kiindulási pontja.

6.4 Positive test

Természetesen a farokban lévő nulla elemek nem növelik a farok párosítási potenciálját. Ez az észrevétel lehetővé teszi, hogy a j-edik elemhez tartozó farok párosítási potenciáljára T_j-nél pontosabb becslést adjunk.

Legyen $p = \max(j | b_j > 0)$ a b sorozat pozitív elemeinek száma.

Lemma 20 *If b is an n-good sequence, then*

$$H_j - j(j-1) \geq \min(R_j, j(p-j), (p-j)b_{j+1}). \tag{33}$$

Proof. A minimumban $j(p-j)$ azt jelzi, hogy a fej külső párosítási teljesítményéhet a fej elemei egyenként legfeljebb annyival tudnak hozzájárulni, ahány pozitív elem van a farokban. A $(p-j)b_{j+1}$ elem pedig azt fejezi ki, hogy a farokban lévő $p-j$ elem mindegyikének legfeljebb annyi a hozzájárulása, mint a farokban lévő legnagyobb elem. □

A 20. lemmában javasolt egyszerű tesztet a következő algoritmus végzi.

Input. n: a csúcsok száma ($n \geq 1$);
$b = (b_1, b_2, \ldots, b_n)$: szabályos sorozat.

Output. L: logikai változó (L = FALSE azt jelzi, hogy b biztosan nem jó, míg L = TRUE azt mutatja, hogy a teszt *nem tudta eldönteni,* hogy b jó-e).

Working variabke:. i: ciklusváltozó;
p: a bemenetben lévő pozitív elemek száma
H: Globális változó.

POSITIVE-TEST(n, b, L)
$p = n$
while $b_p = 0$
    $p = p - 1$
for $i = 1$ **to** n
    if $H_i - j(j-1) > \min(R_j, jb_j, (p-j)b_{j+1}$
    L = FALSE
    return L

07 L = TRUE
08 **return** $\mathcal{L}$

Ennek az algoritmusnak a futási ideje a legjobb $\Theta(1)$ és a legrosszabb $\Theta(n)$ között változik.

6.5 Halfing of the head

We get a better estimation of the inner capacity of the head, than the binomial coefficient gives in (31).

Lényeges körülmény, hogy a 18. és 19. lemmák használata *lineáris* időt igényel (míg a korábbi két tétel négyzetes futási időt). Az Erdős-Gallai tétel javítása viszont csak a másodfokú futási idő együtthatóját csökkenti, nagyságrendjét azonban nem.

6.6 Composite test

A továbbiakban a futási idő és a hatékonyság együttes optimalizálására törekszünk.

7 Properties of the approximate algorithms

A közelítő algoritmusok hatékonyságát a szabályos sorozatok tesztelésével vizsgáljuk. Adott A algoritmusnak az n hosszúságú szabályos sorozatokra vonatkozó hatékonyságát az algoritmus által elfogadott (ki nem zárt) és jó n hosszúságú sorozatok számának hányadosával jellemezzük. Ezt a hányadost $R_A(n)$-nel jelöljük és az A algoritmus n hosszúságú sorozatokra vonatkozó hibájának nevezzük.

A következő közelítő algoritmusokat vizsgáljuk:
1) PÁROSSÁG-TESZT
2) BINOMIÁLIS-TESZT
3) POZITÍV-TESZT
4) FEJFELEZŐ-TESZT
5) FAROKFELEZŐ-TESZT

Ha $n = 2$, akkor (??) szerint $\rho(n) = \binom{3}{2} = 3$ szabályos sorozat van: $(1, 1)$, $(1, 0)$ és $(0, 0)$. Ha egy szabályos sorozat elemeinek összege páros, akkor *páros sorozatnak* nevük. Az n hosszúságú páros sorozatok számát $\pi(n)$-nel jelöljük. Ezzel a jelöléssel $\pi(2) = 2$. A BINOMIÁLIS-TESZT által elfogadott, n hosszúságú sorozatok számát $\beta(n)$-nel jelölve $\beta(2) = 2$. Az n hosszúságú helyreállítható sorozatok számát jelöljük $\gamma(n)$-nel. Ekkor $\gamma(2) = 2$ és a BINOMIÁLIS-TESZT hibája (hatékonysága) $R_{BT}(2) = 2/2 = 1$.

Ha $n = 3$, akkor a szabályos sorozatok száma $\rho(3) = 10$. Ezek közül a (2,2,2), (2,2,0), (2,1,1), (2,0,0), (1,1,0) és (0,0,0) páros, azaz $\pi(3) = 6$. Ezek közül a Binomiális-teszt kizárja a (2,2,0) és (2,0,0) sorozatokat, így $\beta(3) = 4$. A megmaradt 4 sorozat helyreállítható, így $\gamma(3) = 4$.

Ha $n = 4$, akkor a szabályos sorozatok száma $\rho(4) = 35$. Ezek közül 19 a páros, és a következő 11 a helyreállítható: (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) és (0,0,0,0). A 16 páros sorozat közül Binomiális-teszt is kizárja azt az öt sorozatot, amelyeket az Erdős-Gallai, így $\beta(4) = \gamma(4) = 11$ és $R_{BT}(4) = 11/11 = 1$.

Ezek szerint a legfeljebb 4 hosszúságú sorozatokra nézve a Binomiális-teszt hibátlanul kiszűri a nem jó sorozatokat. A 2. táblázat ezeket az adatokat foglalja össze.

A táblázat is mutatja, hogy míg $n \leq 4$ esetén $\beta(n) = \gamma(n)$, azaz a Binomiális-teszt ugyanannyi sorozatot fogad el, mint a pontos algoritmusok, $n > 4$ esetén egyre nő a Binomiális-teszt hibája: $n = 5$ esetén még csak egyetlen páros sorozatról nem ismeri fel, hogy rossz, $n = 6$ esetén már hatszor hibázik.

Eddig például Erdős és Moser [54], Kleitman és Winston [41], Winston és Kleitman [79], Stanley [73], Ruskey et al. [63], Barnes és Savage [3, 4], Simion [66], Frank, Savage and Selers [20], Rødseth et al. [62] publikáltak foksorozatok számával kapcsolatos eredményeket. Az általunk vizsgált sorozatok számával kapcsolatos eredmények találhatók az *On-line Encyclopedia of Integer Sequences* című honlapon [68, 69, 70] is.

A 3 táblázatban $\rho(n)$ értéke $n = 24$-ig az EIS A001700 sorozata [68], $\epsilon(n)$ értéke $n = 23$-ig az EIS A005654 sorozata [70], $\gamma(n)$ értéke pedig $n = 23$-ig az EIC A0004251-es sorozata [69] sorozata. A többi értéket mi határoztuk meg.

Eddig első sorban soros algoritmusokat alkalmaztunk, de vannak párhuzamos eredmények is [56, 65, 72].

8 Running time of the classical testing algorithms

A tesztek elemzésénél a korlátos, illetve a szabályos sorozatokat vettük alapul. A szabályos sorozatok halmaza a legkisebb olyan halmaz, melynek elemszámát egyszerű explicit képlettel meg tudtuk adni. Az $n - 1 \geq a_i \geq 1$ és az $n - 1 \geq a_i \geq 0$ feltételeknek eleget tevő n-*korlátos sorozatok* halmazának elemszámát is könnyű megadni, de ezen halmazok elemszáma túl gyorsan nő n növekedtével. A monoton sorozatok elemzéséhez szerencsére nem kell *minden* korlátos sorozatot előállítani: elegendő a szabályos sorozatokat előállítani, és a rájuk vo-

n	$\rho(n)$	$\epsilon(n)$	$\beta(n)$	$\pi(n)$	$\varphi(n)$	$\tau(n)$	$\gamma(n)$
1	1	1	1	1	1	1	1
2	3	2	2	2	2	2	2
3	10	6	4	4	4	4	4
4	35	19	11	11	11	11	11
5	126	66	31	31	31	31	31
6	462	236	103	102	???		102
7	1716	868	349	342	???		342
8	6435	3235	1256	1213	???		1213
9	24310	12190	4577	4361	???		4361
10	92378	46252	17040	16016	???		16016
11	352716	176484	63944	59348	???		59348
12	1352078	676270	242218	222117	???		222117
13	5200300	2600612	922239	836315	???		836315
14	20058300	10030008	???????	3530534	???		3166852
15	77558760	38781096	???	???	???		12042620
16	300540195	150273315	???	???	???		45967479
17	1166803110	583407990	???	???	???		675759564
18	4537567650	2268795980	???	???	???		675759564
19	17672631900	8836340260	???	???	???		2600672458
20	68923264410	34461678394	???	???	???		10029832754
21	269128937220	134564560988	???	???	???		38753710486
22	1052049481860	526024917288	???	???	???		149990133774
23	4116715363800	2058358034616	???	???	???		581393603996
24	16123801841550	???	???	???	???		???
25	**63205303218876**	???	???	???	??		??

Figure 2: The number of regular, even, binomial, positive, tailtested and good sequences.

natkozó hatékonysági jellemzőket a nekik megfelelő gyakoriságokkal súlyozni. Például egy azonos elemekből álló *homogén* szabályos sorozatnak egyetlen korlátos sorozat felel meg, míg a különböző elemekből álló $(n, n-1, \ldots)$ ”szivárvány” sorozatnak $n!$ különböző korlátos sorozat felel meg.

Az alapvető pontos algoritmusokat kétféle módon próbáljuk gyorsítani (azaz várható futási idejüket csökkenteni). Az egyik út, hogy csökkentjük az általuk

n	$\epsilon(n)$	$100\beta(n)/\epsilon(n)/$	$100\pi(n)/\epsilon(n)$	$100\varphi(n)/\epsilon(n)$	$100\gamma(n)/\epsilon(n)$
2	3	2	2	2	
3	10	6	4	4	
4	35	19	11	11	
5	126	66	32	31	$32/31 = 1,032$
6	462	236	108	102	$108/102 = 1,059$
7	1716	868	376	342	$376/342 = 1,099$
8	6435	3235	???	1213	??
9	24310	12190	???	4361	??
10	92378	46252	???	16016	??
11	352716	176484	???	59348	??
12	????	676270	???	222117	??
13	????	2600612	???	836315	??
14	????	10030008	???	3166852	??
15	????	38781096	???	12042620	??
16	????	150273315	???	45967479	??
17	????	583407990	???	675759564	??
18	????	2268795980	???	675759564	??
19	????	8836340260	???	2600672458	??
20	????	34461678394	???	10029832754	??
21	?????	134564560988	???	38753710486	??
22	????	526024917288	???	149990133774	??
23	????	2058358034616	???	581393603996	??
24	????	???	???	???	??
25	????	???	???	???	??

Figure 3: Characteristic relativ properties of the approximate algorithms.

elvégzendő ellenőrzések számát. A másik út pedig az, hogy gyors (lineáris) előtesztekkel igyekszünk a rossz sorozatok jelentős részét kiszűrni, hogy csak a lehetséges bemenetek kis hányadánál legyen szükség a viszonylag lassú, de pontos alapalgoritmusokra.

Az első típusú javításra példa a rövidítés és az ugrás, a másikra példa a Havel-Hakimi algoritmus kiegészítése a paritás vizsgálatával és az Erdős-Gallai algoritmus kiegészítése a binomiális algoritmussal, valamint a farok és a fej elemzésével.

A következő pontos algoritmusokat vizsgáljuk:
1) HH1 (HHSo): Sorting Havel-Hakimi algorithm.
2) HH2 (HHSh): Shifting Havel-Hakimi algorithm
3) HH3 (HHP): Parity checkig Havel-Hakimi algorithm.
4) HH4: (HHA): Approximate Havel-Hakimi algorithm.
5) EG1: (EG):Erdős-Gallai algorithm.
6) EG2: (EGS): Shortened Erdős-Gallai algorithm.
7) EG3: (EGJ): Jumping Erdős-Gallai algorithm.
8) EG4: Approximate Erdős-Gallai algorithm.

9 Running time of EG-Linear

A következő program a **??**. tétel alapján adott n-re az összes n-páros sorozatot előállítja és teszteli, hogy jók-e. A teljes program futási ideje minden sorozatra $O(n)$, így a teljes futási idő $O(n\epsilon(n))$.

Input. n: a csúcsok száma ($n \geq 1$).

Outputt. γ: az n-jó sorozatok száma.

Working variables. i és j: ciklusváltozók;
$b = (b_1, \dots, b_n)$: b_i az éppen tesztelt páros sorozat i-edik eleme; $H = (H_0, H_1, \dots, H_n)$: H_i az éppen tesztelt b első i elemének az összege; h: a vizsgált b sorozat metszéspontja.

EG-Linear(n, γ)

01 $H_0 = 0$ ▷ 01–03. sor: kezdeti értékek beállítása
02 $\gamma = 1$
03 $h = n - 2$
04 **for** $i = 1$ **to** n ▷ 04–05. sor: első vizsgált sorozat előállítása
05 $\quad b_i = n - 1$
06 $\quad H_i = H_i + b_i$ ▷ 06. sor: H_i-k előállítása
07 **while** $b_1 > 0$ ▷ 07. sor: van-e még vizsgálandó sorozat?
08 **if** $b_h < h$ ▷ 08–09. sor: h frissítése
09 $\quad h = h - 1$
10 $\quad$ **for** $i = 1$ **to** $h - 1$ ▷ 10–17. sor: aktuális sorozat tesztelése
11 $\quad\quad$ **if** $H_i > i(i-1) + i(h-i) + H_n - H_h$
12 $\quad\quad\quad$ **go to** 18 ▷ áttérés új sorozatra
13 $\quad$ **for** $i = h$ **to** $n - 1$
13 $\quad\quad$ **if** $H_i > i(i-1) + H_n - H_i$
14 $\quad\quad\quad$ **go to** 18 ▷ áttérés új sorozatra

```
15    γ = γ + 1
16    go to 18
17 return γ                                  ▷ 17. sor: program befejezése
18 if b_n ≥ 2                                 ▷ 18–51. sor: új sorozat előállítása
19    b_n = b_n − 2
20    H_n = H_n − 2
21    go to 07
22 if b_n == 1
23    b_{n−1} = b_{n−1} − 1
24    b_n = b_{n−1}
25    H_{n−1} = H_{n−1} − 1
26    if b_n páratlan
27       b_n = b_n − 1
28    H_n = H_{n−1} + b_n
29    go to 07
30 if b_n == 0
31    j = n − 1
32    while b_j == 0
33            j = j − 1
34     if  b_j ≥ 2
35         b_j = b_j − 1
36         H_j = H_j − 1
37         for k = j + 1 to n − 1
38              b_k = b_j
39              H_k = H_j + (k − j)b_j
40         if (n − j)b_j páros
41            b_n = b_n − 1
42         H_n = H_{n−1} + b_n
43    else b_{j−1} = b_{j−1} − 1                         ▷ itt b_j = 1
44          H_{j−1} = H_{j−1} − 1
45          for k = j to n − 1
46               b_k = b_{j−1}
47               H_k = H_{j−1} + (k − j + 1)b_j
48               if (n − jb_n páratlan
49                  b_n = b_n − 1
50               H_n = H_{n−1} + b_n
51    go to 07
```

n	(n)	(n), mp	$\mathrm{Op}(n)$	$(n)/\ (n)/n$, mp	$\mathrm{Op}(n)/\ (n)/n$
1	???	???	???	???	???
2	???	???	???	???	???
3	???	???	???	???	???
4	???	???	???	???	???
5	???	???	???	???	???
6	???	???	???	???	???
7	???	???	???	???	???
8	???	???	???	???	???
9	???	???	???	???	???
10	???	???	???	???	???
11	???	???	???	???	???
12	???	???	???	???	???
13	???	???	???	???	???
14	???	???	???	???	???
15	???	???	???	???	???
16	???	???	???	???	???
17	???	???	???	???	???
18	???	???	???	???	???
19	???	???	???	???	???
20	???	???	???	???	???
21	???	???	???	???	???
22	???	???	???	???	???
33	???	???	???	???	???
24	???	???	???	???	???
25	???	???	???	???	???

Figure 4: Total and amortized running time of EG-LINEAR in secundum, resp. in the number of executed operations műveletszámban kifejezve.

Acknowledgements. The authors thank Tamás F. Móri [**?**] for proving Lemma 13 and 14, Ádám Mányoki (TFM World Kereskedelmi és Szolgáltató Kft.) for the 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

[1] M. Ascher, Mu torere: an analysis of a Maori game. *Math. Mag.* **60,** (2) (1987) 90–100. ⇒15

[2] D. Avis, K. Fukuda, Reverse search for enumeration, *Discrete Appl. Math.* **2** (1993) 21–46. ⇒

[3] T. M. Barnes, C. D. Savage, A recurrence for counting graphical partitions, *Electron. J. Combin.* **2,** (1995), Research Paper 11, 10 pages (electronic). ⇒14, 17, 22

[4] T. M. Barnes, C. D. Savage, Efficient generation of graphical partitions. *Discrete Appl. Math.* **78,** (1–3) (1997) 17–26. ⇒1, 14, 17, 22

[5] L. B. Beasley, D. E. Brown, K. B. Reid, Extending partial tournaments. *Math. Comput. Modelling* **50,** (1) (2009) 287–291. ⇒2

[6] S. Bereg, H. Ito, Transforming graphs with the same degree sequence. In: (ed. by H. Ito et al.) *The Kyoto Int. Conf. on Computational Geometry and Graph Theory,* LNCS **4535.** Springer-Verlag, Berlin, Heidelberg. 2008. pp. 25–32. ⇒2

[7] N. Bodei, Degree sequences of graphs, Master thesis, Dept. of Operation Research of Eötvös Loránd University (supervisor A. Frank), Budapest, 2010. ⇒2

[8] S. Bozóki S., J. Fülöp, A. Poesz: On pairwise comparison matrices that can be made consistent by the modification of a few elements. *CEJOR Cent. Eur. J. Oper. Res.* **19** (2011) 157–175. ⇒2

[9] Bozóki S.: J. Fülöp, L. Rónyai: On optimal completion of incomplete pairwise comparison matrices. *Math. Comput. Modelling* **52,** (2010) 318–333. ⇒2

[10] A. R. Brualdi, K. Kiernan: Landau's and Rado's theorems and partial tournaments, *Electron. J. Combin.* **16(#N2),** (2009) (6 pp). ⇒2

[11] Burns, J. M.: *The number of degree sequences.* PhD Dissertation, MIT, 2007. ⇒14, 17

[12] A. N. Busch, G. Chen, M. S. Jacobson, Transitive partitions in realizations of tournament score sequences. *J. Graph Theory* **64(1),** (2010), 52–62. ⇒2

[13] S. A. Coudum, A simple proof of the Erdős-Gallai theorem on graph sequences. *Bull. Austral. Math. Soc.* **33,** (1986) 67–70. ⇒4

[14] T. H. Cormen, Ch. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms.* Third edition, The MIT Press/McGraw Hill, Cambridge/New York, 2009. ⇒4

[15] C. I. Del Genio, Z. Kim, H., Toroczkai, K. E. Bassler, Efficient and exact sampling of simple graphs with given arbitrary degree sequence. PLoS ONE **5(4)**, e10012 (2010). ⇒2

[16] P. Erdős, T. Gallai, Graphs with vertices having prescribed degrees (Hungarian), *Mat. Lapok* **11,** (1960) 264–274. ⇒1, 2, 4

[17] P. Erdős, L. B. Richmond, On graphical partitions. Combinatorica **13,** (1) (1993) 57–63. ⇒

[18] P. L. Erdős, I. Miklós, I., Z. Toroczkai, *A simple Havel-Hakimi type algorithm to realize graphical degree sequences of directed graphs.* Electron. J. Combin. **17(1),** (2010) R66 (10 pp). ⇒2

[19] A. Frank, *Connections in Combinatorial Optimization.* Oxford University Press, Oxford, 2011. ⇒2, 3

[20] D. A. Frank, C. D. Savage, J. A. Sellers, On the number of graphical forest partitions. *Ars Combin.* **65** (2002) 33–37. ⇒14, 22

[21] S. L. Hakimi, On the realizability of a set of integers as degrees of the vertices of a simple graph. J. SIAM Appl. Math. **10,** (1962) 496–506. ⇒ 1, 2, 4

[22] S. L. Hakimi, On realizability of a set of integers as degrees of the vertices of a linear graph II. Uniqueness. SIAM J. Appl. Math., **11,** (1) (1963) 135–147. ⇒

[23] V. Havel, A remark on the existence of finite graphs (Czech); *Časopis Pěst. Mat.* **80** (1955), 477–480. ⇒1, 3, 4

[24] P. Hell, D. Kirkpatrick, Linear-time certifying algorithms for near-graphical sequences. Discrete Math. **309(18),** (2009) 5703–5713. ⇒2

[25] A. Iványi, Testing of score sequences (Hungarian). In: (szerk. ???) *XXV. Hungarian Conference of Operation Research* (Debrecen, 2001. október 17–20), 53–53. ⇒2, 3

[26] A. Iványi, Reconstruction of complete interval tournaments. Acta Univ. Sapientiae, Informatica, **1,** (1) (2009) 71–88. ⇒2

[27] A. Iványi, Reconstruction of complete interval tournaments. II. *Acta Univ. Sapientiae, Mathematica*, **2(,** (1) (2010) 47–71. ⇒2

[28] A. Iványi, Deciding the validity of the score sequence of a soccer tournament. In: *Open problems of the Egerváry Research Group,* Budapest, 2011. ⇒3

[29] A. Iványi, Directed graphs with prescribed score sequences. In: *The 7th Hungarian-Japanese Symposium on Discrete Mathematics and Applications* (Kyoto, May 31 - June 3, 2011), 114–123. ⇒2

[30] A. Iványi, L. Lucz, Erdős-Gallai test in linear time. *Combinatorica* (submitted). ⇒

[31] A. Iványi, L. Lucz, P. Sótér, Linear Erdős-Gallai test (Hungarian). *Alk. Mat. Lapok* (submitted). ⇒2, 8, 10

[32] A. Iványi, L. Lucz, P. Sótér, Parallel Erdős-Gallai algorithm. *Central European J. Operation Research* (submitted). ⇒

[33] A. Iványi, S. Pirzada, Comparison based ranking, in: *Algorithms of Informatics, Vol. 3,* ed. A. Iványi, AnTonCom, Budapest 2011, 1262–1311. ⇒2

[34] I. Kátai, Personal communication, Budapest, 2010. ⇒16

[35] K. K. Kayibi, M. A. Khan, S. Pirzada, A. Iványi, Random sampling of minimally cyclic digraphs with given imbalance sequence. *Acta Univ. Sapientiae, Math.* (submitted). ⇒2

[36] K. Kern, D. Paulusma, The new FIFA rules are hard: complexity aspects of sport competitions. *Discrete Appl. Math.* **108,** (3) (2001) 317–323. ⇒3

[37] K. Kern, D. Paulusma, The computational complexity of the elimination problem in generalized sports competitions, *Discrete Optimization* **1** (2004) 205–214. ⇒3

[38] K. Kern, D. Paulusma, The computational complexity of the elimination problem in generalized sports competitions. ⇒3

[39] H. Kim, Z. Toroczkai, I. Miklós, P. L. Erdős, l. A. Székely, Degree-based graph construction. *J. Physics: Math. Theor. A* **42,** (39) (2009) 392–401. ⇒2

[40] D. J. Kleitman, D. L. Wang, Algorithms for constructing graphs and digraphs with given valencies and factors. *Discrete Math.* **6** (1973) 79–88. ⇒

[41] D. J. Kleitman, K. J. Winston, Forests and score vectors, *Combinatorica* **1,** (1) (1981) 49–54. ⇒14, 22

[42] D. E. Knuth, *The Art of Computer Programming. Volume 4A, Combinatorial Algorithms.* Addison–Wesley, Upper Saddle River, 2011. ⇒2

[43] A. Kohnert, Dominance order and graphical partitions. *Elec. J. Comb.* **11(1),** (2004) No. 4. 17 pp. ⇒

[44] G. Zs. Kovács, N. Pataki, Rangsorolási algoritmusok elemzése. Scientific student paper, Faculty of Sciences of Eötvös Loránd University, Budapest, 2002. http://compalg.inf.elte.hu/~tony/Oktatas/TDK/. ⇒3

[45] M. D. LaMar, Algorithms for realizing degree sequences of directed graphs. arXiv-0906:0343ve [math.CO], 7 June 2010. ⇒

[46] H. G. Landau, On dominance relations and the structure of animal societies. III. The condition for a score sequence. *Bull. Math. Biophys.* **15,** (1953) 143–148. ⇒2

[47] F. Liljeros, C. R. Edling, L. Amaral, H. Stanley, Y. Åberg, The web of human sexual contacts. *Nature* **411,** (2001) 907–908. ⇒2

[48] L. Lovász, *Combinatorial Problems and Exercises* (corrected version of the second edition). AMS Chelsea Publishing, Boston, 2007. Magyarul: *Kombinatorikai problémák és feladatok.* Typotex, Budapest, 1999. ⇒3

[49] L. Lucz, A. Iványi, P. Sótér, S. Pirzada, Testing and enumeration of football sequences. ⇒2

[50] D. Meierling, L. Volkmann, L.: A remark on degree sequences of multigraphs. *Math. Methods Oper. Res.* **69(2),** (2009) 369–374. ⇒2

[51] N. Metropolis, P. R. Stein, The enumeration of graphical partitions. *European J. Comb.* **1(2),** (1980) 139–153. ⇒

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

[53] I. Miklós, P. L. Erdős, L. Soukup, A remark on degree sequences of multigraphs. (2011) (submitted). ⇒ 2

[54] J. W. Moon, *Topics on Tournaments.* Holt, Rinehart, and Winston, New York, 1968. ⇒ 14, 22

[55] M. E. J. Newman, A. L. Barabási, *The Structure and Dynamics of Networks.* Princeton University Press, Princeton, NJ. 2006. ⇒ 2

[56] G. Pécsy, L. Szűcs, Parallel verification and enumeration of tournaments. *Stud. Univ. Babeş-Bolyai, Inform.* **45(2),** (200) 11–26. ⇒ 22

[57] S. Pirzada, *Graph Theory.* Orient Blackswan (to appear). ⇒ 2

[58] S. Pirzada, A. Iványi, Imbalances in digraphs. In: *Abstracts of Joint Conference on Mathematics and Computer Science* (Siófok, February 2012, submitted). ⇒

[59] S. Pirzada, A. Iványi, M. A. Khan, Score sets and kings. In: *Algorithms of Informatics*, Vol. 3, ed. by A. Iványi. AnTonCom, Budapest 2011, 1451–1490. ⇒ 2

[60] S. Pirzada, T. A. Naikoo, T. A., U. T. Samee, U. T., A. Iványi, Imbalances in directed multigraphs. *Acta Univ. Sapientiae, Informatica* **2(1),** (2010) 47–71. ⇒ 2

[61] S. Pirzada, Iványi, Zhou, On k-hypertournament losing scores, *Acta Univ. Sapientiae, Informatica* **2(2),** (2010) 184–193. ⇒ 2

[62] Ø J. Rødseth, J. A. Sellers, H. Tverberg, Enumeration of the degree sequences of non-separable graphs and connected graphs. *European J. Comb.* **30,** (5), 1309–1319. ⇒ 2, 14, 22

[63] F. Ruskey, R. Cohen, P. Eades, A. Scott, Alley CAT's in search of good homes. *Congressus Numerantium,* **102** (1994) 97–110. ⇒ 1, 14, 22

[64] G. Sierksma, H. Hoogeveen, Seven criteria for integer sequences being graphic. *J. Graph Theory* **15,** (2) (1991) 223–231. ⇒ 4

[65] B. Siklósi, *Soros és párhuzamos algoritmusok összehasonlítása sportversenyekkel kapcsolatos problémákban.* Programtervező matematikus diplomamunka. ELTE TTK, Budapest, 2001. ⇒ 22

[66] R. Simion, Convex polytopes and enumeration, *Advances in Applied Math.* **18(2)** (1996) 149–180. ⇒14, 22

[67] N. J. A. Sloane (Ed.), *Encyclopedia of Integer Sequences.* 2011. ⇒3, 10

[68] N. J. A. Sloane, *The number of ways to put* $n+1$ *indistinguishable balls into* $n+1$ *distinguishable boxes.* In: The On-line Encyclopedia of the Integer Sequences. 2011. ⇒14, 22

[69] N. J. A. Sloane, *The number of degree-vectors for simple graphs.* In: The On-line Encyclopedia of the Integer Sequences. 2011. ⇒1, 14, 22

[70] N. J. A. Sloane, *The number Number of bracelets with* n *red, 1 pink and* $n-1$ *blue beads.* In: The On-line Encyclopedia of the Integer Sequences. 2011. ⇒14, 22

[71] N. J. A. Sloane, S. Plouffe S. *The Encyclopedia of Integer Sequences.* Academic Press, 1995. ⇒14, 15

[72] D. Soroker, *Optimal parallel construction of prescribed tournaments. Discrete Appl. Math.* **29(1),** (1990) 113–125. ⇒22

[73] R. Stanley, A zonotope associated with graphical degree sequence. In: *Applied geometry and discrete mathematics, Festschr. 65th Birthday Victor Klee.* DIMACS Series in Discrete Mathematics and Theoretical Computer Science. **4** (1991) 555-570. ⇒14, 22

[74] A. Tripathi, H. Tyagy, A simple criterion on degree sequences of graphs. *Discrete Appl. Math.* **156,** (18) (2008) 3513–3517. ⇒

[75] A Tripathi, S. Vijay, A note on a theorem of Erdős & Gallai. *Discrete Math.* **265,** (1–3) (2003) 417–420. ⇒6, 7

[76] A. Tripathi, S. Venugopalanb, D. B. West, A short constructive proof of the Erdős-Gallai characterization of graphic lists. *Discrete Math.* **310,** (4) (2010) 833–834. ⇒1, 2, 4

[77] E. W. Weisstein, *Degree Sequence.* From MathWorld—Wolfram Web Resource, 2011. ⇒2

[78] E. W. Weisstein, *Graphic Sequence.* From MathWorld—Wolfram Web Resource, 2011. ⇒2

[79] K. J. Winston, D. J. Kleitman, On the asymptotic number of tournament score sequences. *J. Combin. Theory Ser. A*. **35,** (1983) 208–230. ⇒ 14, 22