LINEÁRIS ERDŐS-GALLAI TESZT

IVÁNYI ANTAL, LUCZ LORÁND, SÓTÉR PÉTER

Havel 1955-ben [22], Erdős és Gallai 1960-ban [15], Hakimi 1962-ben [20], Ruskey, Cohen, Eades és Scott 1994-ben [61], Barnes and Savage 1997-ben [4], Tripathi, Venugopalan és West 2010-ben [75] javasoltak módszert annak eldöntésére, hogy nemnegatív egészek sorozata lehet-e egy egyszerű gráf foksorozata. Ezeknek az algoritmusoknak a futási ideje legrosszabb esetben a csúcsok számával legalább négyzetesen nő. Cikkünkben bemutatjuk az Erdős-Gallai-Lineárisan algoritmust, amelynek legrosszabb futási ideje lineáris. Mivel például jó sorozat esetén a sorozat minden elemét tesztelni kell, RAM számítási modell esetén EGL aszimptotikusan optimális. Az új algoritmus segítségével 24, 25, 26, 27 és 28 csúcs esetén meghatároztuk egyszerű gráfok különböző foksorozatainak a számát [67].

1. Bevezetés

A gyakorlatban különböző területeken szükség van objektumok rangsorolására. Ennek egyik elterjedt módszere, hogy az objektumokat páronként összehasonlítjuk, és az összehasonlítás eredményeképpen pontokat adunk az objektumoknak, végül pedig az objektumokat a kapott pontszámok alapján rangsoroljuk. Például Landau biológiai [44], Hakimi kémiai [20], Kim, Toroczkai, Miklós, Erdős és Székely [37] és Newman és Barabási [52] hálózati, Bozóki, Fülöp, Poesz és Rónyai gazdasági [7, 8] Liljeros et al. emberi kapcsolatokra vonatkozó [45], Iványi et al. pedig sportbeli [25, 26, 29, 32, 35, 57, 59] alkalmazásokra hivatkoztak.

A számos lehetséges terminológia közül Erdős Pál és Gallai Tibor [15] cikkének szóhasználatát és jelöléseit követjük. 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.

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, 9, 14, 23, 35, 47, 60, 73, 75, 76, 77], illetve [5, 8, 11, 16, 18, 25, 26, 28, 32, 33, 37, 40, 43, 49, 54, 57, 58]).

Legyenek l, m és u egész számok, továbbá $1 \leq m$ és $0 \leq l \leq u$. Egész számok $b = (b_1, \ldots, b_m)$ sorozatát (l, u, m)-*korlátosnak* (röviden: korlátosnak) nevezzük, ha $l \leq b_i \leq u$ minden $1 \leq i \leq m$ indexre. A $b = (b_1, \ldots, b_m)$ (l, u, m)-korlátos sorozatot (l, u, m)-*szabályosnak* (röviden: szabályosnak) mondjuk, ha $u \geq b_1 \geq \cdots \geq b_m \geq l$. Egy (l, u, m)-szabályos sorozatot *megvalósíthatónak* vagy *grafikusnak* (röviden: jónak) nevezünk, ha létezik olyan G egyszerű gráf, hogy G foksorozata $(b_1, b_2, \ldots, b_m)$. Vizsgálatainkban kitüntetett szerepet játszanak a $0, m-1, m$-szabályos sorozatok, ahol m egy egyszerű gráf csúcsainak száma, 0 és $m-1$ pedig a foksorozat alsó, illetve felső korlátja.

Jelentős számú cikk (például [10, 17, 41, 48]) foglalkozik páros számok *grafikus felbontásaival:* előállítják a $2k$ páros szám pozitív egész öszeadandókra való monoton csökkenő felbontásait, és az így kapott $q = (q_1, \ldots, q_m)$ sorozatok közül – amelyekre $q_1 + \cdots + q_m = 2k$ és $q_m \geq q_{m-1} \geq \cdots \geq q_1$ – szűrik ki a grafikus sorozatokat, vagy pedig rekurzióval eleve csak a grafikus sorozatokat állítják elő.

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 és fontos, hiszen a gráfelmélet több mint ötven éve intenzíven foglalkozik vele. 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 [18, 24, 27, 31, 33, 42, ?]. A labdarúgás is olyan sport, amelyben a döntetlen megengedett eredmény. Vizsgálatainkban mind a tesztelés, mind pedig a megvalósítás során kulcsszerepet játszik a döntetlenek párosítása – ez pedig pontosan az egyszerű gráfok fokainak párosításával ekvivalens feladat.

Az egyszerű irányítatlan gráfok foksorozataival kapcsolatos eredmények és módszerek hasznosak az irányított gráfokra, multigráfokra [21, 25, 26, 36, 43], többrészes gráfokra [55, ?] és a foksorozatokhoz hasonló sorozatokra [55, 58, 59] vonatkozó hasonló feladatok megoldásánál.

Érdemes megemlíteni, hogy a fokszámsorozaok számának meghatározásával kapcsolatos nehézségek miatt annak is van irodalma (lásd például [14, 35, 49]), hogy véletlen mintavétellel becsüljük ezeket a számokat.

Melléktermékként javítottuk és bővítettük az EIS [66] adatbázist: egyrészt a benne lévő ismert sorozatok új elemeivel, másrészt a cikkben definiált új sorozatok elemeinek megadásával. Módszerünk az összes jó sorozat gazdaságos előállítására is alkalmas (vesd össze Ruskey [61] 1994-es, valamint Barnes and Savage [4] 1997-es cikkével).

A cikk felépítése a következő. A bevezető első rész után a témakör klasszikus algoritmusait foglaljuk össze. A harmadik részben új pontos algoritmusokat, a negyedik részben leszámlálási eredményeket, az ötödik részben pedig új tesztelő algoritmusokat ismertetünk. A hatodik részben a közelítő algoritmusok hibáját, míg a hetedikben a pontos algoritmusok futási idejét elemezzük. Végül a nyolcadik részben a grafikus sorozatok számának nagyságrendjét jellemezzük.

2. Klasszikus pontos algoritmusok

Ebben a részben a témakör két klasszikus algoritmusát ismertetjük.

2.1. Havel-Hakimi algoritmus

A feladat megoldására az első módszert Vaclav Havel cseh matematikus javasolta 1955-ben [22, 46].

1. TÉTEL. (Havel [22]) *Ha* $n \geq 1$, *a* $(b_1, \ldots, b_n)$ *n-szabályos sorozat akkor és csak akkor n-jó, ha a* $(b_2-1, b_3-1, \ldots, b_{b_1}-1, b_{b_1+1}-1, b_{b_1+2}, \ldots, b_n)$ *sorozat* $(n-1, n-1)$*-jó.*

Bizonyítás. Lásd [22]. □

Az eredeti tétel $n \geq 3$ korlátozó feltételt tartalmaz, de könnyű ezt az $n \geq 1$ feltételre módosítani.

Ha ezen tétel alapján írunk egy rekurzív algoritmust, akkor annak futási ideje a RAM számítási modell [12] szerint legjobb esetben – például az $(n-1, 0, \ldots, 0) = (n-1, 0^{n-1})$ bemenetre – $\Theta(1)$, legrosszabb esetben pedig – például a *homogén* $(n-1)^n$ vagy a *tranzitív* $(n-1, n-2, \ldots, 1, 0)$ bemenetre – $\Theta(n^2)$. 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 a teljes gráfnak $\binom{n}{2} = \Theta(n^2)$ éle van, bármely – a megvalósítást is biztosító – RAM algoritmus legrosszabb futási ideje nem lehet kisebb, mint négyzetes.

1962-ben Louis Hakimi [20] Haveltől függetlenül publikálta ugyanezt az eredményt, ezért ma a tételt rendszerint *Havel-Hakimi tételnek*, a módszert pedig *Havel-Hakimi algoritmusnak* nevezik.

Az algoritmust később irányított gráfokra [16, 25, 26, 38] is kiterjesztették.

2.2. Erdős-Gallai algoritmus

Időrendben a következő eredmény Erdős Pál és Gallai Tibor alábbi szükséges és elégséges feltétele [15] volt.

Nemnegatív egészek adott $b = (b_1, \ldots, b_n)$ sorozata esetén a sorozat első i elemét a sorozat i elemhez tartozó *fejének*, míg a többi elemét a i elemhez tartozó *farkának* nevezzük. A fejelemek összegét H_i, míg a farokelemek összegét T_i jelöli ($i = 1, \ldots, n$). A $\sum_{k=i+1}^{n} \min(i, b_k)$ összeget pedig C_i-vel jelöljük és a farok *kapacitásának* nevezzük. Ha egy b sorozatra H_n páros, akkor a sorozatot *párosnak,* egyébként *páratlannak* nevezzük.

2. Tétel. (Erdős, Gallai, [15]) *Ha $n \geq 1$, az n-szabályos $(b_1 \ldots, b_n)$ sorozat akkor és csak akkor jó, ha*

$$H_n \quad \textit{páros} \tag{1}$$

és

$$H_i - i(i-1) \leq C_i \quad (i = 1, \ldots, n-1). \tag{2}$$

Bizonyítás. Lásd [13, 15, 62, 75]. □

Bár ez a tétel csak ellenőriz, futási ideje a legjobb $\Theta(n)$ és a legrosszabb $\Theta(n^2)$ között változik. A közelmúltban Tripathi et al. [75] publikáltak a tételre konstruktív bizonyítást, amely jó bemenet esetén $O(n^3)$ idő alatt egy megoldást is előállít.

3. Új pontos algoritmusok

Ebben a részben a klasszikus algoritmusok néhány gyorsított változatát mutatjuk be.

3.1. Nullamentes algoritmusok

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.

3. Következmény. *Ha $n \geq 1$, a $(b_1, \ldots, b_n)$ (n, n)-szabályos sorozat akkor és csak akkor n-jó, ha $b_n = 0$ vagy a $(b_1, \ldots, b_p)$ sorozat (p, p)-jó.*

Bizonyítás. 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. Eltoló Havel-Hakimi algoritmus

Havel és Hakimi eredeti tételének természetes algoritmikus megfelelőjét HHR-nek (Rendező Havel-Hakimi) 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 az Eltoló Havel-Hakimi algoritmust (HHE) kapjuk.

3.3. Paritásos Havel-Hakimi algoritmus

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 nullamentes 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.

3.4. Rövidített Erdős-Gallai algoritmus

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

Ennél is hasznosabb észrevételt tartalmaz a következő lemma.

Tripathi és Vijay 2003-as cikkében [74] szerepel az az észrevétel, hogy az Erdős-Gallai tételben a (2) egyenlőtlenséget elég csak addig ellenőrizni, amíg $H_i > i(i-1)$ teljesül.

4. Lemma. (Tripathi és Vijay [74]) *Ha $n \geq 1$, az n-szabályos $b = (b_1 \ldots, b_n)$ sorozat akkor és csak akkor n-jó, ha*

$$H_n \quad \textit{páros} \tag{3}$$

és

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

ahol

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

Bizonyítás. 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 7. részben vizsgáljuk.

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

3.5. Ugró Erdős-Gallai algoritmus

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

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 $b_{\sigma_1}, b_{\sigma_2}, \ldots, b_{\sigma_q}$ elemek.

5. Tétel. (Tripathi, Vijay [74]) *Az n-szabályos $b = (b_1 \ldots, b_n)$ sorozat akkor és csak akkor n-jó, ha*

$$H_n \quad \textit{páros} \tag{6}$$

és

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

Bizonyítás. Lásd [74]. □

Később majd az Erdős-Gallai-Lineárisan algoritmusban kihasználjuk, hogy a (7) egyenlőtlenségben σ_q mindig n, és ezért $H_n - n(n-1)$ mindig teljesül, így elég az egyenlőtlenséget $(i = q - 1)$-ig ellenőrizni.

A következő program az Erdős-Gallai algoritmusnak az 3. és 4. lemma, valamint a 5. tétel alapján gyorsított változatát mutatja be.

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

Kimenet. 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;
$H = (H_1, \ldots, H_n)$: 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-Ugró(n, b, L, γ)

```
01 p = n                                   ▷ 01–03. sor: nullamentesítés
02 while b_p = 0
03         p = p − 1
04 H_1 = b_1                               ▷ 04–09. sor: paritás ellenőrzése
05 for i = 2 to p
06       H_i = H_{i−1} + b_i
07 if H_p páratlan
08    L = False
09    return L
10 b_{p+1} = 0                             ▷ 11–23. sor: fej igényének ellenőrzése
11 i = 1
```

```
while i ≤ p és i(i − 1) < H_i
        if b_i = b_{i+1}
            i = i + 1
            E = 0
            for j = i + 1 to p
            E = E + min(j, b_j)
        if H_i > i(i − 1) + E
            L = FALSE
            return L
L = TRUE
return L
```

Ennek az algoritmusnak a futási ideje a legjobb $\Theta(1)$ és a legrosszabb $\Theta(n^2)$ között változik.

3.6. Lineáris Erdős-Gallai algoritmus

A következő Erdős-Gallai-Lineárisan algoritmus kihasználja, hogy a b bemeneti sorozat monoton. Ennek köszönhetően a C_i kapacitásokat minden i-re konstans időben meg tudja határozni, azaz nincs szüksége arra, hogy a megfelelő farok elemeit egyenként megvizsgálja. A gyors számolás kulcsa a mutatókat tartalmazó $m(b)$ sorozat.

Adott b sorozat esetén legyen $m(b) = (m_1, \ldots, m_{n-1})$, ahol m_i a b sorozat legnagyobb indexű olyan elemére mutat, ameyik legalább akkora, mint i.

A b sorozat b_i elemének ellenőrzésekor két eset van: ha $i > m_i$, akkor a C_i kapacitás egyszerűen számítható: $H_n - H_i$, mivel a farok minden b_j elemének hozzájárulása csak b_j.

Ha viszont $i \le m_i$, akkor a C_i-t definiáló szummát két részre bontjuk: az első részhez a farok azon b_j kezdő elemeinek hozzájárulása tartozik, amelyekre teljesül $b_j \ge i$, a második részhez pedig a többi elem. Legyen $q(b) = q = \max_{1 \le i \le n}\{i \mid i(i-1) \le H_i\}$.

6. Tétel. *Ha $n \ge 1$, a $b = (b_1 \ldots, b_n)$ n-szabályos sorozat akkor és csak akkor n-jó, ha*

$$H_n \quad \textit{páros}, \tag{8}$$

továbbá

$$H_i \le i(k-1) + H_n - H_k \quad (i = 1, \ldots, q), \tag{9}$$

ahol

$$k(b) = k = \begin{cases} m_i, & \textit{if } i \le m_i, \\ i, & \textit{if } i > m_i. \end{cases} \tag{10}$$

Bizonyítás. Megmutatjuk, hogy a tételben szereplő feltétel ekvivalens a 2. tétel feltételeivel.

A (8) feltétel pontosan megegyezik az (1) feltétellel.

Ha $i \le m_i$, akkor

$$H_i \le i(i-1) + (m_i - i + 1)i + H_n - H_{m_i} \tag{11}$$

és ha $i > m_i$, akkor

$$H_i \le i(i-1) + H_n - H_i. \tag{12}$$

Ha (**??**) jobb oldalán kiemeljük i-t, akkor a

$$H_i \le im_i + H_n - H_{m_i} \tag{13}$$

egyenlőtlenséget kapjuk. Ha (**??**)-ba (10) alapján behelyettesítjük k-t, akkor az $i \le m_i$ esetben (**??**)-t, az $i > m_i$ esetben pedig (**??**)-t kapjuk. □

A következő program a 6. tétel alapján adott n-re tetszőleges n-szabályos sorozatról eldönti, hogy jó-e. A program futási ideje minden sorozatra $O(n)$. Érdemes megjegyezni, hogy akár a bemenő sorozat rendezettségétől is eltekinthetünk, mivel a sorozat elemei egész számok és mindegyik a $[0, n-1]$ intervallumba esik, így szükség esetén $O(n)$ idő alatt rendezni tudjuk a sorozatot.

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

Kimenet. L: logikai változó, melynek értéke True, ha a bemenet jó, és False egyébként.

Munkaváltozók. i és j: ciklusváltozók;
$H = (H_1, \ldots, H_n)$: H_i az éppen tesztelt b első i elemének az összege;
$m = (m_1, \ldots, m_{n-1})$: m_i a legnagyobb indexű olyan b_j indexe, amely legalább i;
$H_0 = 0$: segédváltozó a H sorozat elemeinek kiszámításához;
$b_0 = n - 1$: segédváltozó az m sorozat elemeinek kiszámításához;
q: ellenőrzésre eddig az indexig van szükség;
k: ellenőrzés egyszerűsítéséhez használt változó.

Erdős-Gallai-Lineárisan(n, b, L)

```
01 H_0 = q = 0                         ▷ 01. sor: kezdeti értékek beállítása
02 for i = 1 to n                      ▷ 02–03. sorok: H elemeinek kiszámítása
03     H_i = H_{i-1} + b_i
04     if i(i-1) < H_i                 ▷ 04–05. sorok: q kiszámítása
05         q = i
06 if H_n  páratlan                    ▷ 06–08. sor: paritás ellenőrzése
07     L = False
08     return
09 b_0 = n - 1                         ▷ 09. sor: segédváltozó beállítása
10 for j = n downto b_1 + 1            ▷ 10–11. sorok: nagy indexű mutatók beállítása
11     m_j = 0
12 for i = 1 to n                      ▷ 12–16. sorok: mutatók beállítása
```

```
13     if b_i < b_{i-1}
14        for j = b_{i-1} downto b_i + 1
15            m_j = i - 1
16     m_{b_i} = i
17 for j = b_n - 1 downto 1          ▷ 17–18. sorok: nagy indexű mutatók beállítása
18     m_j = n - 1
19 if i ≤ m_i                        ▷ 19–21. sorok: k beállítása
20     k = m_i
21 else k = i
22 for i = 1 to q                    ▷ 22–25. sorok: sorozat ellenőrzése
23     if H_i > i(k - 1) + H_n - H_k
24        L = FALSE
25        return L
26 L = TRUE                          ▷ 26–27. sor: a program TRUE értékkel befejeződik
27     return L
```

7. KÖVETKEZMÉNY. *Nemnegatív egészek n-szabályos $b = (b_1, \ldots, , b_n)$ sorozatáról $O(n)$ idő alatt eldönthető, hogy létezik-e olyan G egyszerű gráf, amelynek b a foksorozata.*

Bizonyítás. ????? □

3.7. Gyors Erdős-Gallai algoritmus

A ERDŐS-GALLAI-LINEÁRISAN algoritmus egyik lehetséges alkalmazása, hogy meghatározzuk a jó sorozatok számát olyan n értékekre, amelyekre eddig a nagy számolásigény miatt nem volt ismert: Sloane *On-line Encyclopedia of Integer Sequences* című honlapja [66] az $n = 23$ értékig tartalmazza a jó sorozatok számát.

Az alábbi ERDŐS-GALLAI-GYORSAN algoritmus a lineáris legrosszabb eset mellett azt is igyekszik kihasználni, hogy ha lexikografikus sorrendben ellenőrizzük a szóbajövő sorozatokat, akkor a szomszédos sorozatok bizonyos tulajdonságai nagyon hasonlóak, ezért adott sorozat jellemzői az őt megelőző sorozat jellemző adataiból konstans idő alatt meghatározhatóak.

Igyekszünk az ellenőrizendő sorozatok számát is csökkenteni.

Ennek egy egyszerű megoldása, hogy eleve csak a páros sorozatokat állítjuk elő. További ötlet, hogy csak a nullamentes sorozatokat vizsgáljuk. A nullát tartalmazó n-jó sorozatok között ugyanis a 16. lemma szerint pontosan $\gamma(n-1)$ jó sorozat van. Igaz, hogy aszimptotikusan a nullát tartalmazó sorozatok a páros sorozatok elhanyagolható részét adják, de az általunk most gyakorlatilag vizsgált $n \in [4, 30]$ tartományban még jelentős a részarányuk.

Lényeges gyorsítást jelent az is, hogy a sorozatokat csak az ellenőrző pontokban vizsgáljuk.

Az Erdős-Gallai-Gyorsan program azt is kihasználja, hogy a szomszédos sorozatok ellenrző pontjainak a listája átlagosan konstans idő alatt származtatható a megelőző sorozat adataiból. A kiindulási értékek szintén könnyen számíthatók: az első – $\mathbf{q} = (n-1)^n$ – sorozatra a C lista üres (azaz egyáltalán nem kell elllenőrzést végeznünk), a mutatók listája pedig kezdetben $m = \big((n-1)^{n-1}$

Az Erdős-Gallai-Gyorsan algoritmus előállítja és megvizsgálja az n-páros, nullamentes sorozatokat, és kimenetként megadja a $\gamma(n)$ értéket. Az algoritmus kihasználja, hogy a páros sorozatok lexikografikusan csökkenő sorozatában szomszédos sorozatok több lényeges paramétere hasonló, ezért ezek a paraméterek a vizsgált b' sorozatot megelőző b sorozat adott paraméteréből gyorsan meghatározhatóak.

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.

Bemenet. n: a csúcsok száma ($n \geq 4$); (azért korlátozzuk n-et alulról, hogy a programot mentesítsük a rövid sorozatok speciális tulajdonságainak figyelembe vételétől).

Kimenet. γ: az n-jó sorozatok száma.

Munkaváltozók. i és j: ciklusváltozók;
$b = (b_1, \ldots, b_n)$: b_i az éppen tesztelt páros, nullamentes sorozat i-edik eleme;
$b_0 = n-1$: segédváltozó a H sorozat elemeinek kiszámításához;
$H = (H_1, \ldots, H_n)$: H_i az éppen tesztelt b első i elemének az összege;
H_0: segédváltozó H elemeinek számolásához;
$C(b) = C = (c_1, c_2, \ldots, c_{q-1})$: ellenőrző pontok maximális hosszúságú listája;
c: ellenőrző pontok száma az aktuális C listában;
$m = m_1, \ldots, m_{n-1}$: m_i a legnagyobb indexű olyan b_j indexe, amely legalább i; q: ellenőrzésre eddig az indexig van szükség; k: ellenőrzés egyszerűsítéséhez használt változó.

Erdős-Gallai-Gyorsan(n, γ)
01 $H_0 = c = 0$ ▷ 01–03. sor: kezdeti értékek beállítása
02 $\gamma = 1$
03 ???
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** $b_2 \geq 2$ ▷ 07. sor: van-e még vizsgálandó sorozat?
08 $\qquad$???
09 $\qquad$???
10 $\qquad$ **for** $i = 1$ **to** c ▷ 10–15. sor: aktuális sorozat tesztelése
11 $\qquad\quad$ **if** ???
12 $\qquad\qquad$ **if** ???
13 $\qquad\qquad\quad$???
14 $\qquad\quad$ **else if** ???

15 **go to** 19 ▷ áttérés új sorozatra
16 $\gamma = \gamma + 1$ ▷ 16. sor: jó sorozat esetén γ növelése
17 **go to** 19 ▷ áttérés új sorozatra
18 **return** γ ▷ 18. sor: program befejezése
19 **if** $b_n \geq 3$ ▷ 19–??. sor: új sorozat előállítása
20 $b_n = b_n - 2$ ▷ 19–25. sor: ha $b_n = 3$
21 $H_n = H_n - 2$
22 **if** $b_n == b_{n-1} - 2$
23 $c = c + 1$
24 $C_c = n - 1$
25 **go to** 07
26 **else if** $b_n == 2$ ▷ 26–46. sor: ha $b_n = 2$
27 **if** $b_{n-1} > 2$
28 $b_{n-1} = b_{n-1} - 1$
29 $H_{n-1} = H_{n-1} - 1$
30 $b_n = b_{n-1}$
31 $H_n = H_n + b_n - 3$
32 **if** b_{n-1} páros
33 $b_n = b_n - 1$
34 $C_c = n - 2$
35 $c = c + 1$
36 $C_c = n - 1$
37 **if** $b_{n-2} > b_{n-1} + 1$
38 $c = c + 1$
39 $C_c = n - 1$
40 **else** $C_c = n - 2$
41 **else if** $b_{n-1} == b_n$
42 $C_c = n - 2$
43 **else** $c = c + 1$
44 $C_c = n - 1$
45 **go to** 07
46 **else** $j = n - 1$ ▷ 46–??. sor: ha $b_n = 1$
47 **while** $b_j == 1$
48 $j = j - 1$
49 **if** $b_j > 2$
50 **for** $k = j$ **to** n
51 $b_k = b_j - 1$
52 $H_k = H_k - 1 + (k - j)(b_j - 1)$
53 **if** $(n - j)(b_j + 1)$ páros
54 $b_n = b_n - 1$
55 $H_n = H_n - 1$
56 **if** $j > 1$

if $b_j < b_{j-1} - 1$
  $c = c - 1$
  if $b_n < b_{n-1}$
    $c = c + 1$
    $C_c = n - 1$
  else $C_c = j - 1$
    if $b_n < b_{n-1}$
      $c = c + 1$
      $C_c = n - 1$
go to 07
else **for** $k = j - 1$ **to** n
  $b_k = b_{j-1} - 1$
  $H_k = H_k - 1 + (k - j + 1)(b_{j-1} - 1)$

???
???
???
???
???
???
go to 07

4. Leszámlálási eredmények

Eddig például Avis és Fukuda [2], Barnes és Savage [3, 4], Burns [10], Erdős és Moser [50], Frank, Savage and Sellers [19], Kleitman és Winston [39], Rødseth, Sellers, Tverberg [60], Ruskey et al. [61], Simion [64], Stanley [72], Winston és Kleitman [78] publikáltak foksorozatok leszámlálására vonatkozó eredményeket. Az általunk vizsgált sorozatok számával kapcsolatos eredmények találhatók Sloane és Ploffe [65], valamint Stanley [71] könyvében és az *On-line Encyclopedia of Integer Sequences* című honlapon [67, 68, 69] is.

8. Lemma. *Ha l, m és u egész számok, továbbá $l \geq u$ és $1 \geq m$, akkor az (l, u, m)-korlátos sorozatok $\kappa(l, u, m)$ száma*

$$\kappa(l, u, m) = (u - l + 1)^m. \tag{14}$$

Bizonyítás. A korlátos sorozatok számára vonatkozó képlet közvetlen adódik abból, hogy a sorozat mind az m eleme $u - l + 1$ lehetséges értéket vehet fel. □

9. Lemma. *Ha* 1, m *és* u *egész számok, továbbá* $l \geq u$ *és* $1 \geq m$, *akkor az* (l,u,m)-*szabályos sorozatok* $\rho(l,u,m)$ *száma*

$$\rho(l,u,m) = \binom{m+u-l}{m}. \tag{15}$$

Bizonyítás. Egy $b = (b_1,\ldots,b_m)$ (l,u,m)-szabályos sorozat esetén legyen $b' = (b'_1,\ldots,b'_m)$, ahol $b'_i = b_i + m - i$. A lehetséges b és b' sorozatok halmazai között kölcsönösen egyértelmű kapcsolat áll fenn. A különböző b' sorozatok száma pedig annyi, ahány féleképpen a különböző l, $l+1$, …, $u+m-1$ számok – azaz $u+m-l$ szám – közül m számot ki tudunk választani. □

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

10. Lemma. *Ha* $n \geq 1$, *akkor*

$$\frac{\rho(n+2)}{\rho(n+1)} > \frac{\rho(n+1)}{\rho(n)}, \tag{16}$$

$$\lim \rho_{n\to\infty} = 4, \tag{17}$$

továbbá

$$\frac{4^n}{\sqrt{4\pi n}}(1-\frac{1}{2n}) < \rho(n) < \frac{4^n}{\sqrt{4\pi n}}(1-\frac{1}{8n+8}). \tag{18}$$

Bizonyítás. A (45) egyenlőség alapján

$$\frac{\rho(n+2)}{\rho(n+1)} = \frac{(2n+3)!(n+1)n!}{(n+2)!(n+1)!(2n+1)!} = \frac{4n+6}{n+2} = 4 - \frac{2}{n+2}, \tag{19}$$

ahonnan (16) és (17) is közvetlenül adódik.

(18) belátásához felhasználjuk a Stirling-formula következő alakját [12]: ha $n \geq 1$, akkor

$$n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n}e^{\tau_n} \tag{20}$$

ahol

$$\frac{1}{12n+1} < \tau_n < \frac{1}{12n}. \tag{21}$$

□

11. Lemma. (Ascher [1], Sloane and Pfoffe [65]) *Ha* $n \geq 1$, *akkor az* n-*páros sorozatok* $\epsilon(n)$ *száma*

$$\epsilon(n) = \frac{1}{2}\left(\binom{2n-1}{n} + \binom{n-1}{\lfloor n \rfloor}\right). \tag{22}$$

Bizonyítás. Lásd [65]. □

A 9. 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.

12. Lemma. *Ha* $n \geq 1$, *akkor*

$$\frac{\epsilon(n+2)}{\epsilon(n+1)} > \frac{\epsilon(n+1)}{\epsilon(n)}, \tag{23}$$

$$\lim \epsilon_{n \to \infty} = 4, \tag{24}$$

továbbá

$$\frac{4^n}{\sqrt{\pi n}}(1 - \delta_3(n)) < \epsilon(n) < \frac{4^n}{\sqrt{\pi n}}(1 - \delta_4(n)), \tag{25}$$

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

Bizonyítás. A bizonyítás hasonló a 10. lemma bizonyításához. □

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

13. Következmény. *Ha* $n \geq 1$, *akkor*

$$\frac{\epsilon(n+1)}{\rho(n+1)} < \frac{\epsilon(n)}{\rho(n)} \tag{26}$$

és

$$\lim_{n \leftarrow \infty} \frac{\epsilon(n)}{\rho(n)} = \frac{1}{2}. \tag{27}$$

Bizonyítás. ???? □

Bár az alapfeladatban nemnegatív elemekből álló sorozatok szerepelnek, algoritmusaink – a futási idő csökkentése érdekében – csak a sorozatok pozitív kezdőszeletét vizsgálják. Ennek várható hatását jellemzi a következő két állítás, amelyek a nullát tartalmazó sorozatok számát és a sorozatokban lévő nullák átlagos számát adják meg.

14. Lemma. *Ha* $n \geq 1$, *akkor az* n*-szabályos sorozatok közül*

$$\binom{2n-2}{n-1} = \frac{n}{2n-1}\rho(n). \tag{28}$$

tartalmaz legalább egy nullát.

Bizonyítás. A nullát tartalmazó n-szabályos sorozatok halmaza kölcsönösen egyértelműen leképezhető az $(n-1,n)$-szabályos sorozatok halmazára. Az utóbbi halmaz elemszáma pedig a 9. lemma szerint

$$\binom{2n-2}{n-1} = \frac{(2n-2)!n}{n(n-1)!(2n-1)} = \frac{n}{2n-1}\binom{2n-1}{n}. \tag{29}$$

□

15. Lemma. *Ha $n \geq 1$, akkor az n-szabályos sorozatokban lévő nullák átlagos száma*

$$\frac{\sum_{i=0}^{n} i\binom{2n-3}{n-i}}{\binom{2n-1}{n}}. \tag{30}$$

Bizonyítás. Az n-szabályos sorozatokban $0,\ 1,\ldots,n-1$ vagy n nulla van. Az i nullát és $n-i$ pozitív elemet tartalmazó n-szabályos sorozatok száma ugyanannyi, mint ahány féleképpen $2n-3$ különböző elem közül $n-i$ kiválasztható. □

A pontos algoritmusokról szóló 3.1. részben beláttuk, hogy elég az n-páros sorozatok nullamentes prefixét megvizsgálni ahhoz, hogy eldöntsük, jó-e a vizsgált sorozat. Mivel a 14 páros sorozatoknak aszimptotikusan csak nullmértékű hányada tartalmaz nullát (és ez a hányad a gyakorlat számára legérdekesebb n-ekre sem nagy), konkrét sorozatok vizsgálatánál nem jelentős az időmegtakarítás. Amikor viszont az összes n-páros sorozatot elemezzük (az átlagos futási idő vagy $\gamma(n)$ meghatározása érdekében), nagyon hasznos a következő lemma.

Legyen $\zeta(n)$ a nullamentes jó n-páros sorozatok száma.

16. Lemma. *Ha $n \geq 2$, akkor az n-jó sorozatok száma*

$$\gamma(n) = \zeta n + \gamma(n-1). \tag{31}$$

Bizonyítás. Az n-jó sorozatokban vagy $b_n = 0$, vagy $b_n > 0$. Az előbbiekben vagy $b_1 = n-1$, vagy $b_1 < n_1$. Ha $b_1 = n-1$ és $b_n = 0$, akkor a b sorozat biztosan rossz, mert nincs benne elég pozitív elem. A $b_1 < n-1$ és $b_n = 0$ tulajdonságú sorozatok $n-1$ hosszú fejei pontosan az $(n-1)$-jó sorozatok. □

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.

17. Lemma. (Móri [51]). *Ha $n \geq 1$, akkor az n-korlátos sorozatok átlagos szivárvány száma*

$$\frac{n^2}{2n-1}. \tag{32}$$

Bizonyítás. □

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

18. LEMMA. (Móri [51]). *Ha $n \geq 1$, akkor az n-szabályos sorozatok átlagos szivárvány száma*

$$\frac{n}{2} + \frac{1}{4} + \frac{3}{8n-4}. \tag{33}$$

Bizonyítás. 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 9. lemma szerint az n-szabályos sorozatok száma $\rho(n) = \binom{2n-1}{n}$, az n-szabályos sorozatok átlagos szivárványszáma

$$\frac{\sum_{k=1}^{n} k\binom{n}{k}\binom{n-1}{k-1}}{\binom{2n-1}{n}}. \tag{34}$$

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

$$\frac{n^2}{2n-1} = \frac{n}{2} + \frac{1}{4} - \frac{3}{8n-4}. \tag{35}$$

□

A következő állítás a rendező és az eltoló Havel-Hakimi algoritmusok futási idejének összehasonlításánál hasznosak.

19. LEMMA. *Ha $n \geq 1$, akkor egy n-szabályos sorozatot $b = (b_1^{e_1}, \ldots, b_q^{e_q})$ alakban felírva az e_j kitevők („futamhosszak") várható értéke*
konstans ???

Bizonyítás. ???? □

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, 10]. Ezek segítségével sikerült a jó sorozatok számára vonatkozó alábbi aszimptotikus korlátokat bizonyítani.

20. LEMMA. (Burns [10]) *Léteznek olyan pozitív c és C állandók, hogy a jó sorozatok $\gamma(n)$ száma a következő korlátok közé esik:*

$$\frac{4^n}{cn} < \gamma(n) < \frac{4^n}{(\log n)^C \sqrt{n}}\ . \tag{36}$$

5. Tesztelő algoritmusok

Sorozatok megvalósíthatóságának vizsgálata során természetes észrevétel, hogy a b sorozat i-hez tartozó fejének H_i fokszámigényét részben belső (az adott fejen belüli), részben pedig külső (a fejnek megfelelő farokhoz tartozó) fokszámokkal elégítjük ki.

Először egy „paritásos”, majd egy „pozitív”, egy „binomiális”, egy „haszontalan” egy végül egy „fejfelező” tesztelő/szűrő algoritmust mutatunk be.

5.1. Paritás teszt

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.

21. LEMMA. *Ha $n \geq 1$ és b n-jó sorozat, akkor*

$$H_n \quad \textit{páros.} \tag{37}$$

Bizonyítás. Egy egyszerű gráf minden éle kettővel növeli a fokszámok összegét. □

Ezt az állítást a 2. tétel következményeként is megkaphatjuk. A 21. lemmában javasolt tesztet a következő algoritmus végzi.

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

Kimenet. 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);
$H = (H_1, \ldots, H_n)$: H_i a b sorozat első i elemének összege;
ϵ: a vizsgált páros sorozatok száma.

Munkaváltozó. i: ciklusváltozó.

PARITÁS-TESZT(n, b, L)

$H_1 = 0$
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 $\Theta(n)$.

5.2. Pozitív teszt

A farokban lévő nulla elemek nem növelik a farok párosítási lehetőségeit. Ez az észrevétel lehetővé teszi, hogy az i-edik elemhez tartozó farok foklekötési lehetőségeire (potenciáljára) T_i-nél pontosabb becslést adjunk. Ez a teszt a Havel-Hakimi algoritmus első menetének megfelelő ellenőrzést végzi el.

22. Következmény. *Ha b n-jó sorozat, akkor*

$$b_1 \leq p - 1. \tag{38}$$

Bizonyítás. A (38) egyenlőtlenség azt a követelményt fejezi ki, amelyet a Havel-Hakimi algoritmus az első iterációs menetben, illetve az Erdős-Gallai algoritmus a (2) egyenlőtlenség $i = 1$ esetben való ellenőrzésével megvalósít. □

A 22. következményen alapuló tesztet a következő algoritmus végzi.

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

Kimenet. 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).
p: a bemenetben lévő pozitív elemek száma.

Munkaváltozó. Nincs.

```
Pozitív-teszt(n, b, H, T, L, p, π)
p = n
while b_p = 0
        p = p − 1
if H_1 > p − 1
    L = False
    return L
L = True
return L
```

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

5.3. Binomiális teszt

Harmadik tesztünk az Erdős-Gallai tétel másik szükséges feltételének ötletét terjeszti ki. Lényege, hogy a fej igényének a fejen belül ki nem elégíthető részét a faroknak, a farok igényének belül ki nem elégíthető részét a fejnek kell kielégítenie, végül a teljes sorozat igényét a fej és farok együttműködésével, valamint a fej és a farok belső éleivel kell kielégíteni. Az algoritmus nevét arról kapta, hogy a fej és a farok belső éleinek a számát egy-egy binomiális együttható segítségével becsüljük. Legyen p a b sorozat pozitív elemeinek a száma.

23. LEMMA. *Ha $n \geq 1$ és b n-jó sorozat, akkor*

$$2H_i \leq i(i-1) + T_i \quad (i = 1, \ldots, p). \tag{39}$$

Bizonyítás. A (39) egyenlőtlenség azt fejezi ki, hogy a fej H_i igényét a legfeljebb $i(i-1)$ belső lehetőség és a farok legfeljebb T_i kapacitása segítségével kell kielégíteni, ahol $T_I = H_n - H_i$. □

A 23. lemmában javasolt tesztet végzi el a következő program.

Bemenet. n: a csúcsok száma ($n \geq 1$);
$b = (b_1, b_2, \ldots, b_n)$: n-páros sorozat;
p: a b sorozat pozitív elemeinek a száma;
$H = (H_1, \ldots, H_n)$: H_i a b sorozat első i elemének összege.

Kimenet. 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).

Munkaváltozók. i: ciklusváltozó.

```
BINOMIÁLIS-TESZT(n, b, H, L)
p = n
while b_p == 0
        p = p − 1
if p == 1
   L = FALSE
   return L
H_1 = 0
for i = 2 to p
          H_i = H_{i−1} + b_i
for i = 1 to p
      if   2H_i > i(i − 1) + H_p
           L = FALSE
           return H, T, L
L = TRUE
return H, L
```

Az algoritmus azért kezdi b végénél p meghatározását, mert a 16. lemma szerint kevés nulla várható a sorozatokban.

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

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

Megjegyezzük, hogy BINOMIÁLIS-TESZT $i = 1$ esetén elvégzi POZITÍV-TESZT munkáját, ezért a POZITÍV-TESZT algoritmusra nincs szükségünk.

Felmerült, hogy a BINOMIÁLIS-TESZT algoritmust is csak az ellenőrző pontokon alkalmazzuk, a szimulációs kísérletek azonban azt mutatták, hogy ezzel csökkenne az algoritmus hatékonysága. A rövidített változat (n helyett csak g-ig végezzük az ellenőrzést) azonban nyilván igaz.

n helyett p viszont gyengítené az algoritmust, mert például a rossz $(2, 2, 0)$ sorozatot *nem* szűrné ki. Mivel azonban mi csak a páros nullamentes algoritmusokat vizsgáljuk, a $2, 2, 0$ és hasonló sorozatokat egyetlen algoritmusunk sem teszteli (mert ezeket már a sorozatok előállítása során kiszűrjük).

5.4. Haszontalan teszt

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.

24. Lemma. *Ha b n-jó sorozat, akkor*

$$H_i \leq \min\left(i(i-1), H_i\right) + T_i \quad (i = 1, \ldots, n-1). \tag{40}$$

Bizonyítás. A (40) egyenlőtlenség bal oldalán szereplő H_i a b sorozat fejének fokszámigénye. Az egyenlőtlenség jobb oldalán $i(i-1)$ a fejen belüli csúcspárok számából adódó korlát, míg H_i abból adódik, hogy a fej belső éleihez legfeljebb annyi szabad fokot tudunk felhasználni, mint a fej elemeinek összege. □

Az új követelmény csak látszólag javítja a tesztelés hatékonyságát, mert ha a fejelemek összege kisebb vagy egyenlő a mellette lévő binomiális együtthatónál, akkor a minimum már T_i nélkül is biztosítja az egyenlőtlenség teljesülését. 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 az 3. részben a 4. lemmát), másrészt további, *hasznos* közelítő algoritmusoknak lesz a kiindulási pontja.

5.5. Rövidített és ugró binomiális teszt

5.6. Fej felezése

A b sorozat fokpárosító lehetőségeinek az eddigieknél pontosabb becslését kaphatjuk, ha a fejet két részre osztjuk. Legyen $\lfloor i/2 \rfloor = h_i$. Ekkor a $(b_1, \ldots, b_{h_i})$ sorozatot az i indexhez tartozó fej *elejének*, a $(b_{h_i+1}, \ldots, b_i)$ sorozatot pedig az i indexhez tartozó fej *végének* nevezzük.

25. Lemma. *Ha $n \geq 1$ és b n-jó sorozat, akkor*

$$H_i \leq 2\min(X_{i,1} + X_{i,2}, +X_{i,3} + \min(T_i, i(p-i)) \quad (i = 1, \ldots, p), \tag{41}$$

ahol

$$X_{i,1} \leq \min(H_{h_i}, T_n - T_{h_i}, h_i(p-i)), \tag{42}$$

$$X_{i,2} \leq \min(H_i - H_{h_i}, T_n - T_{h_i}, (i - h_i)(p - i)) \tag{43}$$

és

$$X_{i,3} = \min(h_i(h_i - 1), H_i - H_{h_i} - X_{i,1}/2). \tag{44}$$

Bizonyítás. Legyen G a b sorozatot megvalósító G gráf. Ekkor az i indexhez tartozó fej H_i fokszámösszegét lekötő éleinek halmazát öt részhalmazra osztjuk: a fej eleje és a farok, a fej vége és a farok közötti, a fej két része közötti, valamint a fej részein belüli élekre. Az egyes részhalmazokba tartozó élek száma legyen rendre $X_{i,1}, \ldots, X_{i,5}$.

$X_{i,1}$ legfeljebb a fej elemeinek H_{h_i} összege, legfeljebb a farok elemeinek $T_n - T_{h_i}$ összege, és legfeljebb a fej elejéből és a farokból képezhető párok $h_{h_i}(n - h_i)$ szorzata lehet, azaz a (42) egyenlőtlenségnek teljesednie kell.

Hasonló gondolatmenettel adódik megkapjuk (43)-t is.

Ugyanakkor annak is teljesednie kell, hogy sem a fej, sem a farok nem lépi túl a saját kapacitását, azaz $2X_3 \leq \min(2(X_1 + X_2), H_i, T_n - T_i)$. Legyen $Z = \min(2(X_1 + X_2), H_i, T_n - T_i)$.

Ehhez jön a fej elejének X_2-vel becsült további hozzájárulása és a fej végének X_3-mal becsült további hozzájárulás

X_1 nem lehet nagyobb, mint a fej elejében (H_{h_i}), illetve a fej végében $(H_i - H_{h_i})$ lévő elemek összegének kétszerese. Ugyancsak igaz, hogy a fej elejéhez tartozó h_i elem egyenként legfeljebb b_{h_i+1}, míg a fej végéhez tartozó $i - h_i$ elem egyenként legfeljebb b_1 fokkal kapcsolódhat a fej másik részéhez. És az is igaz, hogy a két rész elemeiből legfeljebb $h_i(i - h_i)$ pár képezhető.

A fej elejének a külsőt kiegészítő belső hozzájárulását egyrészt a lehetséges belső párok $\binom{h_i}{2}$ száma, másrészt a külső hozzájárulás után megmaradt szabad kapacitás korlátozza. A fej végének kiegészítő hozzájárulását hasonlóképpen becsülhetjük a $\min((i - h_i)(i - h_i - 1), H_i - H_{i+1} - X_1/2)$ kifejezéssel. $\square$

A 25. lemmában javasolt tesztet a következő algoritmus végzi.

Bemenet. n: a csúcsok száma $(n \geq 1)$;
$b = (b_1, b_2, \ldots, b_n)$: n-faroktesztelt sorozat;
$H = (H_1, \ldots, H_n)$: H_i a b sorozat első i elemének összege;
$T = (T_1, \ldots, T_n)$: T_i a b sorozat utolsó $n - i$ elemének összege.

Kimenet. 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).
β: a megvizsgált pozitív sorozatok közül elfogadott sorozatok száma.

Munkaváltozók. i: ciklusváltozó
$X = (X_1, X_2, X_3)$: ahol X_1 a fej elejének és végének közös kapacitására adott felső korlát, X_2 a fej elejének belső kapacitására, X_3 pedig a fej végének belső kapacitására adott felső korlát;
p: a b sorozatban lévő lévő pozitív elemek összege.

FEJFELEZŐ-TESZT$(n, b, H, T, p, L, \varphi)$
01 $\varphi = 0$

for $i = 2$ **to** $n - 1$
    $h = \lfloor i/2 \rfloor$
    $X_1 = 2\min(H_h, H_i - H_h, \min(h, b_{h+1})\min((i-h), b_1)), i(i-h))$
    $X_2 = \min(H_h - X_1/2, h(h-1))$
    $X_3 = \min(H_i - H_h - X_1/2, (n-h)(n-h-1))$
    if $H_i - (X_1 + X_2 + X_3) > \min(T_i, i(p-i))$
        $L = $ FALSE
        return L
$\varphi = \varphi + 1$
$L = $ TRUE
return L, φ

Az algoritmus futási ideje legjobb esetben $\Theta(1)$, legrosszabb esetben $\Theta(n)$.

Hasonló módon a farok felezése is további sorozatokat kiszűrését tenné lehetővé, de a szimulációs kísérletek tanúsága szerint ez nem csökkentené a várható futási időt.

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

5.7. Összetett teszt

Az ÖSSZETETT-TESZT algoritmus egymás után alkalmazza az eddig ismertetett teszteket a PARITÁS-TESZT, BINOMIÁLIS-TESZT, POZITÍV-TESZT, FEJFELEZŐ-TESZT, FAROKFELEZŐ-TESZT sorrendben.

ÖSSZETETT-TESZT$(n, b, \epsilon, \beta, \varphi, L)$

$\epsilon = \beta = \pi = \varphi = \tau = 0$
BINOMIÁLIS-TESZT(L, ϵ, β)
FEJFELEZŐ-TESZT(L, φ)
$L = $ TRUE
return $\epsilon, \beta, \varphi, L$

Az érdekesség kedvéért megjegyezzük, hogy ha $k > 0$, akkor egyik tesztelő algoritmus sem szűrné ki a (k) alakú sorozatokat, mivel azonban ezek *nem* szabályos sorozatok, így bemenetként nem fordulhatnak elő.

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

6. Közelítő algoritmusok hatékonysága és futási ideje

A tesztek elemzésénél a szabályos és páros sorozatokat vettük alapul. A páros sorozatok halmaza a legkisebb olyan halmaz, melynek elemszámát explicit képlettel meg tudjuk adni. Az $n-1 \geq b_i \geq 1$ feltételeknek eleget tevő *n-korlátos sorozatok* halmazának elemszámát is könnyű megadni, de ezen halmazok elemszáma túl gyorsan nő n növekedtével. A szabályos sorozatok elemzéséhez szerencsére nem kell *minden* korlátos sorozatot előállítani: elegendő a szabályos sorozatokat előállítani, és a rájuk vonatkozó 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 $(n, n-1, \ldots, 1, 0)$ „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 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. A második típusra pedig példa, hogy

Az első típusú javításra példa az Erdős-Gallai algoritmus ugrása. A második típusra pedig példa a Havel-Hakimi algoritmus kiegészítése előzetes paritásvizsgálattal, valamint az Erdős-Gallai algoritmus kiegészítése nullamentesítéssel.

A futási idők csökkentése érdekében *minden* algoritmus csak a páros, nullamentes sorozatokat vizsgálta.

Adott A algoritmusnak az n hosszúságú szabályos sorozatokra vonatkozó hatékonyságát az A algoritmus által kizárt n hosszúságú sorozatok és az ugyanolyan hosszúságú szabályos sorozatok számának hányadosával jellemezzük. Ezt a hányadost $\mathrm{R_A}(n)$-nel jelöljük és az A algoritmus n hosszúságú sorozatokra vonatkozó *hatékonyságának* nevezzük (lásd a 3. ábrát).

A következő közelítő algoritmusokat vizsgáljuk:

1) Nullamentesítő-teszt

2) Binomiális-teszt

3) Fejfelező-teszt

A szabályos sorozatoknak aszimptotikusan a fele páros sorozat. Az 1. ábra az n-szabályos sorozatok számát megadó 9, valamint az n-páros sorozatok számát megadó 11 lemmák alapján mutatja a konvergencia gyorsaságát (ami a képletek alapján természetes).

A 2. ábra a Nullamentes-teszt, Binomiális-teszt, Fejfelező-teszt és az Erdős-Gallai-Gyorsan programok futási eredményeit tartalmazza.

2 022 337 118 015 338

A 3. ábra a nullamentes, binomiális és fejtesztelt sorozatoknak a szabályos sorozatokhoz viszonyított részarányát mutatja be.

A 4. ábra a közelítő algoritmusok elemenkénti átlagos futási idejét és műveletszámát tartalmazza. Az algoritmusok bemenete az összes páros nullamentes sorozat volt. A mikromásodpercben mért idő és a műveletszám is tartalmazza a sorozatok előállítására fordított időt és műveletszámot is.

Ha $n = 2$, akkor (45) szerint $\rho(n) = \binom{3}{2} = 3$ szabályos sorozat van: $(1,1)$, $(1,0)$

n	$\rho(n)$	$\epsilon(n)$	$\epsilon(n)/\rho(n)$
1	1	1	1.0000000000000
2	3	2	0.6666666666667
3	10	6	0.6000000000000
4	35	19	0.5428571428571
5	126	66	0.5238095238095
6	462	236	0.5108225108225
7	1716	868	0.5058275058275
8	6435	3235	0.5027195027195
9	24310	12190	0.5014397367339
10	92378	46252	0.5006819805581
11	352716	176484	0.5003572279114
12	1352078	676270	0.5001708481315
13	5200300	2600612	0.5000888410284
14	20058300	10030008	0.5000427753100
15	77558760	38781096	0.5000221251603
16	300540195	150273315	0.5000107057227
17	1166803110	583407990	0.5000055150693
18	4537567650	2268795980	0.5000026787479
19	17672631900	8836340260	0.5000013755733
20	68923264410	34461678394	0.5000006701511
21	269128937220	134564560988	0.5000003432481
22	1052049481860	526024917288	0.5000001676328
23	4116715363800	2058358034616	0.5000000856790
24	16123801841550	8061901596814	0.5000000419280
25	63205303218876	31602652961516	0.5000000213918
26	247959266474052	123979635837176	0.5000000104862
27	973469712824056	486734861612328	0.5000000053420
28	3824345300380220	1912172660219260	0.5000000026224
29	15033633249770520	7516816644943560	0.5000000013342
30	59132290782430712	29566145429994736	0.5000000006558
31	232714176627630544	116357088391374032	0.5000000003333
32	916312070471295267	458156035385917731	0.5000000001640
33	3609714217008132870	1804857108804606630	0.5000000000833
34	14226520737620288370	7113260369393545740	0.5000000000410
35	56093138908331422716	28046569455332514468	0.5000000000208
36	221256270138418389602	110628135071477978626	0.5000000000103
37	873065282167813104916	436532641088444120108	0.5000000000052
38	3446310324346630677300	1723155162182151654600	0.5000000000026
39	13608507434599516007800	6804253717317430635800	0.5000000000013
40	53753604366668088230810	26876802183368505747610	0.5000000000006

1. ábra. Páros és szabályos sorozatok száma és ezen számok aránya.

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

n	$\zeta(n)$	$\beta(n)$	$\varphi(n)$	$\gamma(n)$	$\gamma(n+1)/\gamma(n)$
1	**0**	**1**	**0**	**1**	2.000000
2	**1**	**2**	**2**	**2**	2.000000
3	**4**	**4**	**4**	**4**	2.750000
4	**11**	**4**	**11**	**11**	2.818182
5	???	**31**	**31**	**31**	3.290323
6	???	103	**102**	**102**	3.352941
7	???	349	343	**342**	3.546784
8	???	1256	???	1213	3.595218
9	???	4577	???	4361	3.672552
10	???	17040	???	16016	3.705544
11	???	63944	???	59348	3.742620
12	???	242218	???	222117	3.765200
13	???	922369	???	836315	3.786674
14	???	???	???	3166852	3.802710
15	???	???	???	12042620	3.817067
16	???	???	???	45967479	3.828918
17	???	???	???	176005709	3.839418
18	???	???	???	675759564	3.848517
19	???	???	???	2600672458	3.856630
20	???	???	???	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	3.898843
29	???	???	???	2022337118015338	3.902378

2. ábra. Nullamentes, binomiális, fejfelező és jó sorozatok száma, valamint a jó sorozatok szomszédos helyeken vett értékeinek hányadosa.

$\beta(n)$-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) $R_{\mathrm{BT}}(2) = 2/2 = 1$.

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 $\epsilon(3) = 6$. Ezek közül a Binomiális-teszt kizárja a (2,2,0) és (2,0,0) sorozatokat, így $\beta(3) = 4$. A megmaradt 4 sorozat jó, így $\varphi(3) = \tau(3) = \gamma(3) = 4$.

Ha $n = 4$, akkor a szabályos sorozatok száma $\rho(4) = 35$. Ezek közül 19 a páros, és a következő 11 jó: (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),

n	$\zeta(n)/\rho(n)$	$\beta(n)/\rho(n)$	$\varphi(n)/\rho(n)$	$\gamma(n)/\rho(n)$
1	0	1.00000	1.00000	1.00000
2	0.33333	0.66667	0.66667	0.66667
3	0.40000	0.40000	0.40000	0.40000
4	0.31429	0.31429	0.31429	0.31429
5	???	0.24603	0.24603	0.24603
6	???	0.22294	0.22078	0.22078
7	???	0.20338	0.19988	0.19930
8	???	0.19518	???	0.18850
9	???	0.18828	???	0.17939
10	???	0.18446	???	0.17337
11	???	0.18129	???	0.16826
12	???	0.17914	???	0.16428
13	???	???	???	0.16082
14	???	???	???	0.15788
15	???	???	???	0.15527
16	???	???	???	0.15295
17	???	???	???	0.15084
18	???	???	???	0.14893
19	???	???	???	0.14716
20	???	???	???	0.14552
21	???	???	???	0.14400
22	???	???	???	0.14257
23	???	???	???	0.14123
24	???	???	???	0.13996
25	???	???	???	???

3. ábra. A nullamentes, binomiális, fejtesztelt és jó sorozatok számának részaránya a szabályos sorozatokhoz viszonyítva.

(2,1,1,0), (1,1,1,1), (1,1,0,0) és (0,0,0,0). A 19 páros sorozat közül Binomiális-teszt is kizárja azt a nyolc sorozatot, amelyeket az Erdős-Gallai kizárna, így $\beta(4) = \gamma(4) = \varphi(4) = \gamma(4) = 11$.

A $\rho(5) = 126$ szabályos sorozat közül $\epsilon(5) = 66$ a páros, ezek között pedig $\beta(5) = 31$ a binomiális. Ezek a sorozatok mind jók, azaz $\varphi(5) = \tau(5) = \gamma(5) = 31$.

A $\rho(6) = 462$ szabályos sorozat közül $\epsilon(6) = 236$ a páros, amelyek között $\beta(6) = 103$ binomiális sorozat van. Binomiális-teszt a 102 jó sorozat mellett az (5,5,3,3,3,1) rossz sorozatot is elfogadja. Ezek szerint a legfeljebb 5 hosszúságú sorozatokra nézve a Binomiális-teszt hibátlanul kiszűri a nem jó sorozatokat, a 6 hosszú sorozatokra azonban már csak közelítő algoritmus. A Fejfelező-teszt ezzel a sorozattal is megbirkózik, ezért $\varphi(6) = \tau(6) = \gamma(6) = 102$.

??????????????????????????????????????Eddig jó A $\rho(7) = 1716$ szabályos sorozat között $\epsilon(6) = 868$ a páros, melyek közül $\beta(7) = 376$ a binomiális. A binomiális sorozatok között még 34 rossz van, melyek közül a Pozitív-teszt a 27 jó sorozat mellett a következő 7 rosszat is elfogadja: $(6, 6, 6, 4, 4, 4, 2)$, $(6, 6, 5, 4, 4, 4, 1)$,

n	Binomiális, μs	Binomiális, művelet	Fejfelező, μs	Fejfelező, művelet
1	???	???	???	
2	???	???	???	
3	???	???	???	
4	???	???	???	
5	???	???	???	
6	???	???	???	
7	???	???	???	
8	???	???	???	
9	???	???	???	
10	???	???	???	
11	???	???	???	
12	???	???	???	
13	???	???	???	
14	???	???	???	
15	???	???	???	
16	???	???	???	
17	???	???	???	
18	???	???	???	
19	???	???	???	
20	???	???	???	

4. ábra. Közelítő algoritmusok elemenkénti átlagos műveletszáma.

$(6,6,4,4,4,3,1)$, $(6,6,4,3,3,3,1)$, $(6,6,3,3,3,2,1)$, $(6,5,3,3,3,1,1)$, $(5,5,3,3,3,1,0)$. A következő Fejfelező-teszt ezek közül a $(6,6,4,3,3,3,1)$ kivételével mindet kiszűri, így $\varphi(7) = 343$. A következő Farokfelező-teszt $i = 4$ mellett legfeljebb $8+2$ fokot tud lekötni a fej eleje és a farok részei között, legfeljebb további $4+0$ fokot a fej vége és a farok részei között, legfeljebb további 8 fokot a fej két része között, és két fokot a fej elején belül. Ez azonban összesen csak $10+4+8+2 = 24$ fok, ami kevesebb a sorozat $H_7 = 26$ összfokszámánál. Tehát $\tau(7) = \gamma(7) = 342$.

A 2. ábrán minden sorban az első pontos értéket félkövéren írtuk. Eszerint $n \leq 4$ 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.

Pozitív-teszt $n = 5$-ig hibátlan, a Fejfelező-teszt $n = 6$-ig, a Farokfelező-teszt pedig $n = 7$-ig.

A 3. ábrán $\rho(n)$ értéke $n = 24$-ig az EIS A001700 sorozata [67], $\epsilon(n)$ értéke $n = 23$-ig az EIS A005654 sorozata [69], $\gamma(n)$ értéke pedig $n = 23$-ig az EIC A0004251-es sorozata [68]. A többi értéket mi határoztuk meg: sem $\rho(25), \dots \rho(40)$, sem $\epsilon(24), \dots, \epsilon(40)$, sem a $\beta(n)$, $\pi(n)$, $\varphi(n)$ és $(\tau(n)$ értékek nem szerepelnek az

EIS-ben.

Ebben a cikkben első sorban a soros algoritmusokkal kapott eredményekről számoltunk be.

A témakörben vannak párhuzamos eredmények is [53, 63, 70]. Saját párhuzamos eredményeinket a 8. részben ismertetjük.

7. Pontos algoritmusok futási ideje

A pontos algoritmusok sorozatonkénti átlagos futási idejét n függvényében mikromásodpercben az alábbi 5. ábra tartalmazza. Az algoritmusok csak a páros, nullamentes sorozatokat vizsgálták. A futási idő tartalmazza a sorozatok előállítási idejét is sorozatok előállítási idejét

n	HH	HHE	EG	EGU	EGL
1	???	???	???	???	???
2	???	???	???	???	???
3	???	???	???	???	???
4	???	???	???	???	???
5	???	???	???	???	???
6	???	???	???	???	???
7	???	???	???	???	???
8	???	???	???	???	???
9	???	???	???	???	???
10	???	???	???	???	???
11	???	???	???	???	???
12	???	???	???	???	???
13	???	???	???	???	???
14	???	???	???	???	???
15	???	???	???	???	???
16	???	???	???	???	???

5. ábra. Pontos algoritmusok elemenkénti átlagos futási ideje műveletszámban.

A következő pontos algoritmusokat vizsgáljuk:

1) HH: Rendező Havel-Hakimi algoritmus (HH).

2) HHE: Eltoló Havel-Hakimi algoritmus (HHE).

3) EG: Erdős-Gallai algoritmus (EG).

4) EGU: Erdős-Gallai algoritmus ugrásokkal (EGU).

5) EGL: Erdős-Gallai algoritmus ugrásokkal lineárisan (EGL).

A pontos algoritmusok sorozatonkénti átlagos futási idejét n függvényében mikromásodpercben a 5. ábra tartalmazza.

Az EG-Lineárisan-Mind algoritmus teljes és az elemenkénti futási idõit tartalmazza a 6. ábra.

n	$\epsilon(n)$	$\tau(n)$, mp	$Op(n)$	$\tau(n)/\epsilon(n)/n$, mp	$Op(n)/\epsilon(n)/n$
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	???	???	???	???	???

6. ábra. Az EG-Lineárisan algoritmus elemenkénti futási ideje műveletszámban kifejezve.

A 7. ábra azt mutatja, hányszor van az Erdős-Gallai-Lineárisan algoritmusnak $1, \ldots, n-1$ menetre szüksége, hogy a rossz sorozatokat kiszűrje.

Az 8. ábrán az látható, hogy az EG-Lineárisan ciklusmagja átlagosan (gyakoriságokkal súlyozva) hányszor futott le, ha csak a rossz sorozatokat vesszük figyelembe, illetve a ha minden sorozatot figyelembe veszünk (mindkét esetben csak a páros, nullamentes sorozatokat vettük figyelembe).

A 9. ábra pedig a jó sorozatok kezdő elem szerinti megoszlását mutatja be. Ezek az adatok segítenek a $\gamma(n)$ számítását végző Erdős-Gallai-Gyorsan program párhuzamos megvalósításánál (a számítások több részre osztásában).

8. Grafikus sorozatok száma

A 2. ábra 1-től 28 csúcsig tartalmazza a grafikus sorozatok számát. A táblázat úgy készült, hogy párhuzamosítottuk az Erdős-Gallai-Gyorsan algoritmust. Mivel

n 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
3 2 0 0
4 8 0 0 0
5 33 2 0 0 0
6 122 12 0 0 0 0
7 459 65 2 0 0 0 0
8 1709 289 24 0 0 0 0 0
9 6421 1228 176 4 0 0 0 0 0
10 24205 4951 1013 67 0 0 0 0 0 0
11 91786 19603 5126 610 11 0 0 0 0 0 0
12 349502 76414 23755 4274 208 0 0 0 0 0 0 0
13 1336491 296036 104171 25293 2277 29 0 0 0 0 0 0 0
14 5128246 1142470 439155 133946 18673 666 0 0 0 0 0 0 0 0
15 19739076 4404813 1803496 655291 127116 8603 81 0 0 0 0 0 0 0 0
Ezek az UGRÓ LEG adatai

7. ábra. Az EG-Lineárisan elutasítás előtti meneteinek száma.

n=3 11 /6 /3 = 0.611111
n=4 47 /19 /4 = 0.618421
n=5 192 /66 /5 = 0.581818
n=6 792 /236 /6 = 0.559322
n=7 3229 /868 /7 = 0.531435
n=8 13268 /3235 /8 = 0.512674
n=9 54244 /12190 /9 = 0.494431
n=10 222057 /46252 /10 = 0.480102
n=11 906558 /176484 /11 = 0.466979
n=12 3698529 /676270 /12 = 0.455751
n=13 15063277 /2600612 /13 = 0.445554
n=14 61286926 /10030008 /14 = 0.436454
n=15 249056158 /38781096 /15 = 0.428140
n=16 1011175412 /150273315 /16 = 0.420557
n=17 4101727713 /583407990 /17 = 0.413567
n=18 16625570580 /2268795980 /18 = 0.407107

8. ábra. Az EG-Lineárisan menetszámai súlyozott átlagának n-ed része, ha a jó sorozatokat is figyelembe vesszük, illetve ha nem.

viszonylag sok processzor vett részt a számolásban, viszont bizonytalan volt, hogy az egyes processzorok meddig vehetnek részt a számolásban, a feladatot *szeleteknek* nevezett kisebb részekre bontottuk. Célszerű volt, hogy a szeletek feldolgozása hasonló ideig tartson.

Mivel a futási idő csökkentése érdekében az Erdős-Gallai-Gyorsan algoritmus csak nullamentes sorozatokat állít elő és tesztel, a szeletekre bontás alapja a 9. lemma alábbi következménye.

26. Következmény. *Ha Ha m és u pozitív egész számok, akkor az $(1, u, m)$-szabályos*

n/b_1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1				1	1	1	1	1	1	1	1	1	1	1
2	1	1				1	1	1	1	1	1	1	1	1	1
3	1	1	2				1	1	1	1	1	1	1	1	1
4	1	1	4	4					1	1	1	1	1	1	1
5	1	2	7	10	11				1	1	1	1	1	1	1
6	1	3	10	22	35	31				1	1	1	1	1	1
7	1	3	14	34	78	110	102				1	1	1	1	1
8	1	4	18	54	138	267	389	342				1	1	1	1
9	1	4	23	74	223	503	968	1352	1213				1	1	1
10	1	15	28	104	333	866	1927	3496	4895	4361					1
11	1	5	34	134	479	1356	3471	7221	12892	17793	16016				
12				1	1	1	1	1	1	1	1	1	1	1	1
13				1	1	1	1	1	1	1	1	1	1	1	1
14				1	1	1	1	1	1	1	1	1	1	1	1

9. ábra. A jó sorozatok b_1 szerinti eloszlása.

sorozatok $\rho(1,u,m)$ *száma*

$$\rho(1,u,m) = \binom{m+u-1}{m}. \tag{45}$$

Bizonyítás. A 9. lemmában alkalmazzuk az $l = 1$ helyettesítést. □

Feltételezzük, hogy az n-szabályos nullamentes sorozatok halmazának szeletekre való felbontásánál az egyes szeletek futási ideje arányos a hozzájuk tartozó $\rho(1,u,m)$-szabályos sorozatok számával.

Nézzünk egy példát. Legyen $n = 6$. Ekkor a most bizonyított 26. következmény szerint az $(1,5,6)$-szabályos sorozatok száma $\binom{10}{6} == 210$. Tegyük fel, hogy a 210 sorozatot tartalmazó halmazt 8 hasonló méretű szeletre akarjuk bontani. Ekkor egy szelet átlagos mérete $210/8 = 26.25$ sorozat. Ennek felső egész része 27. Ez lesz a szeletek maximális megengedett mérete. A 210 $(1,5,6)$-szabályos sorozat kezdőelem szerinti megoszlását a 10. ábra tartalmazza.

i	$\rho(1,i,6) = \binom{4+i}{5}$
1	1
2	6
3	21
4	56
5	126

10. ábra. Az $(1,5,6$-szabályos sorozatok megoszlása kezdőelem szerint.

Mivel $1+6 \leq 27$ és $1+6+21 \geq 27$, az 1-gyel és 2-vel kezdődő sorozatok kerülnek az S_1 szeletbe. Mivel $21 \leq 27$ és $21+56 \geq 27$, a 3-mal kezdődő sorozatok alkotják az S_2 szeletet. Mivel $\lceil 56/27 \rceil = 3$, a 4-gyel kezdődő sorozatokat legalább 3 szeletre bontjuk. Végül $\lceil 126/27 \rceil = 5$, így az öttel kezdődő sorozatokhoz legalább 5 szeletet rendelünk. Ez idáig legalább 10 szelet.

i	$\rho(1,i,5) = \binom{4+i}{5}$
1	1
2	5
3	15
4	35

11. ábra. A 4-gyel kezdődő $(1,4,6$-szabályos sorozatok megoszlása második elemük szerint.

A 12. ábra a 4-gyel kezdődő, $(1,4,6)$ sorozatok számának a második elem szerinti eloszlását mutatja. Eszerint az S_3 szeletbe a $(4,1)$, $(4,2)$ és $(4,3)$ kezdetű sorozatok kerülnek (összesen 21 sorozat), míg a $(4,4)$ kezdetű sorozatok legalább két szeletet elfoglalnak.

A 12. ábra mutatja a $(4,4)$-gyel kezdődő sorozatok harmadik elem szerinti eloszlását.

i	$\rho(1,i,4) = \binom{2+i}{3}$
1	1
2	4
3	10
4	20

12. ábra. A $(4,4)$-gyel kezdődő, $(1,4,6$-szabályos sorozatok megoszlása harmadik elemük szerint.

A 12. ábra alapján az S_4 szeletbe a $(4,4,1)$, $(4,4,2)$ és $(4,4,3)$ kezdetű, míg az S_5 szeletbe a $(4,4,4)$ kezdetű sorozatok kerülnek.

A 13. ábra mutatja az 5-tel kezdődő $(1,5,6)$-szabályos sorozatok második elem szerinti eloszlását.

A 13. ábra alapján az S_6 szeletbe az $(5,1)$, $(5,2)$ és $(5,3)$ kezdetű sorozatok (összesen 21 sorozat) kerülnek. A 14. ábra az $(5,4)$ kezdetű sorozatok harmadik elem szerinti eloszlását mutatja.

A 14. ábra szerint az S_7 szelet az $(5,4,1)$, $(5,4,2)$ és $(5.4,3)$, míg az S_8 szelet az $(5,4,4)$ kezdetű sorozatokat tartalmazza. A 14. ábra az $(5,5)$-tel kezdődő sorozatok harmadik elem szerinti megoszlását mutatja.

A 15. ábra szerint az S_9 szelet az $(5,5,1)$, $(5,5,2)$ és $(5,5,3)$ kezdetű, az S_10 szelet pedig az $(5,5,4)$ kezdetű sorozatokat tartalmazza. A 16. ábra az $(5,5,5)$-tel kezdetű

i	$\rho(1,i,4) = \binom{3+i}{4}$
1	1
2	5
3	15
4	35
5	70

13. ábra. Az (5)-tel kezdődő $(1,5,5)$-szabályos sorozatok megoszlása masodik elemük szerint.

i	$\rho(1,i,3) = \binom{2+i}{3}$
1	1
2	4
3	10
4	20

14. ábra. Az (5,4)-gyel kezdődő $(1,4,4)$-szabályos sorozatok megoszlása harmadik elemük szerint.

i	$\rho(1,i,3) = \binom{2+i}{3}$
1	1
2	4
3	10
4	20
5	35

15. ábra. Az (5,5)-tel kezdődő $(1,5,4)$-szabályos sorozatok megoszlása harmadik elemük szerint.

sorozatok negyedik elem szerint megoszlását mutatja.

A 16. ábra szerint az S_{11} szelet az $(5,5,5,1)$, $(5,5,5,2)$, $(5,5,5,3)$ és $(5,5,5,4)$ kezdetű sorozatokat, míg az $\mathrm{S}(_{12}$ szelet az $(5,5,5,5)$ kezdetű sorozatokat tartalmazza.

A 17. ábra az $(1,5,6)$-szabályos sorozatok 12 szeletre bontását mutatja.

Most tekintsünk egy másik példát: legyen $n = 29$. Az $n = 28$ esetben szerzett tapasztalatok alapján feltesszük, hogy a tiszta futási idő összesen 2500 nap lesz. Feltételezve, hogy a gépek egy részét csak éjszakára kapjuk meg, legyen egy szelet maximális futási ideje 12 óra. Ekkor egyenletes eloszlás mellett 5000 szelet lenne.

A 29-szabályos, nullamentes sorozatok száma a 26. következmény és a Stirling

i	$\rho(1,i,2) = \binom{1+i}{2}$
1	1
2	3
3	6
4	10
5	15

16. ábra. Az (5,4)-gyel kezdődő $(1,5,3)$-szabályos sorozatok megoszlása a harmadik elem szerint.

S_i	szelet kezdő sorozata	szelet elemszáma
S_1	$(2,2,2,2,2,2)$	7
S_2	$(3,3,3,3,3,3)$	21
S_3	$(4,3,3,3,3,3)$	21
S_4	$(4,4,3,3,3,3)$	15
S_5	$(4,4,4,4,4,4)$	20
S_6	$(5,3,3,3,3,3)$	21
S_7	$(5,4,3,3,3,3)$	15
S_8	$(5,4,4,4,4,4)$	20
S_9	$(5,5,3,3,3,3)$	15
S_{10}	$(5,5,4,4,4,4)$	20
S_{11}	$(5,5,5,4,4,4)$	20
S_{12}	$(5,5,5,5,5,5)$	15

17. ábra. A nullamentes 6-szabályos sorozatok felbontása 12 szeletre.

formula szerint

$$\nu(1,28,29) = \binom{56}{29} = \frac{56!}{29!27!} \sim \left(\frac{56}{e}\right)^{56} \left(\frac{e}{29}\right)^{29} \left(\frac{e}{27}\right)^{27} \left(\frac{\sqrt{2\pi 56}}{\sqrt{2\pi 29}\sqrt{2\pi 27}}\right). \quad (46)$$

A ??? nullamentes, $(1,28,29)$-szabályos sorozat kezdőelem szerinti megoszlását a 18. ábra tartalmazza.

???

A $\gamma(n)$ függvény nagyságrendjére vonatkozó legpontosabb eredmény Jason Matthew Burnsnek Richard P. Stanley témavezetésével készített és a Massachusetts Institute of Technology Matematikai Tanszékén 2007-ben megvédett doktori értekezésében található. Eszerint léteznek olyan pozitív c, C és k állandók, hogy a jó sorozatok

i	$\rho(1, i, 29) = \binom{27+i}{28}$
1	1
2	29
3	435
4	4495
5	81840
6	???
7	???
8	???
9	???
10	???
11	???
12	???
13	???
14	???
15	???
16	???
17	???
18	???
19	???
20	???
21	???
22	???
23	???
24	???
25	???
26	???
27	???
28	???

18. ábra. A $(1, 28, 29)$-szabályos sorozatok megoszlása kezdőelem szerint.

$\gamma(n)$ száma a következő korlátok közé esik:

$$\frac{c4^n}{n} < \gamma(n) < \frac{C4^n}{(\log n)^k \sqrt{n}} \ . \tag{47}$$

Tegyük fel, hogy a felső becslés nagyságrendje alkalmas k esetén pontos, azaz léteznek olyan pozitív C és k_n számok, hogy

$$\gamma(n) = \frac{C4^n}{(\log n)^{k_n} \sqrt{n}} + o\left(\frac{4^n}{(\log n)^{k_n} \sqrt{n}}\right), \tag{48}$$

ahol n növekedtével a k_n sorozat konvergens.

Ekkor

$$\frac{\gamma(n+1)}{\gamma(n)} = 4\ \frac{\sqrt{n}}{\sqrt{n+1}}\ \frac{(\log n)_n^k}{(\log(n+1))_n^k}, \tag{49}$$

ahonnan a $k_{n+1} = k_n = k$ helyettesítéssel azt kapjuk, hogy

$$k = \frac{\log \frac{\gamma(n+1)}{\gamma(n)} - \log 4 - \log \frac{\sqrt{n}}{\sqrt{n+1}}}{\log \frac{\log n}{\log(n+1)}}. \tag{50}$$

A 19. ábra k számításának részeredményeit tartalmazza.

n	$\ln \frac{\gamma(n+1)}{\gamma(n)}$	$\ln \frac{\sqrt{n}}{\sqrt{n+1}}$	$\ln \frac{\ln n}{\ln(n+1)}$	k
2	0.3010	0.0880	-0.2000	1.064820717840770
3	0.4393	0.0625	-0.1010	0.992544624686593
4	0.4500	0.0485	-0.0649	1.598857474425710
5	0.5172	0.0396	-0.0466	0.970505819319945
6	0.5254	0.0335	-0.0358	1.204156239958088
7	0.5496	0.0290	-0.0288	0.805904015852462
8	0.5557	0.0256	-0.0240	0.867547492228616
9	0.5650	0.0229	-0.0204	0.698741682404903
10	0.5688	0.0207	-0.0176	0.710301436698686
11	0.5732	0.0189	-0.0155	0.645354379756533
12	0.5758	0.0174	-0.0137	0.645728379631260
13	0.5783	0.0161	-0.0124	0.623261906868982
14	0.5801	0.0150	-0.0113	0.623236569060196
15	0.5817	0.0140	-0.0101	0.617468512786572
16	0.5831	0.0132	-0.0094	0.619497652173067
17	0.5843	0.0124	-0.0087	0.620531345029615
18	0.5853	0.0117	-0.0081	0.624435032861388
19	0.5862	0.0111	-0.0075	0.628464622512213
20	0.5870	0.0106	-0.0070	0.633641383390237
21	0.5877	0.0101	-0.0067	0.639087464420730
22	0.5884	0.0097	-0.0062	0.645030554474696
23	0.5891	0.0092	-0.0059	0.651169668041924
24	???	???	???	???
25	???		???	???

19. ábra. k becslése.

A 19. ábra szerint az első négy oszlop szigorúan monoton (ami a megfelelő képletek alapján természetes), míg a k értékekre $(n = 2)$-től $(n = 13)$-ig az jellemző, hogy a páratlan n-hez tartozó k értékek *kisebbek* a szomszédos $n-1$ és $n+1$ értékhez tartozó k-nál. Ha pedig $n \geq 16$, akkor a k érték *nagyobb*, mint az $n-1$ csúcsszámhoz tartozó k.

Hasonló lokális minimumot találtak például a [30] cikk szerzői véletlen mátrixok vizsgálata során.

Köszönetnyilvánítás. A szerzők köszönik Móri Tamás docensnek (ELTE TTK) a 17. és a 18. lemmák bizonyítását, Kása Zoltán egyetemi tanárnak (Sapientia Erdélyi Magyar Tudományegyetem) a kézirat javítására vonatkozó javaslatait, Mányoki Ádámnak (TFM World Kereskedelmi és Szolgáltató Kft.) és Sándor Antal programtervező maetmatikusnak pedig a programok nagy processzorigényű futtatásához nyújtott segítséget. A kutatás az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg (a támogatás száma TÁMOP 4.2.1/B-09/1/KMR-2010-0003).

Hivatkozások

[1] ASCHER, M.: *Mu torere: an analysis of a Maori game.* Math. Mag. **60(2),** (1987) 90–100. ⇒ 13

[2] AVIS, D., FUKUDA, K. *Reverse search for enumeration.* Discrete Appl. Math. **2,** (1993) 21–46. ⇒ 12

[3] BARNES, T. M., SAVAGE, C. D.: *A recurrence for counting graphical partitions.* Electron. J. Combin. **2,** (1995) R11, 10 pp. ⇒ 12, 16

[4] BARNES, T. M., SAVAGE, C. D. *Efficient generation of graphical partitions. Discrete Appl. Math.* **78(1–3),** (1997) 17–26. ⇒ 1, 2, 12, 16, 41

[5] BEASLEY, L. B., BROWN D. E., REID, K. B.: *Extending partial tournaments.* Math. Comput. Modelling **50(1),** (2009) 287–291. ⇒ 2

[6] BEREG S., ITO, H.: *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] BOZÓKI, S., FÜLÖP, J., POESZ, A.: *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. ⇒ 1

[8] BOZÓKI S., FÜLÖP J., RÓNYAI, L.: *On optimal completion of incomplete pairwise comparison matrices.* Math. Comput. Modelling **52,** (2010) 318–333. ⇒ 1, 2

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

[10] BURNS, J. M.: *The number of degree sequences.* PhD Dissertation, MIT, 2007. ⇒ 2, 12, 16

[11] BUSCH A. N., CHEN G., JACOBSON M. S.: *Transitive partitions in realizations of tournament score sequences.* J. Graph Theory **64(1),** (2010), 52–62. ⇒ 2

[12] CORMEN, T. H., LEISERSON, CH. E., RIVEST, R. L., STEIN, C.: *Introduction to Algorithms.* Third edition, The MIT Press/McGraw Hill, Cambridge/New York, 2009. Magyarul: *Algoritmusok.* Műszaki Könyvkiadó, Budapest, 2003. ⇒ 3, 13

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

[14] DEL GENIO, C. I., KIM, H., TOROCZKAI, Z., BASSLER, K. E.: *Efficient and exact sampling of simple graphs with given arbitrary degree sequence.* PLoS ONE **5(4)**, e10012 (2010). ⇒ 2

[15] ERDŐS, P., GALLAI, T.: *Gráfok előírt fokú pontokkal.* Mat. Lapok **11,** (1960) 264–274. ⇒ 1, 3, 4, 41

[16] ERDŐS, P. L., MIKLÓS, I., TOROCZKAI, Z.: *A simple Havel-Hakimi type algorithm to realize graphical degree sequences of directed graphs.* Electron. J. Combin. **17(1),** (2010) R66, 10 pp. ⇒ 2, 3

[17] Erdős, P., Richmond L. B.: *On graphical partitions.* Combinatorica **13(1),** (1993) 57–63. ⇒2

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

[19] Frank, D. A., Savage, C. D., Sellers, J. A.: *On the number of graphical forest partitions. Ars Combin.* **65,** (2002) 33–37. ⇒12

[20] Hakimi, S. L.: *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, 3, 41

[21] Hakimi, S. L.: *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. ⇒2

[22] Havel, V.: A remark on the existence of finite graphs (cseh); *Časopis Pěst. Mat.* **80,** (1955), 477–480. ⇒1, 3, 41

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

[24] Iványi, A.: *Pontsorozatok ellenőrzése.* In: (szerk. Tóth B.) *XXV. Magyar Operációkutatási Konferencia* (Debrecen, 2001. október 17–20), 53–53. ⇒2

[25] Iványi, A.: *Reconstruction of complete interval tournaments.* Acta Univ. Sapientiae, Inform., **1(1),** (2009) 71-88. ⇒1, 2, 3

[26] Iványi, A.: *Reconstruction of complete interval tournaments. II.* Acta Univ. Sapientiae, Math., **2(1),** (2010) 47–71. ⇒1, 2, 3

[27] Iványi, A.: *Deciding the validity of the score sequence of a soccer tournament.* In (ed. A. Frank): Open problems of the Egerváry Research Group, Budapest, 2011. ⇒2

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

[29] Iványi, A., Lucz, L., Sótér, P.: *On the Erdős-Gallai and Havel-Hakimi algorithms.* Acta Univ. Sapientiae, Inform. (benyújtva). ⇒1

[30] Iványi A., Kátai, I.: *Testing of random matrices,* Acta Univ. Sapientiae, Math., **3(1),** (2011) 99–126. ⇒36

[31] Iványi, A., Madarász, J., Németh, Zs.: *Algoritmusok hatékonyságának oktatása.* In (szerk. Cser László és herdon Miklós) *Az Informatika'2011 előadásainak kivonatai* (Debrecen, 2011. augusztus 24–26). Debreceni Egyetem, Debrecen, 2011. ⇒2

[32] Iványi, A., Pirzada, S.: *Comparison based ranking.* In (ed. A. Iványi): Algorithms of Informatics, Vol. 3. AnTonCom, Budapest 2011, 1262–1311. ⇒1, 2

[33] Iványi, A., Pirzada, S.: *Hiperversenyek helyreállítása.* In: *XXIX. Magyar Operációkutatási Konferencia* (Balatonőszöd, 2001. október 17–20), 53–53. ⇒2

[34] Kátai, I.: *Szóbeli közlés.* Budapest, 2010. ⇒16

[35] Kayibi K., Khan M. A., Pirzada S., Iványi A.: Random sampling of minimally cyclic digraphs with given imbalance sequence. Acta Univ. Sapientiae, Math. (submitted). ⇒1, 2

[36] Kemnitz, A., Dolff, S.: *Score sequences of multitournaments,* Congr. Num., **127,** (1997) 85–95. ⇒2

[37] Kim, H., Toroczkai, Z., Miklós, I., Erdős, P. L., Székely, L. A.: Degree-based graph construction. J. Physics: Math. Theor. A **42(39),** (2009) 392–401. ⇒1, 2

[38] Kleitman, D. J., Wang, D. L.: *Algorithms for constructing graphs and digraphs with given valencies and factors.* Discrete Math. **6** (1973) 79–88. ⇒3

[39] Kleitman, D. J., Winston K. J.: *Forests and score vectors.* Combinatorica **1(1),** (1981) 49–54. ⇒12

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

[41] Kohnert, A.: *Dominance order and graphical partitions.* Elec. J. Comb. **11(1),** (2004) No. 4. 17 pp. ⇒2

[42] Kovács, G. Zs., Pataki, N.: *Rangsorolási algoritmusok elemzése.* TDK dolgozat. ELTE TTK, Budapest, 2002. 39 oldal. ⇒2

[43] LaMar, M. D. *Algorithms for realizing degree sequences of directed graphs.* arXiv-0906:0343ve [math.CO], 7 June 2010. ⇒2

[44] Landau, H. G.: *On dominance relations and the structure of animal societies. III. The condition for a score sequence.* Bull. Math. Biophys. **15,** (1953) 143–148. ⇒1

[45] Liljeros, F., Edling, C. R., Amaral, L., Stanley, H., Åberg, Y.: *The web of human sexual contacts.* Nature **411,** (2001) 907–908. ⇒1

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

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

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

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

[50] Moon, J. W.: *Topics on Tournaments.* Holt, Rinehart, and Winston, New York, 1968. ⇒12

[51] Móri, T.: Szóbeli közlés. Budapest, 2011. ⇒15, 16

[52] Newman, M. E. J., Barabási, A. L.: *The Structure and Dynamics of Networks.* Princeton University Press, Princeton, NJ. 2006. ⇒1

[53] Pécsy G., Szűcs, L.: *Parallel verification and enumeration of tournaments.* Stud. Univ. Babeş-Bolyai, Inform. **45(2),** (200) 11–26. ⇒28

[54] Pirzada, S.: *Graph Theory.* Orient Blackswan, 2011, to appear. ⇒2

[55] Pirzada, S., Al-Assaf, A. M., Kayibi, K. K.: *On imbalances in oriented multipartite graphs.* Acta Univ. Sapientiae, Math., **3(1),** (2011) 34–42. ⇒2

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

[57] Pirzada, S., Iványi, A., Khan, M. A.: *Score sets and kings.* In (ed. A. Iványi): Algorithms of Informatics, Vol. 3, ed. A. Iványi. AnTonCom, Budapest 2011, 1451–1490. ⇒1, 2

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

[59] Pirzada, S., Zhou G., Iványi A.: *On k-hypertournament losing scores,* Acta Univ. Sapientiae, Inform. **2(2),** (2010) 184–193. ⇒1, 2

[60] Rødseth, Ø. J., Sellers, J. A., Tverberg, H.: *Enumeration of the degree sequences of non-separable graphs and connected graphs.* European J. Comb. **30(5),** 1309–1319. ⇒2, 12

[61] Ruskey, F., Cohen, R., Eades, P., Scott, A.: *Alley CAT's in search of good homes.* Congr. Num., **102,** (1994) 97–110. ⇒1, 2, 12, 41

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

[63] Siklósi, B.: *Soros és párhuzamos algoritmusok összehasonlítása sportversenyekkel kapcsolatos problémákban.* Programtervező matematikus diplomamunka. ELTE TTK, Budapest, 2001. 69 oldal. ⇒28

[64] Simion, R.: *Convex polytopes and enumeration.* Advances in Applied Math. **18(2)** (1996) 149–180. ⇒12

[65] Sloane N. J. A., Plouffe S.: *The Encyclopedia of Integer Sequences.* Academic Press, 1995. ⇒12, 13, 14

[66] Sloane N. J. A. (szerk.): *Encyclopedia of Integer Sequences.* 2011. ⇒2, 9

[67] Sloane N. J. A.: *The number of ways to put $n+1$ indistinguishable balls into $n+1$ distinguishable boxes.* In (ed. N. J. A. Sloane): The On-line Encyclopedia of the Integer Sequences. 2011. ⇒1, 12, 27, 41

[68] Sloane N. J. A.: The number of degree-vectors for simple graphs. In (ed. N. J. A. Sloane): The On-line Encyclopedia of the Integer Sequences. 2011. ⇒12, 27

[69] Sloane N. J. A.: *The number of bracelets with n red, 1 pink and $n-1$ blue beads.* In (ed. N. J. A. Sloane): The On-line Encyclopedia of the Integer Sequences. 2011. ⇒12, 27

[70] Soroker, D.: *Optimal parallel construction of prescribed tournaments.* Discrete Appl. Math. **29(1),** (1990) 113–125. ⇒28

[71] Stanley, R.: *Enumerative Combinatorics. Vol. 2.* Cambridge University Press, Cambridge, 1997. ⇒12

[72] Stanley, R.: *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. ⇒12

[73] Tripathi, A., Tyagy, H.: *A simple criterion on degree sequences of graphs.* Discrete Appl. Math. **156(18),** (2008) 3513–3517. ⇒2

[74] Tripathi, A., Vijay, S.: *A note on a theorem of Erdős & Gallai.* Discrete Math. **265(1–3),** (2003) 417–420. ⇒5, 6

[75] Tripathi, A., Venugopalan, S., West, D. B.: *A short constructive proof of the Erdős-Gallai characterization of graphic lists.* Discrete Math. **310(4),** (2010) 833–834. ⇒ 1, 2, 4, 41

[76] Weisstein, E. W.: *Degree sequence.* From MathWorld—Wolfram Web Resource, 2011. ⇒2

[77] Weisstein, E. W.: *Graphic sequence.* From MathWorld—Wolfram Web Resource, 2011. ⇒2

[78] WINSTON, K. J., KLEITMAN, D. J.: *On the asymptotic number of tournament score sequences.* J. Combin. Theory Ser. A. **35,** (1983) 208–230. ⇒12

Beérkezett:

IVÁNYI ANTAL
tony@compalg.inf.elte.hu

LUCZ LORÁND
lorand.lucz@gmail.com

SÓTÉR PÉTER
mapoleon@freemail.hu

Eötvös Loránd Tudományegyetem
Informatikai Kar
1117 Budapest, Pázmány Péter sétány 1/C

Budapest, 2011. szeptember 7.

LINEAR ERDŐS-GALLAI TEST

ANTAL IVÁNYI, LORÁND LUCZ, PÉTER SÓTÉR

Havel in 1955 [22], Erdős and Gallai in 1960 [15], Hakimi in 1962 [20], Ruskey, Cohen, Eades and Scott 1994-ben [61], Barnes and Savage in 1997 [4], Tripathi, Venugopalan and West in 2010 [75] 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 ERDŐS-GALLAI-LINEAR, whose worst running time is linear. 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 ERDŐS-GALLAI-LINEAR is asymptotically optimal. Using this algorithm we determined the number of the degree sequences of the simple graphs for $n = 24,\ 25,\ 26,\ 27$ and $n = 28$ (*Online Encyclopedia of Integer Sequences* [67] contains these numbers only for $n = 1,\ \ldots,\ 23$).