MULTIGRÁFOK FOKSOROZATAI

IVÁNYI ANTAL és LUCZ LORÁND (2012. február 7.)

Havel 1955-ben [23], Erdős és Gallai 1960-ban [17], Hakimi 1962-ben [22], Tripathi, Venugopalan és West 2010-ben [72], Özkan [51] 2011-ben javasoltak módszert annak eldöntésére, hogy nemnegatív egészek sorozata lehet-e egy egyszerű gráf foksorozata. 1974-ben Chungphaisan [15] kiterjesztette a csúcspárok között legfeljebb $b \geq 1$ élet tartalmazó multigráfokra mind a Havel-Hakimi, mind pedig at Erdős-Gallai tételt. Ezeknek az ismert algoritmusoknak a legrosszabb futási ideje legalább négyzetes. Cikkünkben bemutatjuk a Chungphaisan-Erdős-Gallai algoritmus lineáris és annak gyorsított változatát. A Chungphaisan-Havel-Hakimi algoritmust pedig úgy javítjuk és gyorsítjuk, hogy $b = 1$ mellett $b = 2$ esetén is lineáris futási idejű.

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 [41], Hakimi kémiai [22], Kim et al. [34] és Newman és Barabási [50] hálózati, Bozóki, Fülöp, Kéri, Poesz és Rónyai gazdasági [8, 9, 33], Liljeros et al. emberi kapcsolatokra vonatkozó [42], Iványi et al. pedig sportbeli [25, 26, 30, 31, 32, 54, 56] alkalmazásokra hivatkoztak.

Legyenek a, b és n egészek, $n \geq 1$ és $b \geq a \geq 0$. Az (a, b, n)-*gráfok* olyan hurokmentes – irányított vagy irányítatlan – gráfok, melyek csúcshalmaza $V = \{v_1, \ldots, v_n\}$ és a különböző v_i és v_j csúcsok legalább a és legfeljebb b éllel vannak összekötve. Eszerint az *egyszerű irányítatlan gráfok* $(0, 1, n)$-gráfok, míg a *tournamentek* $(1, 1, n)$-gráfok.

Irányított gráfok esetén ha v_i és v_j összehasonlításakor v_i kap egy pontot, akkor annak a gráfban v_i-ből v_j-be menő irányított él felel meg. Irányítatlan gráfok esetén viszont csúcspárok kapják a pontot és annak a két csúcsot összekötő irányítatlan él felel meg.

Ebben a cikkben elsősorban azt vizsgáljuk, hogy nemnegatív egész számok $s = (s_1, \ldots, s_n)$ nemnövekvő sorozata és adott a alsó korlát, valamint b felső korlát esetén létezik-e olyan irányítatlan (a, b, n)-gráf, amelynek foksorozata s. Ennek megfelelően – ha mást nem mondunk – a gráf kifejezés irányítatlan gráfot jelent.

Emellett foglalkozunk a foksorozatok számával, amelyet $G(a, b, n)$-nel jelölünk.

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, 16, 24, 32, 45, 51, 57, 70, 72, 73, 74], illetve [5, 9, 12, 18, 20, 25, 26, 31, 34, 37, 40, 47, 53, 54, 55]).

Legyenek l, m és u egész számok, továbbá $1 \leq m$ és $l \leq u$. Egész számok $s = (s_1, \ldots, s_m)$ sorozatát (l, u, m)-*korlátosnak* (röviden: korlátosnak) nevezzük, ha $l \leq s_i \leq u$ minden $1 \leq i \leq m$ indexre. Az $s = (s_1, \ldots, s_m)$ (l, u, m)-korlátos sorozatot (l, u, m)-*szabályosnak* mondjuk, ha $u \geq s_1 \geq \cdots \geq s_m \geq l$.

A vizsgálatok során kitüntetett szerepet játszanak az $(a(n-1), b(n-1), n)$-szabályos sorozatok. Ezeket a sorozatokat (a, b, n)-*grafikusnak* (vagy röviden grafikusnak) nevezzük, ha létezik olyan (a, b, n)-gráf, melynek foksorozata s.

Jelentős számú cikk (például [11, 19, 38, 46]) foglalkozik páros számok *grafikus felbontásaival*: előállítják a $2k$ páros szám pozitív egész összeadandókra 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 $(0, 2k-1, 2k)$-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 grafikus sorozatok ellenőrzésétől, ezért ezek a megszorítások nem jelentik az általánosság korlátozását.

A cikkben csak *teljes* gráfokkal foglalkozunk. Ezekre az jellemző, hogy ha $a \leq c \leq b$, akkor bármely két csúcs között c él is meg van engedve, és az irányított esetben azok tetszőlegesen irányíthatók (azaz eltérünk a teljes gráfok szokásos definíciójától). A *hiányos* gráfoknál bizonyos lehetőségek tiltva vannak. Például a labdarúgásnak [20, 27, 30, 39] olyan irányított $(2, 3, n)$-gráfok felelnek meg, amelyekben a csúcsokat 2 vagy 3 él köti össze, azonban 2 él esetén azok mindig ellentétesen, míg 3 él esetén azok mindig azonosan vannak irányítva.

Míg teljes gráfok esetén a sorozatok tesztelése az operációkutatás folyamos módszereivel kényelmesen megoldható (bár gyakran vannak gyorsabb algoritmusok is), hiányos gráfok esetén ezek a módszerek nem alkalmazhatók.

Cikkünk fő célkitűzése, hogy minél kisebb várható futási idejű algoritmusokat találjunk annak eldöntésére, hogy előírt s szabályos sorozat grafikus-e. Eközben a minden sorozatot helyesen minősítő *pontos*, és a csak a szabályos sorozatok egy részét minősítő *közelítő* algoritmusokkal is foglalkozunk.

Érdemes megemlíteni, hogy a fokszámsorozatok számának meghatározásával kapcsolatos nehézségek miatt annak is jelentős irodalma (lásd például [7, 16, 32, 47]) van, hogy véletlen mintavétellel becsüljük ezeket a számokat.

Melléktermékként bővítettük az On-Line Encyclopedia of Integer Sequences [63] adatbázist. Módszerünk az összes grafikus sorozat gazdaságos előállítására is alkalmas (lásd Ruskey [58] 1994-es, valamint Barnes és Savage [4] 1997-es cikkét).

A cikk felépítése a következő. A bevezető első rész után a $(0, 1, n)$ témakör klasszikus algoritmusait foglaljuk össze. A harmadik részben új pontos algoritmusokat, a negyedikben általános leszámlálási eredményeket, az ötödikben pedig új tesztelő algoritmusokat ismertetünk. A hatodik részben a közelítő algoritmusok hatékonyságát és futási idejét, míg a hetedikben a pontos algoritmusok futási idejét elemezzük. A nyolcadik rész témája a $(0, b, n)$-gráfok potenciális foksorozatainak tesztelése, míg a kilencedikben az (a, b, n)-gráfoké a főszerep. A tizedik részben a $(0, 1, n)$-grafikus sorozatok párhuzamos leszámlálása a téma.

2. Klasszikus pontos algoritmusok $(0, 1, n)$-gráfokhoz

Ebben a részben két, a $(0, 1, n)$-gráfok potenciális foksorozatainak tesztelésére alkalmas klasszikus algoritmust ismertetünk.

2.1. Havel-Hakimi algoritmus (HH)

A feladat megoldására az első módszert Vaclav Havel cseh matematikus javasolta 1955-ben [23, 43]. 1962-ben Louis Hakimi [22] Haveltől függetlenül publikálta ugyanezt az eredményt, ezért ma a tételt rendszerint *Havel-Hakimi tételnek*, a módszert pedig *Havel-Hakimi algoritmusnak* nevezik.

1. Tétel. (Hakimi [22], Havel [23]) *Ha $n \geq 3$, az $(s_1, \ldots, s_n)$ $(0, 1, n)$-szabályos sorozat akkor és csak akkor $(0, 1, n)$-grafikus, ha az $(s_2 - 1, s_3 - 1, \ldots, s_{s_1} - 1, s_{s_1+1} - 1, s_{s_1+2}, \ldots, s_n)$ sorozat $(0, 1, n - 1)$-grafikus.*

Bizonyítás. Lásd [22, 23]. □

A továbbiakban sorozatok ismétlődő elemeinek tömör jelölésére használjuk az $s = (c^d)$ típusú jelölést, ami azt jelzi, hogy a sorozat d darab c-t tartalmaz.

Ha ezen tétel alapján írunk egy rekurzív algoritmust, akkor annak futási ideje legjobb esetben – például az $(n - 1, 0^{n-1})$ bemenetre – $\Theta(1)$, legrosszabb esetben pedig – például a *homogén* $((n-1)^n)$ bemenetre – $\Theta(n^2)$. Ez ugyanis grafikus sorozat, ezért minden elemét ellenőrizni kell. Másrészt az elemek összege négyzetes, és az algoritmus az elemeket egyesével csökkenti nullára. 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). Az alábbi algoritmus csak a bemenet tesztelését végzi el.

A következő Havel-Hakimi algoritmus az 1. tételen alapul. A cikk programjaiban a [13] tankönyvben leírt pszeudokód konvenciókat követjük.

Bemenet. n: csúcsok száma ($n \geq 1$);
$s = (s_1, \ldots, s_n)$: $(0, 1, n)$-szabályos sorozat.

Kimenet. L: s grafikusságát jelző logikai változó (ha s $(0, 1, n)$-grafikus, akkor $L = 1$, egyébként $L = 0$).

Munkaváltozók. i, j: ciklus változók.

Havel-Hakimi(n, s, L)

```
01 for i = 1 to n − 1                        // 01–06. sor: s elemeinek tesztelése
02     if s_{s_i+i} == 0                     // 02–03. sor: s nem grafikus
03         return 0
04     for j = i + 1 to s_i + i
05         s_j = s_j − 1
06     (s_{i+1}, ..., s_n) rendezése nemnövekvő sorrendbe
07 return 1                                  // 07. sor: s grafikus
```

Az algoritmust később irányított gráfokra [18, 25, 26, 35] is kiterjesztették.

2.2. Erdős-Gallai algoritmus (EG)

Időrendben a következő eredmény Erdős Pál és Gallai Tibor alábbi szükséges és elégséges feltétele [17] volt.

Nemnegatív egészek adott $s = (s_1, \ldots, s_n)$ sorozata esetén a sorozat első i elemét a sorozat s_i eleméhez tartozó *fejnek,* míg a többi elemét az s_i elemhez tartozó *faroknak* nevezzük. A fejelemek összegét H_i, míg a farokelemek összegét T_i jelöli ($i = 1, \ldots, n$). A $\sum_{k=i+1}^{n} \min(i, s_k)$ összeget pedig C_i-vel jelöljük és a farok *becsült kapacitásának* nevezzük. Ha egy s sorozatra H_n páros, akkor a sorozatot *n-párosnak,* egyébként *n-páratlannak* nevezzük.

2. Tétel. (Erdős, Gallai, [17]) *Ha $n \geq 1$, a $(0, 1, n)$-szabályos $(s_1, \ldots, s_n)$ sorozat akkor és csak akkor $(0, 1, n)$-grafikus, ha*

$$H_n \quad \textit{páros} \tag{1}$$

és

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

Bizonyítás. Lásd [14, 17, 59, 72]. □

A tétel alapgondolata az, hogy az első i csúcs fokait egyrészt ezen csúcsok közötti élekkel – ezekből legfeljebb $i(i-1)$ van – másrészt a nagyobb indexű csúcsok fokaival lehet lekötni. A nagyobb indexű csúcsokra pedig az jellemző, hogy egyrészt legfeljebb i csúcs egy-egy fokát tudják lekötni,másrészt legfeljebb annyi fokot, mint a saját fokszámuk. A tétel szépségét az adja, hogy ezeknek a természetes szükséges feltételeknek az elégségességét is tartalmazza.

A 2. tételen alapul a következő Erdős-Gallai algoritmus.

Bemenet. n: az s sorozat hossza;
$s = (s_1, \ldots, s_n) : (0, 1, n)$-szabályos sorozat.

Kimenet. L : s grafikusságát jelző logikai változó.

Munkaváltozók. C: aktuális C_i;
i, k: ciklusváltozók.

Erdős-Gallai(n, s, L)

```
01 H_1 = s_1                          // 01–03. sor: H elemeinek kiszámítása
02 for i = 2 to n
03     H_i = H_{i-1} + s_i
04 if H_n páratlan                    // 04–05. sor: paritás ellenőrzése
05   return 0
06 for i = 1 to n - 1                 // 06–11. sor: s tesztelése
07     C = 0                          // 07. sor: t kezdeti értékének beállítása
08     for k = i + 1 to n             // 08–09. sor: C frissítése
09         C = C + min(i, s_k)
10     if H_i - i(i - 1) > C          // 10. sor: szükséges feltétel ellenőrzése
11         return 0                   // 11. sor: s nemgrafikus
12 return 1                           // 12. sor: s grafikus
```

EG memóriaigénye $\Theta(n)$. Bár ez a program csak ellenőriz, futási ideje a legjobb $\Theta(n)$ és a legrosszabb $\Theta(n^2)$ között változik. A közelmúltban Tripathi et al. [72] publikáltak a tételre konstruktív bizonyítást, amely grafikus bemenet esetén $\Theta(n^3)$ idő alatt egy megoldást is előállít.

A szabályos sorozatoknak aszimptotikusan a fele páros sorozat. Az 1. táblázathoz a $(0,1,n)$-szabályos sorozatok számát a majd a 4 szakaszban szereplő (34) képlet alapján, míg a $(0,1,n)$-páros sorozatok számát az ugyancsak a 4 szakaszban következő 14. lemma alapján számítottuk. A táblázat harmadik oszlopa a két számosság hányadosának gyors konvergenciáját szemlélteti $n = 1,\ldots,38$ csúcs esetén.

3. Új pontos algoritmusok $(0,1,n)$-gráfokhoz

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 grafikus-e. Ezt a megfigyelést hasznosítja a következő állítás, amelyben p az s sorozat pozitív elemeinek a számát jelöli.

3. Következmény. *Ha $n \geq 1$, az $(s_1,\ldots,s_n)$ $(0,1,n)$-szabályos sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha $s_1 = 0$ vagy az $(s_1,\ldots,s_p)$ sorozat $(0,1,p)$-grafikus.*

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 Erdős-Gallai (EGn), illetve nullamentes Havel-Hakimi (HHn) algoritmusnak nevezzük.

n	$R(n)$	$E(n)$	$E(n)/R(n)$
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

1. táblázat. A szabályos ($R(n)$) és a páros ($E(n)$) sorozatok száma, valamint ezen számok hányadosa ($E(n)/R(n)$)).

3.2. Rövidített Erdős-Gallai algoritmus (EGr)

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 a következő lemma. Tripathi és Vijay 2003-as cikkében [71] 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 [71]) *Ha* $n \geq 1$, *a* $(0,1,n)$*-szabályos* $s = (s_1, \ldots, s_n)$ *sorozat akkor és csak akkor* $(0,1,n)$*-grafikus, ha*

$$H_n \quad \textit{páros} \tag{3}$$

és

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

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 az $s = (5^{100})$ sorozat esetén (2) jobb oldalát az Erdős-Gallai algoritmus szerint kilencvenkilencszer, míg a rövidített Erdős-Gallai algoritmus szerint csak hatszor 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.

3.3. Ugró Erdős-Gallai algoritmus (EGu)

Az ismétlődő elemeket összevonva egy szabályos $(s_1, \ldots, s_n)$ sorozat $(s_{i_1}^{e_1}, \ldots, s_{i_q}^{e_q})$ alakban is felírható, ahol $s_{i_1} > \cdots > s_{i_q}$, $e_1, \ldots, e_q \geq 1$ és $e_1 + \cdots + e_q = n$. Legyen $g_j = e_1 + \cdots + e_j$ $(j = 1, \ \ldots, q)$.

Az s_i elemet az s sorozat *ugró* elemének nevezzük, ha $i = n$ vagy $1 \leq i \leq n-1$ és $s_i > s_{i+1}$. Ekkor az ugró elemek az $s_{g_1}, \ldots, s_{g_q}$ elemek. Az ugró (vagy ellenőrző) elemeket $c_1 = s_{g_1}, \ldots, c_q = s_{g_q}$ módon jelöljük.

Tripathi és Vijai 2003-ban a [71] cikkben az Erdős-Gallai tétel következő, lényeges gyorsítást lehetővé tevő változatát is bizonyították.

5. TÉTEL. (Tripathi, Vijay [71]) *A* $(0,1,n)$*-szabályos* $s = (s_1, , \ldots, , s_n)$ *sorozat akkor és csak akkor* $(0,1,n)$*-grafikus, ha*

$$H_n \quad \textit{páros} \tag{6}$$

és

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

Bizonyítás. Lásd [71]. □

A következő program az Erdős-Gallai algoritmusnak az 3. és 4. lemma, valamint a 8. tétel alapján gyorsított változatát mutatja be.

Bemenet. n: a csúcsok száma ($n \geq 1$);
$s = (s_1, \ldots, s_n)$: $(0, 1, n)$-szabályos sorozat.

Kimenet. L: s grafikusságát jelző logikai változó.

Munkaváltozó. i, j: ciklusváltozók;
$H = (H_1, \ldots, H_n)$: H_i s első i elemének az összege;
p: s pozitív elemeinek a száma;
s_{p+1}: segédváltozó annak eldöntéséhez, hogy s_p ugró elem-e.

```
ERDŐS-GALLAI-UGRÓ(n, s, L)
01 p = n                                   // 01–03. sor: nullamentesítés
02 while s_p = 0
03         p = p − 1
04 H_1 = s_1                               // 04–08. sor: paritás ellenőrzése
05 for i = 2 to p
06       H_i = H_{i−1} + s_i
07 if H_p páratlan
08    return 0
09 s_{p+1} = 0                             // 09–19. sor: fej igényének ellenőrzése
10 i = 1
11 while i ≤ p és i(i − 1) < H_i
12         while s_i == s_{i+1}
13                 i = i + 1
14         E = 0
15         for j = i + 1 to p
16              E = E + min(j, s_j)
17         if H_i > i(i − 1) + E
18            return 0
19         i = i + 1
21 return 1                                // 20. sor: s grafikus
```

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

Megjegyezzük, hogy az ellenőrzést elég a $(q-1)$-edik ugrópontig folytatni.

A 2. táblázat azt mutatja, hogy $n = 3, \ldots, 15$ csúcs esetén EGu hány menet alatt tudja kizárni a nem $(0, 1, n)$-grafikus sorozatokat a $(0, 1, n)$-szabályos sorozatok tesztelése során. $f_i(n) = f_i$ azoknak az n hosszúságú nem $(0, 1, n)$-grafikus sorozatoknak a száma, amelyek pontosan i tesztelési menetet igényeltek. A táblázat minden sorára jellemző, hogy a maximális menetszám körülbelül $\frac{n}{2}$.

A 3. táblázat tartalmazza a $(0, 1, n)$-szabályos, -grafikus és -nemgrafikus sorozatok számát, valamint az EGu algoritmus számára a nemgrafikus, grafikus és összes

n/i	$R(n) - G(n)$	f_1	f_2	f_3	f_4	f_5	f_6	f_7
3	6	6						
4	24	24						
5	95	91	4					
6	360	338	22					
7	1 374	1 262	102	10				
8	5 222	4 729	409	84				
9	19 949	17 841	1 587	487	34			
10	76 362	67 645	6 025	2 294	398			
11	293 368	257 779	22 802	9 820	2 825	142		
12	1 129 961	986 274	86 292	39 745	15 554	2 096		
13	4 363 985	3 787 213	327 644	156 295	74 542	17 632	659	
14	16 891 448	14 586 597	1 248 368	605 592	327 404	111 872	11 615	
15	65 516 140	56 330 831	4 774 119	2 331 442	1 363 561	599 615	113 316	3 256

2. táblázat. A $(0, 1, n)$-szabályos nem $(0, 1, n)$-grafikus sorozatok eloszlása $n = 3, \ldots, 15$ csúcsra aszerint, hogy az EGu algoritmus hány menet alatt tudja őket kizárni.

n	$R(n)$	$G(n)$	X'	Y'	Z'
3	10	4	0,3333333333	0,5833333333	0,4333333333
4	35	11	0,2500000000	0,5909090909	0,3571428571
5	126	31	0,2084210526	0,6064516129	0,3063492063
6	462	102	0,1768518519	0,6192810458	0,2745310245
7	1 716	342	0,1555416927	0,6219715957	0,2485014985
8	6 435	1 213	0,1388117579	0,6267518549	0,2307886558
9	24 310	4 361	0,1259433778	0,6312007949	0,2165821107
10	92 378	16 016	0,1154618789	0,6336476024	0,2053021282
11	352 716	59 348	0,1068633005	0,6357110908	0,1958472384
12	1 352 078	222 117	0,0996191461	0,6373495350	0,1879565503
13	5 200 300	836 315	0,0934514246	0,6386612700	0,1811323607
14	20 058 300	3 166 852	0,0881205642	0,6397881871	0,1752191576
15	77 558 760	120 426 20	0,0834688999	0,6407780422	0,1700028030

3. táblázat. A $(0, 1, n)$-szabályos és -grafikus sorozatok száma, valamint az Erdős-Gallai-ugró algoritmus által az $n = 3, \ldots, 15$ hosszú sorozatok vizsgálata során végzett tesztek átlagos száma.

sorozat kiszűréséhez szükséges menetek átlagos számát $n = 3, \ldots, 15$ csúcs esetén. A táblázatban szereplő X', Y' és Z' hatékonysági jellemzők definícióját a (29), (30) and (31) képletek tartalmazzák. Figyelemre méltó, hogy az utolsó 3 oszlopban az együtthatók monoton csökkennek.

3.4. Lineáris Erdős-Gallai algoritmus (EGl)

A következő Erdős-Gallai-Lineáris algoritmus kihasználja, hogy az s 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 súlypontokat tartalmazó $w(s)$ sorozat.

Adott s sorozat esetén legyen $w(s) = (w_0, \ldots, w_{n-1})$, ahol $i > s_1$ esetén $w_i = 0$, egyébként pedig w_i az s sorozat legnagyobb indexű olyan elemének indexe, amelyik legalább akkora, mint i.

Az s sorozat s_i elemének ellenőrzésekor két eset van: ha $i > w_i$, akkor a C_i kapacitás egyszerűen számítható: $H_n - H_i$, mivel a farok minden s_j elemének hozzájárulása csak s_j.

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

6. Tétel. (Iványi, Lucz, Móri, Sótér [30]) *Ha* $n \geq 1$, *az* $s = (s_1, \ldots, s_n)$ $(0,1,n)$-*szabályos sorozat akkor és csak akkor* $(0,1,n)$-*grafikus, ha*

$$H_n \quad \textit{páros,} \tag{8}$$

továbbá

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

ahol

$$k(s) = k = \begin{cases} w_i, & \textit{ha } i \leq w_i, \\ i, & \textit{ha } i > w_i. \end{cases} \tag{10}$$

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

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

Ha $i \leq w_i$, akkor

$$H_i \leq i(i-1) + (w_i - i + 1)i + H_n - H_{w_i} \tag{11}$$

és ha $i > w_i$, akkor

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

Ha (11) jobb oldalán kiemeljük i-t, akkor a

$$H_i \leq iw_i + H_n - H_{w_i} \tag{13}$$

egyenlőtlenséget kapjuk. Ha a (9) egyenlőtlenségbe (10) alapján behelyettesítjük k-t, akkor az $i \leq w_i$ esetben a (11), az $i > w_i$ esetben pedig a (12) egyenlőtlenséget kapjuk. □

A következő program a 6. tétel alapján adott n-re tetszőleges n-szabályos sorozatról eldönti, hogy grafikus-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 ($n \geq 1$);
$s = (s_1, \ldots, s_n)$: $(0, 1, n)$-szabályos sorozat.

Kimenet. L: s grafikusságát jelző logikai változó.

Munkaváltozók. i: ciklusváltozó;
$H = (H_1, \ldots, H_n)$: H_i az éppen tesztelt s első i elemének az összege;
$w = (w_1, \ldots, w_{n-1})$: w_i az s_i-hez tartozó súlypont;
y: ellenőrzés egyszerűsítéséhez használt változó (az aktuális s_i vágópontja, azaz w és i maximuma).

Erdős-Gallai-lineáris(n, s, L)
01 $H_1 = s_1$ // 01. sor: H_1 beállítása
02 **for** $i = 2$ **to** n // 02–03. sor: H további elemeinek számítása
03 $\quad H_i = H_{i-1} + s_i$
04 **if** H_n páratlan // 04–06. sor: paritás ellenőrzése
05 $\quad L = 0$
06 $\quad$ **return**
07 $w = n$ // 07. sor: súlypont beállítása
08 **for** $i = 1$ **to** $n - 1$ // 08–16. sor: s elemeinek tesztelése
09 $\quad$ **while** $w > 1$ and $s_w < i$ // 08–10. sor: aktuális súlypont számítása
10 $\qquad w = w - 1$
11 $\quad y = \max(i, w)$ // 11. sor: aktuális vágópont számítása
12 $\quad$ **if** $H_i > i(y-1) + H_n - H_y$
13 $\qquad L = 0$ // 13–14. sor: nemgrafikus s elutasítása
14 $\qquad$ **return** L
15 $L = 1$ // 15–16. sor: s grafikus
16 **return** L

7. Következmény. *A $(0, 1, n)$-szabályos $s = (s_1, \ldots, s_n)$ sorozatról az* EGl *algoritmus $\Theta(n)$ idő alatt dönti el, hogy $(0, 1, n)$-grafikus-e.*

Bizonyítás. A 01–03. sorok $\Theta(n)$ időt igényelnek. Mivel a w súlypontot legfeljebb n-szer frissítjük, ezért a 04–16. sorok időigénye $O(n)$, így az algoritmus futási ideje $\Theta(n)$. □

3.5. Gyors Erdős-Gallai algoritmus (EGgy)

Tripathi és Vijai a [71] cikkben az Erdős-Gallai tétel következő, lényeges gyorsítást lehetővé tevő változatát is bizonyították.

Az ismétlődő elemeket gyakoriságuk segítségével tömörítve a $(0, 1, n)$-szabályos $(s_1, \ldots, s_n)$ sorozat felírható az $(s_{i_1}^{e_1}, \ldots, s_{i_q}^{e_q})$ alakban, ahol $s_{i_1} < \cdots < s_{i_q}$, $e_1, \ldots, e_q \geq 1$ és $e_1 + \cdots + e_q = n$. Legyen $g_j = e_1 + \cdots + e_j$ $(j = 1, \ldots, q)$.

Az s_i elemet az s *ugró pontjának* nevezzük, ha $i = n$ vagy $1 \le i \le n-1$ és $s_i > s_{i+1}$. Ekkor az ugró pontok az $s_{g_1}, \ldots, s_{g_q}$ elemek.

8. Tétel. (Tripathi, Vijay [71]) *Az $s = (s_1, \ldots, s_n)$ szabályos sorozat akkor és csak akkor grafikus, ha*

$$H_n \quad \textit{páros} \tag{14}$$

és

$$H_{g_i} - g_i(g_i - 1) \le \sum_{k=c_i+1}^{n} \min(g_i, s_k) \quad (i = 1, \ldots, q). \tag{15}$$

Bizonyítás. Lásd [71]. □

Megjegyezzük, hogy az ellenőrzést elég a $(q-1)$-edik ugró pontig folytatni.

A következő tétel – EGe és EGu előnyeit egyesítve – a tesztelési idő további csökkentését teszi lehetővé.

9. Tétel. *A $(0,1,n)$-szabályos $s = (s_1, \ldots, s_n)$ sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha igaz az, hogy*

$$H_n \quad \textit{páros} \tag{16}$$

és

$$H_{g_i} \le \begin{cases} H_n - H_{g_i} + g_i(g_i - 1), & \textit{ha} \;\; w_i \le g_i \\ H_n - H_{w_i} + g_i(w_i - 1), & \textit{ha} \;\; w_i > g_i \quad (i = 1, \ldots, q-1). \end{cases} \tag{17}$$

Bizonyítás. A csak az ugró pontokban való tesztelés elégségességét Tripathi és Vijay [71] már bebizonyították. A tételben megadott feltétel ezeket az ellenőrzéseket végzi el, kihasználva a sorozat elemeinek monoton csökkenését, azaz a

$$\sum_{k=g_i+1}^{n} \min(g_i, s_k) \tag{18}$$

összeget nem számolja újra minden esetben, pontosabban nem ebben a formában végzi el a számítást, hanem explicit módon.

A kifejezés értéke a (19) formában adható meg, mégpedig azért, mert a sorozat monotonitása garantálja, hogy a $k \le w_i$ esetén a $\min(i, s_k)$ kifejezés értéke i, míg $k > w_i$ esetén s_k. Ebből következik, hogy

$$\sum_{k=g_i+1}^{n-1} \min(g_i, s_k) = \begin{cases} H_n - H_{g_i}, & \text{ha} \;\; w_i \le g_i \\ H_n - H_{w_i} + g_i(w_i - g_i), & \text{ha} \;\; w_i > g_i. \end{cases} \tag{19}$$

Az eddigiek alapján az eredeti feltételt átírhatjuk a következő alakba:

$$H_{g_i} - g_i(g_i - 1) \le \begin{cases} H_n - H_{g_i}, & \text{ha} \;\; w_i \le g_i \\ H_n - H_{w_i} + g_i(w_i - g_i), & \text{ha} \;\; w_i > g_i. \end{cases} \tag{20}$$

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15
EGu	4	12	16	21	26	32	37	43	49	56	63	70	77	85
$\frac{\text{EGu}}{n}$	2.0	4.0	4.0	4.2	4.3	4.6	4.6	4.8	4.9	5.1	5.3	5.4	5.5	5.7
EGgy	12	15	17	19	21	23	25	27	29	31	33	35	37	39
$\frac{\text{EGgy}}{n}$	6.0	5.0	4.3	3.8	3.5	3.3	3.1	3.0	2.9	2.8	2.8	2.7	2.6	2.6

4. táblázat. Az ugró és a gyors Erdős-Gallai algoritmusok egy sorozatra jutó átlagos műveletigénye.

A (20) egyenlőtlenséget átrendezve megkapjuk a (17) egyenlőtlenséget. □

A most megadott tétel alapján megvalósított EGgy algoritmus és az eddigi legjobb (ugró Erdős-Gallai) algoritmus sorozatonkénti átlagos műveletszámait, valamint a sorozat egyetlen elemére jutó átlagos műveletszámot tartalmazza a 4. táblázat.

A táblázatból leolvasható, hogy az átlagos műveletszám a lineáris algoritmus esetében kevesebb, mint fele annyi, mint az ugró algoritmus esetébn és az n érték növelésével minden lépésben ugyanannyival növekszik. Az utóbbi azért fontos, mert így az n növelésével lépésről lépésre nagyobb az új algoritmussal elért gyorsulás a korábbiakhoz képest. Az utóbbi kijelentés azonban nem meglepő, ha figyelembe vesszük, hogy a korábbi ismert algoritmusok négyzetesek, míg az új algoritmus lineáris futási idejű. Jól látható, hogy a régi módszer esetén a sorozatok egy eleméhez tartozó átlagos műveletszám az n érték növekedésével együtt nőtt, az új módszernél azonban ez a szám lépésről lépésre csökken.

A 9. tétel feltételeit ellenőrzi a következő algoritmus.

Bemenet. n: az s sorozat hossza;
$s = (s_1, \ldots, s_n)$: az ellenőrizendő sorozat.

Kimenet. L: s grafikusságát jelző logikai változó.

Erdős-Gallai-gyors(n, s, L)

```
01 H_1 = s_1                                   // 01. sor: H_1 beállítása
02 for i = 2 to n                              // 02–04. sor: H további értékeinek számítása
03     H_i = H_{i−1} + s_i
04 if H_n páratlan                             // 04–07. sor: paritás ellenőrzése
05     L = 0                                   // 05–06. sor: nemgrafikus sorozat elutasítása
06     return
07 w = n                                       // 07. sor: súlypont kezdeti értéke
08 for i = 1 to n − 1                          // 08–26. sor: sorozat tesztelése
09     if s_i == s_{i+1}                       // 09–11 sor: ugrópont tulajdonság ellenőrzése
10         continue                            // 10. sor: nem ugrópont átlépése
11     while w > 1 és s_w ≤ i                  // 11–12. sor: súlypont frissítése
12             w = w − 1
13     if w < i                                // 13–16. sor: súlypont az ugrópont előtt van
```

14 **if** $H_i > H_n - H_i + i(i-1)$ 14–18. sor: tétel feltételének ellenőrzése
15 $L = 0$ // 15–16. sor: nemgrafikus sorozat elutasítása
16 **return**
17 **else if** $H_i > H_n - H_w + i(w-1)$ // 17–19. sor: súlypont az ugrópont után van
18 $L = 0$ // 18–19. sor: nemgrafikus sorozat elutasítása
19 **return**
20 $L = 1$ // 20–21. sor: grafikus sorozat elfogadása
21 **return L**

10. TÉTEL. *A lineáris Erdős-Gallai ugró algoritmus műveletigénye lineáris.*

Bizonyítás. A 01. sor időigénye $O(1)$, a 02–03. soré $\Theta(n)$, a 04–07. soré $O(1)$, a 08–20. soré $O(n)$, a 21–22. soré pedig $O(1)$. Így az algoritmus teljes műveletigénye $\Theta(n)$. $\square$

3.6. Eltoló Havel-Hakimi algoritmus (HHe)

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.

3.7. Paritásos Havel-Hakimi algoritmus (HHp)

Érdekes gondolat az Erdős-Gallai és a Havel-Hakimi feltételek együttes alkalmazása úgy, hogy először s 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 jelentősen csökken.

3.8. Lineáris Havel-Hakimi tesztelő algoritmus (HHl)

Az EGl algoritmusban kulcsszerepe volt az s_i elemhez tartozó w_i súlypontnak [30], amely $i > s_1$ esetén 0, egyébként a legnagyobb olyan k index, amelyre igaz, hogy $s_k \geq bi$ (ez az egyenlőtlenség a $(0,1,n)$-gráfok esetén az $s_k \geq i$ egyenlőtlenségre egyszerűsödik.)

Ez a súlypont arra is alkalmas, hogy a Havel-Hakimi algoritmus lineáris változatában fontos szereplő legyen. Az algoritmus alapja a következő tétel.

11. Tétel. *Ha $n \geq 1$, az $(s_1, \ldots, s_n)$ $(0,1,n)$-szabályos sorozat akkor és csak akkor $(0,1,n)$-grafikus, ha*

$$s_1 < w_1 \tag{21}$$

és

$$s_i \leq w_i + r_{i-1} \quad (i = 2, \ldots, n-1), \tag{22}$$

ahol

$$w_i = \max(k \geq 0 \mid s_k \geq i) \quad (i = 1, \ldots, n), \tag{23}$$

és

$$r_i = w_i + r_{i-1} - s_i \quad (i = 1, \ldots, n). \tag{24}$$

Bizonyítás. Ha ???? □

Bemenet. n: csúcsok száma $(n \geq 1)$;
$s = (s_1, \ldots, s_n)$: a tesztelendő $(0,1,n)$-szabályos sorozat.

Kimenet. L: s grafikusságát jelző logikai változó.

Munkaváltozó. i: ciklusváltozó;
$r = (r_1, \ldots, r_n)$: r_i az s_i-hez tartozó maradék;
$w = (w_1, \ldots, w_n)$: w_i az i indexhez tartozó súlypont;
$H = (H_1, \ldots, H_n)$: H_i az s sorozat első i elemének összege.

Havel-Hakimi-Lineáris(n, s, L)

```
01 if s_1 == 0                       // 01–03. sor: nullákból álló sorozat elfogadása
02    L = 1
03    return L
04 if s_{s_1+1} == 0                 // 04–06. sor: s_1 tesztelése konstans idő alatt
05    L = 0
06    return L
07 w_1 = n                           // 07-12. sor: az első súlypont és tartalék számítása
08 j = n
09 while s_j ≤ 1 és j > 0
10        w_1 = w_1 − 1
11        j = j − 1
12 r_1 = w_1 − 1 + s_1
13 for i = 2 to n − 1                // 13–21. sor: s tesztelése
14    j = w_{i−1}                    // 14–17. sor: új súlypont kiszámítása
15    while s_j ≤ i and j > 0
16          w_i = w_i − 1
17          j = j − 1
18    if w_i ≥ i                     // 18–??. sor: s grafikus?

19       if s_i > w_i + r_{i−1}
20          L = 0                    // 20–21. sor: s nem grafikus
```

21 **return** L
22 $r_i = w_i - 1 + r_{i-1} - s_i$ // 22. sor: r_i frissítése
23 **if** $w_i < i$
24 **if** $s_i > w_i + r_{i-1}$
25 $L = 0$ // 25–26. sor: s nem grafikus
26 **return** L
27 $r_i = w_i + r_{i-1} - s_i$ // 22. sor: r_i frissítése
28 $L = 1$ 28–29. sor: s grafikus
29 **return** L

12. Tétel. *A* Havel-Hakimi-lineáris *algoritmus futási ideje legjobb esetben* $\Theta(1)$*, legrosszabb esetben* $\Theta(n)$*.*

Bizonyítás. A 01–06. sorok időigénye $O(1)$, és például a (0^n) bemenetre a program a 03. sorban megáll, ezért a legjobb futási idő $O(1)$. A 07–11. sorok időigénye $\Theta(n)$. Mivel a súlypontok számítása legfeljebb n csökkentést igényel, a 12–29. sorok időigénye $O(n)$, ezért a legrosszabb eset $\Theta(n)$. □

3.9. Példák

1. példa. Legyen az első példánk $s = (3^3, 1)$. A 01–12. sorok szerint $r_1 = 0$. Ha $i = 2$, akkor $w_i = 3$ és a 19. sor feltétele nem teljesül, ezért s *nem* $(0, 1, 4)$*-grafikus.* □

2. példa A következő példa $s = (5, 3^2, 2, 1^3)$. Az 01–12. sorokban azt kapjuk, hogy $w_1 = 7$ és $r_1 = 1$. Ha $i = 2$, akkor $w_i = 4$, a 19. sor feltétele nem teljesül és a 22. sor szerint $r_2 = 1$. Ha $i = 3$, akkor $w_i = 3$ és nem teljesül a 24. sor feltétele. Ha $i = 4$, akkor $w_i = 1$ és most sem teljesül a 24. sor feltétele. Ha $i = 5$, akkor teljesül a 09. sor $s_j \leq 1$ feltétele, és ezért s $(0, 1, 7)$*-grafikus.* □

3. példa Legyen $s = (5, 4, 1^5)$. Erre a sorozatra $r_1 = 1$, és ha $i = 2$, akkor $w_i = 2$, ezért a 24. sor feltétele teljesül, így s *nem* $(0, 1, 7)$*-grafikus.* □

4. példa Utolsó példánk legyen $s = (5^2, 4, 3^4)$. Az első 12 sor szerint $r_1 = 1$. Ha $i = 2$, akkor $w_i = 7$ és $r_2 = 1$. Ha $i = 3$, akkor $w_3 = 7$ és $r_3 = 2$. Ha $i = 4$, akkor teljesül a 15. sor $s_i \leq 1$ feltétele, ezért s $(0, 1, 7)$*-grafikus.* □

A következő táblázatokban bemutatjuk, hogyan oszlanak meg a kizárt grafikus és nemgrafikus sorozatok az egyes menetek között. Azt is jellemezzük, hogy átlagosan hány meneten át kell egy grafikus, illetve nemgrafikus sorozatot a kizárásáig tesztelni, és azt is, hogy a menetek hányadrészét fordítjuk átlagosan egy sorozat tesztelésére.

A 5. táblázat a HHl által az i-edik ($i = 1, \ldots, 11$) menetben kiszűrt nem $(0, 1, n)$-grafikus sorozatok számát mutatja $n = 1, \ldots, 11$ csúcs esetén.

A 6. táblázat HHl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt $(0, 1, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10	11
1	0										
2	1	0									
3	6	0	0								
4	22	2	0	0							
5	85	8	2	0	0						
6	311	35	12	2	0	0					
7	1169	128	58	17	2	0	0				
8	4369	488	239	100	24	2	0	0			
9	16524	1805	942	471	173	32	2	0	0		
10	62650	6800	3601	2021	956	289	43	2	0	0	
11	239008	25571	13677	8147	4561	1877	470	55	2	0	0

5. táblázat. HHl i-edik ($i = 1, \ldots, 11$) menetében a $(0, 1, n)$-szabályos sorozatok közül kiszűrt nemgrafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10	11
1	1										
2	2	0									
3	1	3	0								
4	1	8	2	0							
5	1	16	12	2	0						
6	1	29	48	22	2	0					
7	1	47	130	127	35	2	0				
8	1	72	306	488	290	54	2	0			
9	1	104	618	1492	1475	591	78	2	0		
10	1	145	1158	3863	5757	3868	1112	110	2	0	
11	1	195	1998	8890	18440	18662	9053	1958	149	2	0

6. táblázat. HHl i-edik ($i = 1, \ldots, 11$) menetében a $(0, 1, n)$-szabályos sorozatok közül kiszűrt nemgrafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

Legyen $N_i(a, b, n, \mathrm{A})$, illetve $M_i(a, b, n, \mathrm{A})$ az A algoritmus által az (a, b, n)-szabályos vagy (a, b, n)-páros sorozatok vizsgálata során az i-edik ($i = 1, \ldots, n$) menetben kizárt nemgrafikus, illetve grafikus sorozatok száma, továbbá legyen

$$N = \sum_{i=1}^{n-1} n_i \quad \textit{és} \quad M = \sum_{i=1}^{n-1} m_i, \tag{25}$$

$$X(a, b, n, \mathrm{A}) = \frac{\sum_{i=1}^{n-1} i n_i}{N}, \tag{26}$$

$$Y(a, b, n, \mathrm{A}) = \frac{\sum_{i=1}^{n-1} i m_i}{M}, \tag{27}$$

$$Z(a,b,n,\mathrm{A}) = \frac{\sum_{i=1}^{n-1} i(m_i+n_i)}{N+M}, \tag{28}$$

$$X'(a,b,n,\mathrm{A}) = \frac{\sum_{i=1}^{n-1} in_i}{N(n-1)}, \tag{29}$$

$$Y'(a,b,n,\mathrm{A}) = \frac{\sum_{i=1}^{n-1} im_i}{M(n-1)}, \tag{30}$$

$$Z'(a,b,n,\mathrm{A}) = \frac{\sum_{i=1}^{n-1} i(m_i+n_i)}{(N+M)(n-1)}. \tag{31}$$

Az 7. táblázat a HHl algoritmus hatékonyságát jellemzi $a = 0$, $b = 1$ és $n = 1, \ldots, 11$ csúcs esetén.

n/jellemző	X	Y	Z	X'	Y'	Z'
2	1.000000000	1.000000000	1.000000000	1.000000000	1.000000000	1.000000000
3	1.000000000	1.750000000	1.300000000	0.500000000	0.875000000	0.650000000
4	1.083333333	2.454545455	1.514285714	0.361111111	0.818181818	0.504761905
5	1.126315789	3.032258065	1.595238095	0.281578947	0.758064516	0.398809524
6	1.180555556	3.588235294	1.712121212	0.236111111	0.717647059	0.342424242
7	1.220524017	4.111111111	1.796620047	0.203420670	0.685185185	0.299436674
8	1.262734584	4.629843364	1.897435897	0.180390655	0.661406195	0.271062271
9	1.299062610	5.140793396	1.988235294	0.162382826	0.642599175	0.248529412
10	1.335323852	5.650162338	2.083407305	0.148369317	0.627795815	0.231489701
11	1.368874588	6.157056683	2.174534186	0.136887459	0.615705668	0.217453419

7. táblázat. HHl hatékonysági jellemzői $a = 0$, $b = 1$ és $n = 2, \ldots, 11$ csúcs esetén.

Az 7. táblázat 11. sorában található $X'(0,1,11) = 0.136887459$ és $Y'(0,1,11) = 0.615705668$. Eszerint 11 csúcs esetén a nemgrafikus sorozatok kiszűréséhez átlagosan a menetek 14 %-ára, míg a grafikus sorozatok kiszűréséhez átlagosan 62 %-ára van szükség, ahonnan az következik, hogy az összes szűréshez átlagosan a menetek 22 %-át kell végrehajtani.

Érdemes megjegyezni, hogy Tripathi és Vijay ugrópontokról szóló tétele a HHl algoritmus gyorsítására is felhasználható.

4. Általános leszámlálási eredmények

Eddig például Avis és Fukuda [2], Barnes és Savage [3, 4], Burns [11], Erdős és Moser [48], Frank, Savage and Sellers [21], Kleitman és Winston [36], Rødseth, Sellers, Tverberg [57], Ruskey et al. [58], Simion [61], Stanley [69], Winston és Kleitman [75] publikáltak foksorozatok leszámlálására vonatkozó eredményeket. Az általunk vizsgált sorozatok számával kapcsolatos eredmények találhatók Sloane és Ploffe [62], valamint Stanley [68] könyvében és a *The On-Line Encyclopedia of Integer Sequences* című honlapon [64, 65, 66] is.

Ha l, m és u egész számok, továbbá $l \geq u$ és $1 \geq m$, akkor az $s = (s_1, \ldots, s_n)$ (l, u, m)-korlátos sorozatok $B(l, u, m)$ száma

$$B(l, u, m) = (u - l + 1)^m. \tag{32}$$

A (32) képlet közvetlen adódik abból, hogy az s sorozatnak mind az m eleme $u - l + 1$ lehetséges értéket vehet fel.

Az is közvetlenül belátható, hogy ha l, m és u egész számok, továbbá $l \geq u$ és $1 \geq m$, akkor az (l, u, m)-szabályos sorozatok $R(l, u, m)$ száma

$$R(l, u, m) = \binom{m + u - l}{m}. \tag{33}$$

Legyen ugyanis az $s = (s_1, \ldots, s_m)$ (l, u, m)-szabályos sorozat esetén $s' = (s'_1, \ldots, s'_m)$, ahol $s'_i = s_i + m - i$. A lehetséges s és s' sorozatok halmazai között kölcsönösen egyértelmű kapcsolat áll fenn. A különböző s' sorozatok száma pedig annyi, ahány féleképpen a különböző l, $l + 1$, $\ldots$, $u + m - 1$ számok – azaz $u + m - l$ szám – közül m számot ki tudunk választani.

Ha $l = 0$, $u = n - 1$ és $m = n$, akkor az

$$R(0, n - 0, n) = R(n) = \binom{2n - 1}{n} \tag{34}$$

alakot kapjuk.

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

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

$$\frac{R(n+2)}{R(n+1)} > \frac{R(n+1)}{R(n)}, \tag{35}$$

$$\lim R_{n \to \infty} = 4, \tag{36}$$

továbbá

$$\frac{4^n}{\sqrt{4\pi n}}\left(1 - \frac{1}{2n}\right) < R(n) < \frac{4^n}{\sqrt{4\pi n}}\left(1 - \frac{1}{8n + 8}\right). \tag{37}$$

Bizonyítás. A (34) egyenlőség alapján

$$\frac{R(n+2)}{R(n+1)} = \frac{(2n+3)!(n+1)n!}{(n+2)!(n+1)!(2n+1)!} = \frac{4n+6}{n+2} = 4 - \frac{2}{n+2}, \tag{38}$$

ahonnan (35) és (36) is közvetlenül adódik.

(37) belátásához felhasználjuk a Stirling-formula következő alakját [13]: ha $n \geq 1$, akkor

$$n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n} e^{\tau_n} \tag{39}$$

ahol

$$\frac{1}{12n+1} < \tau_n < \frac{1}{12n}. \tag{40}$$

□

1987-ben Ascher [1] a következő képletet vezette le a $(0,1,n)$-páros sorozatok $E(n)$ számára).

14. Lemma. (Ascher [1], Sloane and Pfoffe [62]) *Ha $n \geq 1$, akkor a $(0,1,n)$-páros sorozatok $E(n)$ száma*

$$E(n) = \frac{1}{2}\left(\binom{2n-1}{n} + \binom{n-1}{\lfloor n \rfloor}\right). \tag{41}$$

Bizonyítás. Lásd [1, 62]. □

A (34) képlet és a 14. lemma egybe vetése mutatja, hogy a páros és páratlan sorozatok számának nagyságrendje megegyezik, azonban több a páros sorozat, mint a páratlan. A 14. lemma alapján pontosan meg tudjuk adni $E(n)$ aszimptotikus nagyságrendjét.

15. Lemma. (Iványi, Lucz, Móri, Sótér [30]) *Ha $n \geq 1$, akkor*

$$\frac{E(n+2)}{E(n+1)} > \frac{E(n+1)}{E(n)}, \tag{42}$$

$$\lim E_{n\to\infty} = 4, \tag{43}$$

továbbá

$$\frac{4^n}{\sqrt{\pi n}}(1-\delta_3(n)) < E(n) < \frac{4^n}{\sqrt{\pi n}}(1-\delta_4(n)), \tag{44}$$

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

Bizonyítás. A bizonyítás hasonló a 13. lemma bizonyításához. □

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

16. Következmény. (Iványi, Lucz, Móri, Sótér [30]) *Ha $n \geq 1$, akkor*

$$\frac{E(n+1)}{R(n+1)} < \frac{E(n)}{R(n)} \tag{45}$$

és

$$\lim_{n\leftarrow\infty} \frac{E(n)}{R(n)} = \frac{1}{2}. \tag{46}$$

Bizonyítás. Lásd [30]. □

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.

17. Lemma. *Ha $n \geq 1$, akkor a $(0, 1, n)$-szabályos sorozatok közül*

$$R_z(n) = \binom{2n-2}{n-1} = \frac{n}{2n-1} R(n). \tag{47}$$

tartalmaz legalább egy nullát.

Bizonyítás. A nullát tartalmazó $(0, 1, n)$-szabályos sorozatok halmaza kölcsönösen egyértelműen leképezhető az $(0, n-1, n)$-szabályos sorozatok halmazára. Az utóbbi halmaz elemszáma pedig (34) szerint

$$\binom{2n-2}{n-1} = \frac{(2n-2)!n}{n(n-1)!(2n-1)} = \frac{n}{2n-1}\binom{2n-1}{n} = \frac{n}{2n-1} R(n). \tag{48}$$

□

Egész számokból álló sorozat különböző elemeinek a számát az adott sorozat *szivárványszámának* nevezzük. Legyen $q_n(s)$ valószínűségi változó, amely egy véletlen $(0, 1, n)$-korlátos sorozat szivárványszámát jellemzi. $q_n(b)$ szivárványszámának várható értékét és szórását a következő állítás tartalmazza.

18. Lemma. (Iványi, Lucz, Móri, Sótér [30]) *Legyen σ egy véletlen $(0, n-1, n)$-korlátos sorozat és $q_n(\sigma)$ a szivárványszáma. Ekkor σ $E[q_n(\sigma)]$ várható értéke és $V[q_n(\sigma)]$ szórása a következő:*

$$E[q_n(\sigma)] = n\left[1 - \left(1 - \frac{1}{n}\right)^n\right] = n\left(1 - \frac{1}{e}\right) + O(1), \tag{49}$$

$$\begin{aligned} Var[q_n(\sigma)] &= n\left(1-\frac{1}{n}\right)^n\left[1-\left(1-\frac{1}{n}\right)^n\right] \\ &\quad + n(n-1)\left[\left(1-\frac{2}{n}\right)^n - \left(1-\frac{1}{n}\right)^{2n}\right] \\ &= \frac{n}{e}\left(1-\frac{2}{e}\right) + O(1). \end{aligned} \tag{50}$$

A következő állítás a k szivárványszámú $(0, n-1, n)$-szabályos sorozatok számát adja meg.

19. Lemma. (Iványi, Lucz, Móri, Sótér [30]) *Ha* $1 \le k \le n$ *és* $m \ge 1$, *akkor a* k *szivárványszámú* $(0, n-1, m)$*-szabályos sorozatok* $S(k, m, n)$ *száma*

$$S(k,m,n) = \binom{n}{k}\binom{m-1}{k}, \; k = 1, \ldots, n. \tag{51}$$

Eszerint a véletlen σ $(0, n-1, m)$-szabályos sorozatok $r_n(\sigma)$ szivárványszáma hipergeometriai eloszlású az $n+m-1$, n és m paraméterekkel. Legyen $\rho_n(\sigma)$ egy véletlen $(0,1,n)$-szabályos sorozat és $E[r_n(\sigma)]$, illetve $V[r_n(\sigma)]$ σ várható értéke, illetve szórása. Ekkor $\rho_n(\sigma)$ szivárványszámának várható értékét és szórását a következő állítás tartalmazza.

20. Következmény. (Iványi, Lucz, Móri, Sótér [30]) *Legyen* ρ *egy véletlen* $(0,1,n)$*-szabályos sorozat. Ekkor* ρ $E[r_n(\rho)]$ *várható értéke és* $V[r_n(\rho)]$ *szórása a következő:*

$$E[r_n(\rho)] = \frac{n^2}{2n-1} = \frac{n}{2} + \frac{n}{4n-2} = \frac{n}{2} + O(1), \tag{52}$$

$$V[r_n(b)] = \frac{n^2(n-1)}{2(2n-1)^2} = \frac{n}{8} + \frac{n}{128n^2 - 128n + 32} = \frac{n}{8} + O(1). \tag{53}$$

A pontos algoritmusokról szóló 3.1. részben beláttuk, hogy elég a $(0,1,n)$-páros sorozatok nullamentes prefixét megvizsgálni ahhoz, hogy eldöntsük, grafikus-e a vizsgált sorozat. Mivel a 17. lemma szerint a 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 páros sorozatot elemezzük (az átlagos futási idő vagy $G(n)$ meghatározása érdekében), nagyon hasznos a következő lemma.

Legyen $G_z(n)$ a nullamentes grafikus n-páros sorozatok száma.

21. Lemma. (Iványi, Lucz, Móri, Sótér [30]) *Ha* $n \ge 2$, *akkor a* $(0,1,n)$*-grafikus sorozatok száma*

$$G(n) = G_z(n) + G(n-1). \tag{54}$$

Bizonyítás. A $(0,1,n)$-grafikus sorozatokban vagy $s_n = 0$, vagy $s_n > 0$. Az előbbiekben vagy $s_1 = n-1$, vagy $s_1 < n_1$. Ha $s_1 = n-1$ és $s_n = 0$, akkor az s sorozat biztosan nem grafikus, mert nincs benne elég pozitív elem. Az $s_1 < n-1$ és $s_n = 0$ tulajdonságú sorozatok $n-1$ hosszú fejei pontosan a $(0,1,n-1)$-grafikus sorozatok. □

A grafikus sorozatok $G(n)$ számának jellemzésével kapcsolatos kutatások ígéretes iránya a páros számok pozitív összeadandókra való felbontása, és annak vizsgálata, hogy az ilyen felbontások közül melyek $(0,1,n)$-grafikusak [3, 4, 11]. Ezek segítségével sikerült a grafikus sorozatok számára vonatkozó alábbi aszimptotikus korlátokat bizonyítani.

22. Lemma. (Burns [11]) *Léteznek olyan pozitív c és C állandók, hogy a $(0,1,n)$-grafikus sorozatok $G(n)$ száma a következő korlátok közé esik:*

$$\frac{4^n}{cn} < G(n) < \frac{4^n}{(\log n)^C \sqrt{n}} \,. \tag{55}$$

Nézzük meg, mit várhatunk a HHl algoritmus első hat sorától. Az algoritmus lehetséges bemenetei a $(0, n-1, n)$-szabályos sorozatok. Ezek $R(n)$ száma a (34) képlet szerint

$$R(n) = \binom{2n-1}{n}. \tag{56}$$

HHl első három sora kiszűri például azokat a sorozatokat, amelyek $(n-1)$-gyel kezdődnek és nullával végződnek. Ezek száma (33) szerint

$$B(0, n-1, n-2) = \binom{2n-3}{n-2}. \tag{57}$$

Ezek közül a HHl által kiszűrt sorozatok $R_1(n)$ hányada

$$R_1(n) = \frac{\binom{2n-3}{n-2}}{\binom{2n-1}{n}} = \frac{2(2n-1)}{n} = \frac{1}{4} + \frac{1}{8n-4}\,. \tag{58}$$

HHl pontosan azokat a sorozatokat szűri ki, amelyek $(n-i)$-vel $(i = 1, \ldots, n-2)$ kezdődnek és legalább i nullát tartalmaznak. Rögzített i-re az ilyen sorozatok aszimptotikus részaránya $1/4^i$, így HHl aszimptotikusan a szabályos sorozatokból a

$$\sum_{i=1}^{\infty} \frac{1}{4^i} = \frac{1}{3} \tag{59}$$

összegnek megfelelő hányadot, azaz egy harmad részét szűri ki.

Mivel a grafikus sorozatok aszimptotikus sűrűsége nulla, ezért minden A pontos algoritmusra létezik egy $s_{1,A} + s_{2,A} + \ldots = 1$ sor (valószínűség-eloszlás), amelyben s_i az i-edik menetben kiszűrt hányad. Például $s_{1,A} = 1/3$ minden olyan pontos algoritmusra, amelyik első menetben a PT algoritmust (vagy annak valamilyen lassú változatát) használja az első menetben – ilyen a HH és az EG is.

5. Tesztelő algoritmusok

Sorozatok megvalósíthatóságának vizsgálata során természetes észrevétel, hogy az s 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 „pozitív”, majd egy „paritásos”, egy „binomiális” és végül egy „fejfelező” tesztelő/szűrő algoritmust mutatunk be.

5.1. 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. Legyen p az s sorozat pozitív elemeinek a száma.

23. Következmény. *Ha $n \geq 1$ és $s = (s_1, \ldots, s_n)$ $(0, 1, n)$-grafikus sorozat, akkor*

$$s_1 \leq p - 1 \quad \textit{vagy} \quad s_1 = 0. \tag{60}$$

Bizonyítás. A (60) 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 23. következményen alapuló tesztet a következő algoritmus végzi.

Bemenet. n: a csúcsok száma $(n \geq 1)$;
$s = (s_1, \ldots, s_n)$: $(0, 1, n)$-szabályos sorozat.

Kimenet. L: s grafikusságát jelző logikai változó.

Munkaváltozó. p: a bemenetben lévő pozitív elemek száma.

```
Pozitív-teszt(n, s, L)
L = 0
p = n
while s_p == 0
        p = p - 1
if s_1 > p - 1
    return L
L = 1
return L
```

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

Ennek az algoritmusnak a javított változata az alábbi Gyors-teszt (GyT) [44].

```
Gyors-teszt(n, s, L)
if s_{s_1+1} == 0
    L = 0
    return L
```

A Gyors-teszt ugyanazt az eredményt adja, mint Pozitív-teszt, a futási ideje azonban mindig $\Theta(1)$.

5.2. 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 grafikus sorozatok.

24. LEMMA. *Ha $n \geq 1$ és s $(1,1,n)$-grafikus sorozat, akkor*

$$H_n \quad \textit{páros}. \tag{61}$$

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

Ezt az állítást a 2. tétel következményeként is megkaphatjuk. A 24. lemmában javasolt tesztet a következő algoritmus végzi.

Bemenet. n: a csúcsok száma $(n \geq 1)$;
$s = (s_1, \ldots, s_n)$: $(0,1,n)$-szabályos sorozat.
Kimenet. L: s grafikusságát jelző logikai változó.
Munkaváltozó. i: ciklusváltozó.

PARITÁS-TESZT(n, s, L)
$L = 0$
$H_1 = 0$
for $i = 2$ **to** n
    $H_i = H_{i-1} + s_i$
if H_n páratlan
    return L
$L = 1$
return L

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

5.3. Binomiális teszt (Bt)

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 az s sorozat pozitív elemeinek a száma.

25. LEMMA. *Ha $n \geq 1$ és s $(0,1,n)$-grafikus sorozat, akkor*

$$2H_i \leq i(i-1) + T_i \quad (i = 1, \ldots, p). \tag{62}$$

Bizonyítás. A (62) 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 25. lemmában javasolt tesztet végzi el a következő program.

Bemenet. n: a csúcsok száma ($n \geq 1$);
$s = (s_1, \ldots, s_n)$: páros sorozat;
$H = (H_1, \ldots, H_n)$: H_i az s sorozat első i elemének összege.

Kimenet. L: s grafikusságát jelző logikai változó.

Munkaváltozók. i: ciklusváltozó;
p: az s sorozat pozitív elemeinek a száma.

```
Binomiális-teszt(n, s, L)
p = n
while s_p == 0
        p = p − 1
if p == 1
    L = 0
    return L
H_1 = s_1
for i = 2 to p
      H_i = H_{i−1} + s_i
for i = 1 to p
      if 2H_i > i(i − 1) + H_p
          L = 0
          return L
L = 1
return L
```

Az algoritmus azért kezdi s végénél p meghatározását, mert a 21. 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. A várható futási idő szempontjából viszont a konstans idő alatt hatékony Gyors-teszt hasznos lehet.

Felmerült, hogy a Binomiális-teszt algoritmust is csak az ellenőrző pontokon alkalmazzuk, a szimulációs kísérletek azonban azt mutatták, hogy ezzel csökkenne az algoritmus hatékonysága.

n helyett p viszont gyengítené az algoritmust, mert például a rossz $(2, 2, 0)$ sorozatot *nem* szűrné ki. Ha azonban csak a páros nullamentes sorozatokat vizsgáljuk, a

$2, 2, 0$ és hasonló sorozatokat egyetlen algoritmusunk sem kell tesztelnie (mert ezeket már a bemenő sorozatok előállítása során kiszűrjük).

5.4. Fej felezése (Ft)

Az s sorozat fokpárosító lehetőségeinek az eddigieknél pontosabb becslését kaphatjuk, ha a fejet két részre osztjuk. Legyen $\lfloor i/2 \rfloor = h_i$. Ekkor az $(s_1, \ldots, s_{h_i})$ sorozatot az i indexhez tartozó fej *elejének*, az $(s_{h_i+1}, \ldots, s_i)$ sorozatot pedig az i indexhez tartozó fej *végének* nevezzük.

26. Lemma. *Ha $n \geq 1$ és s $(0, 1, n)$-grafikus sorozat, akkor*

$$\begin{aligned} H_i \leq \min(&\min(H_{h_i}, T_n - T_i, h_i(n-i)) \\ &+ \min(H_i - H_{h_i}, T_n - T_i, (i-h_i)(n-i)), T_i) \\ &+ \min(h_i(i-h_i) + \binom{h_i}{2} + \binom{i-h_i}{2} \quad (i = 1, \ldots, n), \end{aligned} \tag{63}$$

továbbá

$$\min(H_{h_i}, T_n - T_i, h_i(n-i)) + \min(H_i - H_{h_i}, T_n - T_i, (i-h_i)(n-i)) \leq T_i. \tag{64}$$

Bizonyítás. Legyen G az s 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

$$X_{i,1} \leq \min(H_{h_i}, T_n - T_i, h_i(n-i)). \tag{65}$$

Hasonló gondolatmenettel kapjuk, hogy

$$X_{i,2} \leq \min(H_i - H_{h_i}, T_n - T_i, (i-h_i)(n-i)). \tag{66}$$

$X_{i,3}$ legfeljebb $h_i(i - h_i)$ és legfeljebb H_i, ezért

$$X_{i,3} \leq \min(h_i(i-h_i), H_i). \tag{67}$$

$X_{i,4}$ legfeljebb $\binom{h_i}{2}$ és legfeljebb H_{h_i}, így

$$X_{i,4} \leq \min\left(\binom{h_i}{2}, H_{h_i})\right), \tag{68}$$

míg $X_{i,5}$ legfeljebb $\binom{i-h_i}{2}$ és legfeljebb $H_i - H_{h_i}$, ahonnan

$$X_{i,5} \leq \min\left(\binom{i-h_i}{2}, H_i - H_{h_i}, \right). \tag{69}$$

Az is követelmény, hogy a farok részei együtt nem léphetik túl a farok kapacitását, azaz telesüljön

$$X_{i,1} + X_{i,2} \leq T_i. \tag{70}$$

A (65), (66), (67), (68) és (69) egyenlőtlenségeket összegezve azt kapjuk, hogy

$$H_i \leq X_{i,1} + X_{i,2} + X_{i,3} + 2X_{i,4} + 2X_{i,5}. \tag{71}$$

Ha az (65), (66), (67), (68) és (69) egyenlőtlenségeket a (71) egyenlőtlenségbe helyettesítjük, akkor (63) adódik, míg (70) ekvivalens a (64) egyenlőtlenséggel. $\square$

A 26. lemmában javasolt tesztet a következő algoritmus végzi.

Bemenet. n: a csúcsok száma ($n \geq 1$);
$s = (s_1, \ldots, s_n)$: n-faroktesztelt sorozat;
$H = (H_1, \ldots, H_n)$: H_i az s sorozat első i elemének összege;
$T = (T_1, \ldots, T_n)$: T_i az s sorozat utolsó $n - i$ elemének összege.

Kimenet. L: s grafikusságát jelző logikai változó.

Munkaváltozók. i: ciklusváltozó;
$X = (X_1, X_2, X_3, X_4, X_5)$: X_j a fej vége $X_{i,j}$ paraméterének aktuális értéke.

Fejfelező-teszt(n, s, H, T, p, L)

for $i = 2$ **to** $n - 1$
    $h = \lfloor i/2 \rfloor$
    $X_1 = \min(H_h, T_n - T_i, h(n - i))$
    $X_2 = \min(H_i - H_h, T_n - T_i, (i - h)(n - i))$
    $X_3 = \min(h(i - h), H_i)$
    $X_4 = \binom{h}{2}$
    $X_5 = \binom{i-h}{2}$
    if $H_i > X_1 + X_2 + X_3 + 2X_4 + 2X_5$ vagy $X_1 + X_2 > T_i$
        $L = 0$
        return L
$L = 1$
return L

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 szerint ez nem csökkentené a várható futási időt.

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 $E_{\mathrm{A}}(n)$-nel jelöljük és az A algoritmus n hosszúságú sorozatokra vonatkozó *hatékonyságának* nevezzük.

A következő közelítő algoritmusokat vizsgáljuk:

1) Nullamentesítő-teszt (Nt);
2) Binomiális-teszt (Bt);
3) Fejfelező-teszt (Ft).

A 8. táblázat a binomiális és faroktesztelt sorozatok számát, továbbá a (0,1,n)-grafikus sorozatok számát és a grafikus sorozatok száma szomszédos n helyeken felvett értékei hányadosát tartalmazza $n = 1, \ldots, 29$ csúcs esetén.

A 9. táblázat azt jellemzi, hogy a vizsgált közelítő algoritmusok a szabályos sorozatoknak milyen hányadát szűrik ki. A táblázat a nullamentes páros sorozatok száma $(E_z(n))$ mellett tartalmazza a nullamentes binomiális $(B_z(n))$, a nullamentes faroktesztelt $(F_z(n))$ és a grafikus sorozatok $(G(n))$ számának, valamint a szabályos sorozatok számának hányadosát.

A 10. táblázat a Binimiális-teszt és a Fejfelező-teszt algoritmusok futási idejét adja meg másodpercben és műveletszámban $n = 1, \ldots, 20$ csúcsra.

Ha $n = 2$, akkor (34) szerint $R(n) = \binom{3}{2} = 3$ $(0, 1, n)$-szabályos sorozat van: $(1, 1)$, $(1, 0)$ és $(0, 0)$. Az n hosszúságú páros sorozatok számát $E(n)$-nel jelöljük. Ezzel a jelöléssel $E(2) = 2$. A Binomiális-teszt által elfogadott, n hosszúságú sorozatok számát $B(n)$-nel jelölve $B(2) = 2$. Az n hosszúságú grafikus sorozatok számát jelöljük $G(n)$-nel. Ekkor $G(2) = 2$ és a Binomiális-teszt hibája (hatékonysága) $R_{\mathrm{BT}}(2) = 2/2 = 1$.

n	$B_z(n)$	$F_z(n)$	$G_z(n)$	$G(n+1)/G(n)$
1	**1**	**0**	**1**	2.000000
2	**2**	**2**	**2**	2.000000
3	**4**	**4**	**4**	2.750000
4	**11**	**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	3.898843
29			2022337118015338	

8. táblázat. A binomiális $(B_(n))$, faroktesztelt $(F(n))$ $(0,1,-n)$-szabályos sorozatok száma, valamint a $(0,1,n)$-grafikus sorozatok száma (G_n) és a grafikus sorozatok halmazának szomszédos n helyeken felvett számosságai hányadosa $(G(n+1))/G(n)$ $n = 1,\ldots,28$ csúcs esetén.

Ha $n = 3$, akkor a szabályos sorozatok száma $R(n) = 10$. Ezek közül a (2,2,2), (2,2,0), (2,1,1), (2,0,0), (1,1,0) és (0,0,0) páros, azaz $E(3) = 6$. Ezek közül a Binomiális-teszt kizárja a (2,2,0) és (2,0,0) sorozatokat, így $B(3) = 4$. A megmaradt 4 sorozat grafikus, így $F(3) = G(3) = 4$.

Ha $n = 4$, akkor a szabályos sorozatok száma $R(4) = 35$. Ezek közül 19 a páros, és a következő 11 grafikus: (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 19 páros sorozat közül

n	$E_z(n)$	$E_z(n)/R(n)$	$B_z(n)/R(n)$	$F_z(n)/R(n)$	$G(n)/R(n)$
1	0	0.000000	1.000000	1.000000	1.000000
2	1	0.333333	0.666667	0.666667	0.666667
3	2	0.300000	0.400000	0.400000	0.400000
4	9	0.257143	0.314286	0.314286	0.314286
5	28	0.230159	0.246032	0.246031	0.246032
6	110	0.238095	0.222943	0.220779	0.220779
7	396	0.231352	0.203380	0.200466	0.199301
8	1519	0.236053	0.195183	0.191142	0.188500
9	5720	0.235335	0.188276	0.183793	0.179391
10	21942	0.237524	0.184460	0.179502	0.173375
11	83980	0.238098	0.181290	0.175977	0.168260
12	323554	0.239301	0.179145	0.173508	0.164278
13	1248072	0.240000	0.177368	0.171500	0.160821
14	4829708	0.240784	0.176014	0.169960	0.157882
15	18721080	0.241379	0.174884	0.168684	0.155271
16	72714555	0.241946	0.173965	0.167634	0.152950
17	282861360	0.242424	0.173188	0.166738	0.150844
18	1101992870	0.242860	0.172533	0.165972	0.148926
19	4298748300	0.243243	0.171970	0.165306	0.147158
20	16789046494	0.243590	0.171486	0.164725	0.145521
21					0.143997
22					0.142569
23					0.141228
24					0.139961
25					0.138762
26					0.137625
27					0.136542
28					0.135509
29					0.134521

9. táblázat. A nullamentes sorozatok száma, továbbá a binomiális/szabályos, fejtesztelt/szabályos és grafikus/szabályos számarányok.

Binomiális-teszt is kizárja azt a nyolc sorozatot, amelyeket az Erdős-Gallai kizárna, így $B(4) = F(4) = G(4) = 11$.

Az $R(5) = 126$ szabályos sorozat közül $E(5) = 66$ a páros, ezek között pedig $B(5) = 31$ a binomiális. Ezek a sorozatok mind grafikusak, azaz $F(5) = G(5) = 31$.

Az $R(6) = 462$ szabályos sorozat közül $E(6) = 236$ a páros, amelyek között $B(6) = 103$ binomiális sorozat van. Binomiális-teszt a 102 grafikus 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 grafikus sorozatokat, a 6 hosszú sorozatokra azonban már csak közelítő algoritmus. A Fejfelező-

n	Bt, s	Bt, művelet	Ft, s	Ft, művelet
1	0	14	0	15
2	0	41	0	43
3	0	180	0	200
4	0	716	0	815
5	0	2 918	0	3 321
6	0	11 918	0	13 675
7	0	48 952	0	56 299
8	0	201 734	0	233 182
9	0	831 374	0	964 121
10	0	3 426 742	0	3 988 542
11	0	14 107 824	0	16 469 036
12	0	58 028 152	0	67 929 342
13	0	238 379 872	0	279 722 127
14	0	978 194 400	1	1 150 355 240
15	2	4 009 507 932	3	4 724 364 716
16	6	16 417 793 698	13	19 379 236 737
17	26	67 160 771 570	51	79 402 358 497
18	106	274 490 902 862	196	324 997 910 595
19	423	1 120 923 466 932	798	1 328 948 863 507
20	1 627	4 573 895 421 484	3 201	5 429 385 115 097

10. táblázat. A Binomiális-teszt (Bt) és a Fejfelező-teszt (Ht) futási ideje másodpercben és a műveletek számával megadva $n = 1, \ldots, 20$ csúcs esetén.

teszt ezzel a sorozattal is megbirkózik, ezért $F(6) = G(6) = 102$.

Az $R(7) = 1716$ szabályos sorozat között $E(6) = 868$ a páros, melyek közül $B(7) = 376$ a binomiális. A binomiális sorozatok között még 34 rossz van, melyek közül a Pozitív-teszt a 27 grafikus sorozat mellett a következő 7 rosszat is elfogadja: $(6,6,6,4,4,4,2)$, $(6,6,5,4,4,4,1)$, $(6,6,4,4,4,3,1)$, $(6,6,4,3,3,3,1)$, $(6,6,3,3,3,2,1)$, $(6,5,3,3,3,1,1)$, $(5,5,3,3,3,1,0)$. A következő Fejfelező-teszt ezek közül a $(6,6,4,3,3,3,1)$ kivételével mindet kiszűri, így $F(7) = 343$. A cikkben nem ismertetett 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 a Farokfelező-teszt a 7 hosszú bemenetek közük $T(7) = 342$ sorozatot fogad el, így $G(7) = 342$.

A **??**. táblázatban minden sorban az első pontos értéket félkövéren írtuk. Eszerint $n \leq 4$ esetén $B(n) = G(n)$, 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 nemgrafikus, $n = 6$ esetén már hatszor hibázik.

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

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

11. táblázat. Az elvégzett műveletek száma n függvényében a HHr, HHe, EG, EGu, és EGl algoritmusok esetén.

TESZT pedig $n = 7$-ig.

A 1. táblázatban $R(n)$ értéke $n = 24$-ig az OEIS A001700 sorozata [64], $E(n)$ értéke $n = 23$-ig az EIS A005654 sorozata [66], a 8. táblázatban $G(n)$ értéke pedig $n = 23$-ig az EIC A0004251-es sorozata [65]. A többi értéket mi határoztuk meg: $R(25), \ldots, R(40)$, $E(24), \ldots, E(40)$, valamint $B(n)$ és $F(n)$ értékek nem szerepelnek az EIS-ben.

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

A témakörben vannak párhuzamos eredmények is [52, 60, 67]. Saját párhuzamos eredményeinket a 10. részben ismertetjük.

7. Pontos algoritmusok futási ideje

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 11. ábra tartalmazza $n = 1, \ldots, 15$ csúcsra. A sorozatok előállításához szükséges műveleteket beszámítottuk. are included.

A 11. ábra első két oszlopának összehasonlítása azt mutatja, hogy HHe lénygesen

n	$E(n)$	$T(n)$, s	$Op(n)$	$T(n)/E(n)/n$, s	$Op(n)/E(n)/n$
2	2	0	37	0	9.25000000000
3	6	0	148	0	8.22222222222
4	19	0	585	0	7.69736842105
5	66	0	2 339	0	7.08787878788
6	236	0	9 539	0	6.73658192090
7	868	0	38 984	0	6.41606319947
8	3 235	0	160 126	0	6.18724884080
9	12 190	0	656 575	0	5.98464132714
10	46 252	0	2 692 240	0	5.82080774885
11	176 484	0	11 018 710	0	5.67587378511
12	676 270	0	45 049 862	0	5.55126675243
13	2 600 612	0	183 917 288	0	5.44005937537
14	10 030 008	1	750 029 671	0.00000000712149	5.34132654018
15	38 781 096	5	3 055 289 271	0.00000000859525	5.25219687963
16	150 273 315	23	12 434 367 770	0.00000000956590	5.17156346504
17	583 407 990	79	50 561 399 261	0.00000000796537	5.09797604337
18	2 268 795 980	297	205 439 740 365	0.00000000727258	5.03056202928

12. táblázat. Az Erdős-Gallai-lineáris algoritmus teljes és amortizált futási ideje másodpercben és a műveletek számában

gyorsabb, mint HHr, különösen ha n nő. A harmadik és negyedik oszlop összehasonlítása azt mutatja, hogy a futási idő lényegesen csökken, ha a csak az ugró pontokban kell az elemeket tesztelni. Végül a harmadik és ötödik oszlopok együtt a lineáris algoritmusnak a négyzetessel szembeni előnyét jelzik.

A 12. táblázat Erdős-Gallai-Linear futási idejét tartalmazza másodpercben és az elvégzett műveletek számával megadva, továbbá az egy páros sorozatra jutó amortizált műveletszámot.

A 12. táblázat legérdekesebb adatai az utolsó oszlopban vannak. Azt mutatják, hogy a műveletek számát osztva a vizsgált sorozatok hosszával és számával monoton csökkenő sorozatot kapunk (lásd [58]).

A 13. táblázat a $(0, 1, n)$-grafikus sorozatok első elem szerinti eloszlását mutatja $n = 1, \ldots, 12$ csúcs esetén. Ezek az adatok hasznosak az Erdős-Gallai-leszámláló algoritmus tervezéséhez (a feladat szeletekre osztásához).

A 13. táblázatban azt látjuk, hogy a gyakoriságok $(n = 6)$-tól nőnek $(n-2)$-ig, és az utolsó pozitív érték kisebb, mint az utolsó előtti.

8. $(0, b, n)$-gráfok

Ebben a részben a klasszikus tételek $(0, b, n)$-gráfokra való kiterjesztésével foglalkozunk.

n/b_1	0	1	2	3	4	5	6	7	8	9	10	11
1	1											
2	1	1										
3	1	1	2									
4	1	1	4	4								
5	1	2	7	10	11							
6	1	3	10	22	35	31						
7	1	3	14	34	78	110	102					
8	1	4	18	54	138	267	389	342				
9	1	4	23	74	223	503	968	1352	1213			
10	1	5	28	104	333	866	1927	3496	4895	4361		
11	1	5	34	134	479	1356	3471	7221	12892	17793	16016	
12	1	6	40	176	661	2049	5591	13270	27449	47757	65769	59348

13. táblázat. A $(0, 1, n)$-grafikus sorozatok eloszlása s_1 szerint, $n = 1, \ldots, 12$ csúcs esetén

8.1. Erdős-Gallai tétel és Chungphaisan tétele

1974-ben Chungphaisan [15] mind az Erdős-Gallai tételt, mind pedig a Havel-Hakimi tételt kiterjesztette $(0, b, n)$-gráfokra. Az EG tétel kiterjesztése a következő.

27. TÉTEL. (Chungphaisan [15]) *Legyen $n \geq 1$. A $(0, b, n)$-szabályos $s = (s_1, \ldots, s_n)$ sorozat akkor és csak akkor $(0, b, n)$-grafikus, ha*

$$\sum_{i=1}^{n} s_i \quad \textit{páros} \tag{72}$$

és

$$\sum_{i=1}^{j} s_i - bj(j-1) \leq \sum_{k=j+1}^{n} \min(bi, s_k) \quad (j = 1, \ldots, n-1). \tag{73}$$

Bizonyítás. See [15]. □

A tételen alapuló algoritmus legrosszabb esetben négyzetes időt igényel. A következő állítás lehetővé teszi, hogy a $(0, b, n)$-szabályos sorozatokat legrosszabb esetben $\Theta(n)$ idő alatt teszteljük.

28. TÉTEL. (Iványi, [28]) *Ha $n \geq 1$, a $(0, b, n)$-szabályos $s = (s_1, \ldots, s_n)$ sorozat akkor és csak akkor $(0, b, n)$-grafikus, ha*

$$H_n \quad \textit{páros} \tag{74}$$

és

$$H_i > bi(y_i - 1) + H_n - H_y \quad (i = 1, \ldots, n-1), \tag{75}$$

ahol

$$y_i = \max(i, w_i) \quad (i = 1, \ldots, n-1). \tag{76}$$

Bizonyítás. □

A következő Chungphaisan-Erdős-Gallai-lineáris algoritmus (ChEGl) – amely az EG algoritmus természetes általánosítása – $O(n)$ idő alatt eldönti, hogy egy $(0, b, n)$-szabályos sorozat $(0, b, n)$-grafikus-e.

Bemenet. n: csúcsok száma ($n \geq 1$);
$s = (s_1, \ldots, s_n)$: $(0, b, n)$-szabályos sorozat;
b: a gráf két csúcsa között megengedett élek maximális száma.

Kimenet. L: s grafikusságát jelső logikai változó.

Munkaváltozók. i: ciklus változó;
$w = (w_1, \ldots, w_n)$: w_i az i indexhez tartozó súlypont.

Chungphaisan-Erdős-Gallai-lineáris(n, s, b, L)

```
01 H_1 = s_1                       // 01 sor: H_1 kezdeti értékének beállítása
02 for i = 2 to n − 1              // 02–03. sor: H további elemeinek számítása
03     H_i = H_{i−1} + s_i
04 if H_n páratlan                 // 04–06. sor: paritás ellenőrzése
05    L = 0                        // 05–06. sor: páratlan sorozat elutasítása
06    return
07 w = n                           // 07. sor: első súlypont értékének beállítása
08 for i = 1 to n − 1              // 08–16. sor: s tesztelése
09     while s_w < ib és w > 0
10          w = w − 1
11     y = max(i, w)
12     if H_i > bi(y − 1) + H_n − H_y
13        L = 0
14        return L                 // 14. sor: s nem grafikus
15 L = 1                           // 15–16. sor: s grafikus
16 return L
```

29. Tétel. (Iványi, [28]) ChEGl *futási ideje minden esetben* $\Theta(n)$.

Bizonyítás. A 01–06 sorok végrehajtása $\Theta(n)$ időt igényel. Mivel w szigorúan monoton csökken a program végrehajtása során, ezért a 07–14 sorok $O(n)$ időt igényelnek, így az algoritmus futási ideje minden esetben $\Theta(n)$. □

Legyen $b = 3$ és $s = (13, 10, 5, 5, 4, 1)$. $H_6 = 38$ páros. Ha $i = 1$, akkor $w_i = y = 5$ és a 11. sor feltétele $(13 \leq 3 \cdot 1 \cdot (5 - 1))$ nem teljesül. Ha $i = 2$, akkor viszont $w_i = y = 2$ és a feltétel teljesül $(23 > 3 \cdot 2 \cdot (2 - 1)) + 5 + 5 + 4 + 1)$, ezért s *nem* $(0, 3, 6)$*-grafikus.*

Maradjon b 3, de s-et változtassuk meg: legyen $s' = (13, 10, 5, 5, 4, 3)$. Az előző példához képest a futás során az első változás az, hogy amikor $i = 2$, akkor $(23 \leq 3 \cdot 2 \cdot (2 - 1)) + 5 + 5 + 4 + 3$, és így a 11. sorban lévő feltétel nem teljesül, és ugyanez az eredmény $i = 3$, 4 és 4 esetén is, ezért s' $(0, 3, 6)$*-grafikus.*

A 14. táblázat az (a, b, n)-szabályos és (a, b, n)-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs, valamint $a = 0$ és $b = 1$, $a = 0$ és $b = 2$, $a = 2$ és $b = 5$ esetén. A szabályos sorozatok számát a (34) képlettel, az (a, b, n)-grafikus sorozatok számát pedig a Chungphaisan-Erdős-Gallai-lineáris algoritmussal határoztuk meg. Az utolsó oszlop elemeinek meghatározásánál hasznosítottuk a 33. következményt.

n	$R(0,1,n)$	$G(0,1,n)$	$R(0,2,n)$	$G(0,2,n)$	$R(2,3,n)$	$G(2,5,n)$
1	1	1	1	1	1	1
2	3	2	6	3	10	4
3	10	4	35	10	84	23
4	35	11	210	52	715	189
5	126	31	1287	283	6188	1582
6	462	102	8008	1706	54264	13583
7	1716	342	50388	10436	480700	122345
8	6435	1213	319770	65370	4292145	1092573
9	24310	4361	2042975	413111	38567100	9816598
10	92378	16016	13123110	2633537	348330136	88680716
11	352716	59348	84672315	16882153	3159461968	804480107

14. táblázat. Az (a, b, n)-szabályos és (a, b, n)-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs, valamint $a = 0$ és $b = 1$, $a = 0$ és $b = 2$, $a = 2$ és $b = 5$ esetén.

A következő táblázatokban bemutatjuk, hogyan oszlanak meg a kizárt grafikus és nemgrafikus sorozatok az egyes menetek között. Azt is jellemezzük, hogy átlagosan hány meneten át kell egy grafikus, illetve nemgrafikus sorozatot a kizárásáig tesztelni, és azt is, hogy a menetek hányadrészét fordítjuk átlagosan egy sorozat tesztelésére.

A 15. táblázat a ChEGl i-edik $(i = 1, \ldots, 11)$ menetében kiszűrt nemgrafikus sorozatok számát tartalmazza $a = 0$, $b = 2$ és $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10
1	0									
2	3	0								
3	22	3	0							
4	132	26	2	0						
5	824	164	31	4	0					
6	5 084	1 026	276	75	3	0				
7	31 902	6 288	2 018	829	111	5 0				
8	201 366	39 090	13 282	7 231	1 837	203	4	0		
9	1 281 918	244 833	84 340	53 594	20 681	4 259	298	6	0	
10	8 207 232	1 548 774	529 578	365 461	183 262	59 726	8 709	470	5	0
11	52 819 163	9 866 545	3 331 910	2 385 963	1 404 590	632 058	155 070	17 213	660	7

15. táblázat. ChEGL i-edik $(i = 1, \ldots, 11)$ menetében kiszűrt nem $(0, 2, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

A **??**. táblázat a ChEGl i-edik $(i = 1, \ldots, 11)$ menetében kiszűrt $(0, 2, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

A következő 17. táblázat a ChEGl algoritmus hatékonyságát jellemzi $a = 0$, $b = 2$ és $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10
1	1									
2	2	0								
3	1	9	0							
4	1	7	42	0						
5	1	10	29	224	0					
6	1	14	49	183	1 297	0				
7	1	18	70	345	1 143	7 658	0			
8	1	23	97	559	2 326	7 262	46 489	0		
9	1	28	125	846	4 038	15 927	46 074	286 007	0	
10	1	34	159	1 191	6 520	29 629	107 724	295 609	1 779 026	0
11	1	40	193	1 624	9 668	50 663	213 399	728 610	1 900 061	11 154 877

16. táblázat. ChEGl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt $(0, 2, n)$-grafikus sorozatok száma $n = 1, \ \ldots, \ 11$ csúcs esetén.

n/jellemző	X	Y	Z	X'	Y'	Z'
2	$1,000000000$	$1,000000000$	$1,000000000$	$1,000000000$	$1,000000000$	$1,000000000$
3	$1,120000000$	$1,900000000$	$1,342857143$	$0,560000000$	$0,950000000$	$0,671428571$
4	$1,187500000$	$2,820000000$	$1,576190476$	$0,395833333$	$0,940000000$	$0,525396825$
5	$1,232649071$	$3,803030303$	$1,759906760$	$0,308162268$	$0,950757576$	$0,439976690$
6	$1,280785891$	$4,788212435$	$1,957042957$	$0,256157178$	$0,957642487$	$0,391408591$
7	$1,322698224$	$5,770438549$	$2,137870128$	$0,220449704$	$0,961739758$	$0,356311688$
8	$1,363989613$	$6,751572493$	$2,320248929$	$0,194855659$	$0,964510356$	$0,331464133$
9	$1,402468979$	$7,733105601$	$2,496464714$	$0,175308622$	$0,966638200$	$0,312058089$
10	$1,439464334$	$8,714770487$	$2,670148311$	$0,159940482$	$0,968307832$	$0,296683146$
11	$1,474743645$	$9,697001722$	$2,839981439$	$0,147474365$	$0,969700172$	$0,283998144$

17. táblázat. ChEGl hatékonysági jellemzői $a = 0$, $b = 2$ és $n = 1, \ \ldots, \ 11$ csúcs esetén.

8.2. Havel-Hakimi tétel és Chungphaisan tétele

Chungphaisan [15] a következő módon terjesztette ki a Havel-Hakimi tételt.

30. Tétel. (Chungphaisan [15]) *Legyen $n \geq 2$ és $b \geq 1$. Az $s = (s_1, \ldots, s_n)$ $(0, b, n)$-szabályos sorozat akkor és csak akkor $(0, b, n)$-grafikus, ha a j-edik b-redukált $w_j^* = (w_1^*, \ldots, w_{n-1}^*)$ sorozat $(0, b, n)$-grafikus minden $1 \geq j \geq n$ indexre.*

Bizonyítás. Lásd [15]. □

A tételen alapuló algoritmus nagyon lassú. A tétel következő javítása azonban lehetővé teszi, hogy a tesztelést legrosszabb esetben is el tudjuk végezni $O(n)$ idő alatt.

31. Tétel. (Iványi, [28]) *Legyen $n \geq 1$ és $b \geq 1$ Nemnegatív egészek $s = (s_1, \ldots, s_n)$ $(0, 1, n)$-szabályos sorozata akkor és csak akkor $(0, b, n)$-grafikus, ha*

$$??? \tag{77}$$

és

$$???. \tag{78}$$

A következő CHUNGPHAISAN-HAVEL-HAKIMI-LINEÁRIS algoritmus (ChHHl) – amely a HH algoritmus természetes általánosítása – $O(n)$ idő alatt eldönti, hogy egy $(0,b,n)$-szabályos gráf $(0,b,n)$-grafikus-e.

Bemenet. n: csúcsok száma ($n \geq 1$);
$s = (s_1, \dots, s_n)$: $(0,b,n)$-grafikus sorozat;
b: a gráf két csúcsa között megengedett élek maximális száma ($1 \leq b \leq 2$).

Kimenet. L: s grafikusságát jelző logikai változó.

Munkaváltozó. i: ciklus változó;
$w = (w_1, \dots, w_n)$: w_i az i indexhez tartozó súlypont.

CHUNGPHAISAN-HAVEL-HAKIMI-LINEÁRIS(n, s, b, L)

01 $L = 0$ // 01. sor: a gyakoribb érték beállítása

02 **if** $s_1 == 0$ // 02–04. sor: a nullákból álló sorozat grafikus

03 $\quad L = 1$

04 $\quad$ **return** L

05 **if** $s_{\lceil s_1/b+1 \rceil} == 0$ // 05–07. sor: s_1 ellenőrzése konstans idő alatt

06 $\quad$ **return** L

07 $H_1 = s_1$ // 07. sor: H_1 kezdeti értékének beállítása

08 **for** $i = 2$ **to** $n - 1$ // 08–09. sor: H további elemeinek számítása

09 $\quad H_i = H_{i-1} + s_i$

10 **if** H_n páratlan // 10–11. sor: paritás tesztelése

11 $\quad$ **return** L

12 $w_1 = n$ // 12. sor: első súlypont kezdeti értékének beállítása

13 **while** $s_{w_1} < b$ and $w_1 > 0$

14 $\qquad w_1 = w_1 - 1$

15 **if** $s_1 > b(w_1 - 1) + H_n - H_{w_1}$

16 $\quad$ **return** L

17 $r_1 = b(w_1 - 1) + H_n - H_{w_1} - s_1$ // 17. sor: első maradék számítása

18 **for** $i = 2$ **to** $n - 1$ // 18–34. sor: s tesztelése

19 $\quad$ **if** $H_{i-1} \geq H_n/2$ vagy $s_i \leq 1$ vagy $s_{i+1} = 0$ // 19–21. sor: s elfogadása

20 $\qquad L = 1$

21 $\qquad$ **return** L

22 $\quad w_i = w_{i-1}$ // 22–24. sor: w_i frissítése

23 $\quad$ **while** $s_i < bi$ és $w_i > 0$

24 $\qquad\quad w_i = w_i - 1$

25 $\quad$ **if** $w_i \geq i$ // 25–27. sor: esetszétválasztás

26 $\qquad$ **if** $s_i > b(w_i - 1) + r_{i-1} + H_{w_{i-1}} - H_{w_i}$
$\qquad\quad -b(w_{i-1} - w_i)(i - 1)$ // 26. sor: s_i tesztelése

27 $\qquad\quad$ **return** L

28 $\qquad r_i = b(w_i - 1) + r_{i-1} + H_{w_{i-1}} - H_{w_i}$
$\qquad\quad -b(w_{i-1} - w_i)(i - 1) - s_i$ //28. sor: maradék frissítése

29 $\quad$ **else if** $s_i > bw_i + r_{i-1} + H_{w_{i-1}} - H_{w_i}$

n/i	1	2	3	4	5	6	7	8	9	10
1	0									
2	3	0								
3	22	3	0							
4	132	26	2	0						
5	824	164	31	4	0					
6	5 084	1 026	276	75	3	0				
7	31 902	6 288	2 018	829	111	5 0				
8	201 366	39 090	13 282	7 231	1 837	203	4	0		
9	1 281 918	244 833	84 340	53 594	20 681	4 259	298	6	0	
10	8 207 232	1 548 774	529 578	365 461	183 262	59 726	8 709	470	5	0
11	52 819 163	9 866 545	3 331 910	2 385 963	1 404 590	632 058	155 070	17 213	660	7

18. táblázat. ChHHl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt nem $(0, 2, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

30 $\quad\quad -b(w_{i-1} - w_i)(i - 1)$
31 $\quad$ **return** L
32 $\quad r_i = bw_i + r_{i-1} + H_{w_{i-1}} - H_{w_i}$
$\quad\quad -b(w_{i-1} - w_i)(i - 1) - s_i$ //32. sor: maradék frissítése
33 $L = 1$ // 33–34. sor: s elfogadása
34 **return** L

A következő állítás jellemzi ChHHl futási idejét.

32. TÉTEL. (Iványi, [28]) ChHHl futási ideje a legjobb $\Theta(1)$ és a legrosszabb $\Theta(n)$ között változik.

Bizonyítás. A 01–06. sorok végrehajtása $\Theta(1)$ időt igényel. Mivel ezek a sorok a nemgrafikus sorozatok jelentős részét kiszűrik, a legjobb futási idő $\Theta(1)$. A 07–11. sorok végrehajtása $\Theta(n)$ ideig tart. Mivel w szigorúan monoton csökken a program végrehajtása során, ezért a 12–24. sorok $O(n)$ időt igényelnek, így az algoritmus futási ideje minden esetben $\Theta(n)$. $\square$

Legyen $b = 3$ és $s = (13, 10, 5, 5, 4, 1)$. Az ötödik és tizedik sorok feltételei nem teljesülnek és $r_1 = 0$. Ha $i = 2$, akkor $w_i = 5$ és teljesül a 20. sor feltétele, így s *nem* $(0, 1, 6)$*-grafikus.*

A következő példában b maradjon 3, viszont s-et változtassuk meg: legyen $s' = (13, 10, 5, 5, 4, 3)$. Az előző esethez képest annyi a változás, hogy $r_1 = 2$ az első maradék, majd $i = 2$ esetén $w_i = 2$, nem teljesül a 20. sor feltétele és $r_2 = 0$. $i = 3$ esetén teljesül a 19. sor $H_{i-1} \geq H_n/2$ feltétele, ezért s' $(0, 1, 6)$*-grafikus.*

A következő példában legyen $b = 1$ és $s = (4, 3^3, 1)$. A 05. és 10. sorok feltételei nem teljesülnek és $r_1 = 0$. Ha $i = 2$, akkor $w_i = 4$ és nem teljesül a 20. sor feltétele, az $i = 3$ esetben pedig a 19. sorban teljesül a $H_{i-1} \geq H_n/2$ feltétel, azaz s $(0, 1, 5)$*grafikus.*

A 18. táblázat a ChHHl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt nem $(0, 2, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

A 19. táblázat a ChHHl i-edik ($i = 1, \ldots, 11$) menetében kiszűrt $(0, 2, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

A 20. táblázat a ChHHl algoritmus hatékonyságát jellemzi $(0, 2, n)$-szabályos sorozatok és $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10
1	1									
2	2	0								
3	1	9	0							
4	1	7	42	0						
5	1	10	29	224	0					
6	1	14	49	183	1 297	0				
7	1	18	70	345	1 143	7 658	0			
8	1	23	97	559	2 326	7 262	46 489	0		
9	1	28	125	846	4 038	15 927	46 074	286 007	0	
10	1	34	159	1 191	6 520	29 629	107 724	295 609	1 779 026	0
11	1	40	193	1 624	9 668	50 663	213 399	728 610	1 900 061	11 154 877

19. táblázat. ChHHl i-edik $(i = 1, \ldots, 11)$ menetében kiszűrt $(0, 2, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

$\frac{n}{\text{jellemző}}$	X	Y	Z	X'	Y'	Z'
2	$1,000000000$	$1,000000000$	$1,000000000$	$1,000000000$	$1,000000000$	$1,000000000$
3	$1,120000000$	$1,900000000$	$1,342857143$	$0,560000000$	$0,950000000$	$0,671428571$
4	$1,187500000$	$2,820000000$	$1,576190476$	$0,395833333$	$0,940000000$	$0,525396825$
5	$1,232649071$	$3,803030303$	$1,759906760$	$0,308162268$	$0,950757576$	$0,439976690$
6	$1,280785891$	$4,788212435$	$1,957042957$	$0,256157178$	$0,957642487$	$0,391408591$
7	$1,322698224$	$5,770438549$	$2,137870128$	$0,220449704$	$0,961739758$	$0,356311688$
8	$1,363989613$	$6,751572493$	$2,320248929$	$0,194855659$	$0,964510356$	$0,331464133$
9	$1,402468979$	$7,733105601$	$2,496464714$	$0,175308622$	$0,966638200$	$0,312058089$
10	$1,439464334$	$8,714770487$	$2,670148311$	$0,159940482$	$0,968307832$	$0,296683146$
11	$1,474743645$	$9,697001722$	$2,839981439$	$0,147474365$	$0,969700172$	$0,283998144$

20. táblázat. ChHHl hatékonysági jellemzői $a = 0$, $b = 2$ és $n = 1, \ldots, 11$ csúcs esetén.

9. (a, b, n)-gráfok

Chungphaisan tételének közvetlen következménye az alábbi állítás.

33. KÖVETKEZMÉNY. *Legyen $n \geq 2$. Az $s = (s_1, \ldots, s_n)$ (a, b, n)-szabályos sorozat akkor és csak akkor (a, b, n)-grafikus, ha az $s' = (s_1 - a(n-1), \ldots, s_n - a(n-1))$ sorozat $(0, b-a, n)$-grafikus.*

Bizonyítás. Egy (a, b, n)-gráfban minden csúcspár elemei legalább a éllel össze vannak kötve. Ezért ha minden csúcspár esetén eltávolítunk a élet, egy $(0, b-a, n)$-*gráfot* kapunk. $\square$

A 33. következmény szerint a következő három táblázat adatai megegyeznek a $(0, 3, n)$-szabályos sorozatokra vonatkozó hasonló adatokkal.

A 21. és 22. táblázatok a ChEGl i-edik – ahol $(i = 1, \ldots, 4)$, illetve $(i = 5, \ldots, 10)$ – menetében kiszűrt nem $(2, 5, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

A 23. táblázat a CL i-edik $(i = 1, \ldots, 10)$ menetében kiszűrt $(2, 5, n)$-grafikus sorozatok számát tartalmazza $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4
1	0	0	0	0
2	6	0	0	0
3	57	7	0	0
4	475	83	7	0
5	4099	732	163	13
6	35500	6287	2068	441
7	312188	53601	20775	7766
8	2769457	463794	188643	97976
9	24768128	4061297	1658351	1021804
10	222858957	35952854	14508359	9681500
11	2015400842	320927140	127636563	87804078

21. táblázat. ChEGl i-edik ($i = 1, \ldots, 4$) menetében kiszűrt, nem $(2, 5, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	5	6	7	8	9	10
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	14	0	0	0	0	0
7	921	21	0	0	0	0
8	24374	1921	23	0	0	0
9	405996	71152	3572	31	0	0
10	5136605	1554803	186666	6402	34	0
11	55159143	24279000	5343051	452411	10751	43

22. táblázat. ChEGl i-edik ($i = 5, \ldots, 10$) menetében kiszűrt, nem $(2, 5, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

n/i	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0	0
2	3	0	0	0	0	0	0	0	0	0
3	1	19	0	0	0	0	0	0	0	0
4	1	8	141	0	0	0	0	0	0	0
5	1	11	40	1129	0	0	0	0	0	0
6	1	15	60	317	9561	0	0	0	0	0
7	1	19	81	497	2395	82435	0	0	0	0
8	1	24	108	720	3838	19074	722192	0	0	0
9	1	29	136	1016	5733	30725	153657	6385472	0	0
10	1	35	170	1366	8387	47136	247112	1259718	56880031	0
11	1	41	204	1804	11644	70961	385774	2010389	10453559	509514569

23. táblázat. ChEGl i-edik ($i = 1, \ldots, 10$) menetében kiszűrt $(2, 5, n)$-grafikus sorozatok száma $n = 1, \ldots, 11$ csúcs esetén.

A következő 24. táblázat a ChEGl algoritmus hatékonyságát jellemzi $a = 2$, $b = 5$ és $n = 1, \ldots, 11$ csúcs esetén.

$\frac{n}{\text{jellemző}}$	X	Y	Z	X'	Y'	Z'
2	1,000000000	1,000000000	1,000000000	1,000000000	1,000000000	1,000000000
3	1,109375000	1,950000000	1,309523810	0,554687500	0,975000000	0,654761905
4	1,171681416	2,933333333	1,541258741	0,390560472	0,977777778	0,513752914
5	1,219093269	3,944961897	1,739334195	0,304773317	0,986240474	0,434833549
6	1,266350711	4,951175407	1,942282176	0,253270142	0,990235081	0,388456435
7	1,309250339	5,956536499	2,135146661	0,218208390	0,992756083	0,355857777
8	1,350304891	6,960496382	2,325332905	0,192900699	0,994356626	0,332190415
9	1,389017669	7,963928944	2,510223895	0,173627209	0,995491118	0,313777987
10	1,426027860	8,966857120	2,691252565	0,158447540	0,996317458	0,299028063
11	1,461490194	9,969401198	2,868359205	0,146149019	0,996940120	0,286835921

24. táblázat. ChEGl hatékonysági jellemzői $a = 2$, $b = 5$ és $n = 1, \ldots, 11$ csúcs esetén.

10. $(0, 1, n)$-grafikus sorozatok párhuzamos leszámlálása

A 8. táblázat 1-től 29 csúcsig tartalmazza a grafikus sorozatok számát. A táblázat úgy készült, hogy párhuzamosítottuk az Erdős-Gallai-gyorsan algoritmust. Az eredmény a Erdős-Gallai-leszámláló algoritmus, amely minden szóbajövő sorozatot tesztel. Az algoritmus részletes leírása megtalálható a [29] cikkben.

Az algoritmus viszonylag gyors futása érdekében igyekeztünk minél kisebbre választani a vizsgálandó sorozatok halmazát. Ezért csak a nullamentes páros sorozatokat teszteltük, hiszen egyrészt a páratlanok nem lehetnek grafikusak, másrészt a $G(n)$ függvény értékeit a 21. lemma segítségével könnyen kiszámíthatjuk, ha ismerjük a $G_z(n)$ függvény megfelelő értékeit.

Mivel 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.

A Erdős-Gallai-lineáris algoritmus egyik lehetséges alkalmazása, hogy meghatározzuk a grafikus sorozatok számát olyan n értékekre, amelyekre eddig a nagy számolásigény miatt nem volt ismert: Sloane *The On-Line Encyclopedia of Integer Sequences* című honlapja [63] az $n = 23$ értékig tartalmazta a grafikus sorozatok számát. Ezt kiegészítettük $n = 29$ csúcsig [65].

A alábbi Erdős-Gallai-leszámláló (EGl) algoritmus a lineáris legrosszabb eset mellett azt is igyekszik kihasználni, hogy ha lexikografikus sorrendben ellenőrizzük a szóba jövő sorozatokat, akkor a szomszédos sorozatok bizonyos tulajdonságai nagyon hasonlóak, ezért adott sorozat jellemzői az őt megelőző sorozat jellemző adataiból konstans várható 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 $(0, 1, n)$-grafikus sorozatok között ugyanis a 21. lemma szerint pontosan $G(n-1)$ nullamentes grafikus 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 ugró pontokban vizsgáljuk.

Az EGl program azt is kihasználja, hogy a szomszédos sorozatok ellenőrző pontjainak a listája átlagosan konstans idő alatt származtatható a megelőző sorozat adataiból. A kiindulási értékek szintén könnyen számíthatók: az első – $q = (n-1)^n$ – sorozatra a C lista üres (azaz egyáltalán nem kell ellenőrzést végeznünk), a súlypontok listája pedig kezdetben $w = \left((n-1)^{n-1}\right)$.

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

Az ugrópontok $C(s')$ listája rendszerint megegyezik a $C(s)$ listával, és legfeljebb a végén változik egy vagy két elem.

Mivel a futási idő csökkentése érdekében az Erdős-Gallai-leszámláló algoritmus csak nullamentes sorozatokat állít elő és tesztel, a szeletekre bontás alapja az (34) képlet.

Feltételezzük, hogy a $(0, n-1, n)$-szabályos nullamentes sorozatok halmazának szeletekre való felbontásánál az egyes szeletek futási ideje arányos a hozzájuk tartozó $R(1, n-1, n)$-szabályos sorozatok számával.

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.

?????????????

11. Nyitott problémák

Az alábbiakban néhány nyitott problémát, további feladatot mutatunk be.

11.1. Különböző gráfosztályok foksorozatainak leszámlálása

Explicit képletek nem ismertek ismertek a különböző (a, b, n)-gráfok foksorozatainak számára. Ugyanakkor intenzív kutatások folynak olyan gyors módszerek kidolgozásával kapcsolatban, amelyek

Kleitman [36] aszimptotikus korlátokat adott az irányított $(1, 1, n)$-gráfok számára, míg Burns [11] az irányítatlan $(0,1,n)$-gráfokéra. Narayana és Bent [49] rekurzív képleteket javasoltak az irányított $(0,1,n)$-gráfok pontos számának meghatározására. Pécsy és Szűcs [52] Narayana és Bent képleteit felhasználva gyors párhuzamos algoritmussal számították ki a $D(0,1,n)$ függvény értékeit. Iványi et al. [30] 2011-ben ugyancsak párhuzamos algoritmussal határozták meg $G(n)$ értékeit $n = 29$-ig.

11.2. Konkrét problémák bonyolultsága

Ebben a cikkben megmutattuk, hogy nem csak az egyszerű gráfok, hanem még az(a,b,n)-gráfok potenciális foksorozatai is ellenőrizhetők polinomiális, méghozzá lineáris idő alatt.

A hiányos gráfok esetén nehezebbek a feladatok. Például a labdarúgást modellező gráfok esetén nyitott kérdés, hogy egy $(2,3,n)$-szabályos sorozatról eldönthető-e polinomiális idő alatt, hogy megfelel a labdarúgás szabályainak [20].

11.3. További gráfosztályok potenciális foksorozatainak tesztelése

Páros gráfok, többrészes gráfok, hipergráfok

Köszönetnyilvánítás. A szerzők köszönik az ismeretlen lektor jobbító észrevételeit. 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. ⇒ 20

[2] Avis, D., Fukuda, K.: *Reverse search for enumeration.* Discrete Appl. Math. **2,** (1993) 21–46. ⇒ 18

[3] Barnes, T. M., Savage, C. D.: *A recurrence for counting graphical partitions.* Electron. J. Combin. **2,** (1995) R11, 10 pp. ⇒ 18, 22

[4] Barnes, T. M., Savage, C. D.: *Efficient generation of graphical partitions. Discrete Appl. Math.* **78(1–3),** (1997) 17–26. ⇒ 2, 18, 22

[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] Berger, A., Müller-Hannemann, M.: *Uniform sampling of digraphs with a fixed degree sequence.* In: (ed. D. M. Thilikos) WG2010, LNCS **6410** (2010), pp. 220–231. ⇒ 2

[8] 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

[9] Bozóki S., Fülöp J., Rónyai, L.: *On optimal completion of incomplete pairwise comparison matrices.* Math. Comput. Modelling **52,** (2010) 318–333. ⇒1, 2

[10] Brualdi, A. R., Kiernan K.: *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. ⇒2, 18, 22, 23, 45

[12] 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

[13] 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, 19

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

[15] Chungphaisan, V.: *Conditions for sequences to be r-graphical.* Discrete Math. **7,** (1974) 31–39. ⇒1, 35, 38, 50

[16] 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

[17] Erdős, P., Gallai, T.: *Gráfok előírt fokú pontokkal.* Mat. Lapok **11,** (1960) 264–274. ⇒1, 4, 50

[18] 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, 4

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

[20] Frank, A.: *Connections in Combinatorial Optimization.* Oxford University Press, Oxford, 2011. ⇒2, 45

[21] Frank, D. A., Savage, C. D., Sellers, J. A.: *On the number of graphical forest partitions.* *Ars Combin.* **65,** (2002) 33–37. ⇒18

[22] 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, 50

[23] Havel, V.: A remark on the existence of finite graphs (cseh); *Časopis Pěst. Mat.* **80,** (1955), 477–480. ⇒1, 3, 50

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

[25] Iványi, A.: *Reconstruction of complete interval tournaments.* Acta Univ. Sapientiae, Inform., **1(1),** (2009) 71-88. ⇒1, 2, 4

[26] Iványi, A.: *Reconstruction of complete interval tournaments. II.* Acta Univ. Sapientiae, Math., **2(1),** (2010) 47–71. ⇒1, 2, 4

[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.: *Degree sequences of multigraphs.* Annales Univ. Budapest., Computatorica (benyújtva). ⇒35, 36, 38, 40

[29] Iványi, A., Lucz, L.: *Parallel Erdős-Gallai theorem.* Central Eur. J. Oper. Res. (benyújtva). ⇒43

[30] Iványi, A., Lucz, L., Móri F. T., Sótér, P.: *On the Erdős-Gallai and Havel-Hakimi algorithms.* Acta Univ. Sapientiae, Inform. **3(2),** (2011) 230–268. ⇒1, 2, 10, 14, 20, 21, 22, 45

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

[32] 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

[33] Kéri G.: *On qualitatively consistent, transitive and contradictory judgment matrices emerging from multiattribute decision procedures.* Central Eur. J. Oper. Res. **19(2)** (2011) 215–224. ⇒1

[34] 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

[35] Kleitman, D. J., Wang, D. L.: *Algorithms for constructing graphs and digraphs with given valencies and factors.* Discrete Math. **6** (1973) 79–88. ⇒4

[36] Kleitman, D. J., Winston K. J.: *Forests and score vectors.* Combinatorica **1(1),** (1981) 49–54. ⇒18, 44

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

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

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

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

[41] 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

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

[43] 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

[44] Lucz, L., Sótér, P.: *Foksorozatokat ellenőrző algoritmusok.* TDK dolgozat. ELTE IK, Budapest, 2011. ⇒24

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

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

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

[48] Moon, J. W.: *Topics on Tournaments.* Holt, Rinehart, and Winston, New York, 1968. ⇒18

[49] Narayana, T. V., Bent, D. H.: Computation of the number of score sequences in round-robin tournaments. Canad. Math. Bull. **7(1),** (1964) 133–136. ⇒45

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

[51] Özkan, S.: *Generalization of the Erdős-Gallai inequality.* Ars Combin. **98,** (2011) 295–302. ⇒1, 2

[52] Pécsy G., Szűcs, L.: *Parallel verification and enumeration of tournaments.* Stud. Univ. Babeş-Bolyai, Inform. **45(2),** (200) 11–26. ⇒33, 45

[53] Pirzada, S.: *Graph Theory.* Orient Blackswan, 2012, to appear. ⇒2

[54] 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

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

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

[57] 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, 18

[58] Ruskey, F., Cohen, R., Eades, P., Scott, A.: *Alley CAT's in search of good homes.* Congr. Num., **102,** (1994) 97–110. ⇒2, 18, 34

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

[60] Siklósi, B.: *Soros és párhuzamos algoritmusok összehasonlítása sportversenyekkel kapcsolatos problémákban.* Programtervező matematikus diplomamunka. ELTE TTK, Budapest, 2001. 69 oldal. ⇒33

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

[62] Sloane N. J. A., Plouffe S.: *The Encyclopedia of Integer Sequences.* Academic Press, 1995. ⇒18, 20

[63] Sloane N. J. A. (szerk.): *Encyclopedia of Integer Sequences.* 2011. ⇒ 2, 43

[64] 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. ⇒ 18, 33

[65] 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. http://oeis.org/A004251 . ⇒ 18, 33, 43

[66] 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. ⇒ 18, 33

[67] Soroker, D.: *Optimal parallel construction of prescribed tournaments.* Discrete Appl. Math. **29(1),** (1990) 113–125. ⇒ 33

[68] Stanley, R.: *Enumerative Combinatorics. Vol. 2.* Cambridge University Press, Cambridge, 1997. ⇒ 18

[69] 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. ⇒ 18

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

[71] Tripathi, A., Vijay, S.: *A note on a theorem of Erdős & Gallai.* Discrete Math. **265(1–3),** (2003) 417–420. ⇒ 7, 8, 11, 12

[72] 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, 5, 50

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

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

[75] Winston, K. J., Kleitman, D. J.: *On the asymptotic number of tournament score sequences.* J. Combin. Theory Ser. A. **35,** (1983) 208–230. ⇒ 18

Beérkezett:

IVÁNYI ANTAL
tony@compalg.inf.elte.hu
Eötvös Loránd Tudományegyetem
Informatikai Kar, Komputeralgebra Tanszék
1117 Budapest, Pázmány Péter sétány 1/C

LUCZ LORÁND
lorand.lucz@gmail.com
Eötvös Loránd Tudományegyetem
Informatikai Kar, Komputeralgebra Tanszék
1117 Budapest, Pázmány Péter sétány 1/C

DEGREE SEQUENCES OF MULTIGRAPHS

ANTAL IVÁNYI, LORÁND LUCZ

Let a, b and n integers, $0 \leq a < b$ and $n \geq 1$. (a, b, n)-graphs are loopless multigraphs in which any two vertices are connected with an least a and at most b edges. Havel in 1955 [23], Erdős and Gallai in 1960 [17], Hakimi in 1962 [22], Tripathi, Venugopalan and West in 2010 [72] proposed a method to decide, whether a sequence of nonnegative integers can be the degree sequence of a $(0, 1, n)$-graph. Chungphaisan in 1974 [15] extended Havel-Hakimi and Erdős-Gallai theorem for $(0, b, n)$-graphs. All the mentioned algorithms require at least quadratic time in worst case. We extend Erdős-Gallai-Chungphaisan theorem for (a, b, n)-graphs and propose a linear time algorithm, based on our theorem. We also propose a linear time version of the testing Havel-Hakimi algorithm and extend it for $(0, 2, n)$-graphs.