LINEÁRIS ERDŐS-GALLAI TESZT

IVÁNYI ANTAL, LUCZ LORÁND, MÓRI F. TAMÁS

Havel 1955-ben [29], Erdős és Gallai 1960-ban [21], Hakimi 1962-ben [26], Ruskey, Cohen, Eades és Scott 1994-ben [75], Barnes and Savage 1997-ben [6], Tripathi, Venugopalan és West 2010-ben [91] javasoltak módszert annak eldöntésére, hogy nemnegatív egészek sorozata lehet-e egy egyszerű gráf foksorozata. Ezeknek az algoritmusoknak a futási ideje legrosszabb esetben négyzetes. Cikkünkben bemutatjuk az Erdős-Gallai-Lineárisan algoritmust, amelynek legrosszabb futási ideje lineáris. Mivel például jó sorozat esetén a sorozat minden elemét tesztelni kell, RAM számítási modell esetén EGL aszimptotikusan optimális. Az új algoritmus segítségével 24, 25, 26, 27,28 és 29 csúcs esetén meghatároztuk egyszerű gráfok különböző foksorozatainak a számát [81].

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 [55], Hakimi kémiai [26], Kim, Toroczkai, Miklós, Erdős és Székely [48] és Newman és Barabási [66] hálózati, Bozóki, Fülöp, Poesz és Rónyai gazdasági [1, 10, 11, 45, 88] Liljeros et al. emberi kapcsolatokra vonatkozó [56], Iványi et al. pedig sportbeli [32, 33, 36, 39, 43, 71, 73] alkalmazásokra hivatkoztak.

A számos lehetséges terminológia közül Erdős Pál és Gallai Tibor [21] cikkének szóhasználatát és jelöléseit követjük.

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

A sok népszerű feladat közül azt vizsgáljuk, hogyan dönthető el gyorsan, hogy nemnegatív egészek $n \geq 1$ hosszúságú $b = (b_1, \ldots, b_n)$ sorozatához létezik-e olyan egyszerű gráf, amelynek b a foksorozata.

A hasonló feladatokkal kapcsolatban megjegyezzük, hogy mind az irányítatlan, mind pedig az irányított gráfokkal kapcsolatban az utóbbi néhány évben is számos cikk és könyvfejezet jelent meg (például [8, 12, 19, 30, 43, 59, 74, 89, 91, 92, 93], illetve [7, 11, 14, 22, 24, 32, 33, 35, 39, 40, 48, 51, 54, 62, 68, 71, 72]).

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

Jelentős számú cikk (például [13, 23, 52, 60]) foglalkozik páros számok *grafikus felbontásaival:* előállítják a $2k$ páros szám pozitív egész öszeadandókra való monoton csökkenő felbontásait, és az így kapott $q = (q_1, \ldots, q_m)$ sorozatok közül – amelyekre $q_1 + \cdots + q_m = 2k$ és $q_m \geq q_{m-1} \geq \cdots \geq q_1$ – szűrik ki a grafikus sorozatokat, vagy pedig rekurzióval eleve csak a grafikus sorozatokat állítják elő.

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

Cikkünk fő célkitűzése, hogy minél kisebb várható futási idejű algoritmust találjunk annak eldöntésére, hogy előírt b szabályos sorozathoz található-e olyan G egyszerű gráf, melynek foksorozata b. A probléma önmagában is érdekes és fontos, hiszen a gráfelmélet több mint ötven éve intenzíven foglalkozik vele. Számunkra külön motivációt jelent, hogy a feladat részfeladatként jelentkezik, amikor minél kisebb várható futási idejű algoritmust keresünk annak eldöntésére, hogy előírt sorozat lehet-e egy labdarúgó bajnokság pontsorozata [24, 31, 34, 38, 40, 53, ?]. A labdarúgás is olyan sport, amelyben a döntetlen megengedett eredmény. Vizsgálatainkban mind a tesztelés, mind pedig a megvalósítás során kulcsszerepet játszik a döntetlenek párosítása – ez pedig pontosan az egyszerű gráfok fokainak párosításával ekvivalens feladat.

Az egyszerű irányítatlan gráfok foksorozataival kapcsolatos eredmények és módszerek hasznosak az irányított gráfokra, multigráfokra [27, 32, 33, 44, 54], többrészes gráfokra [69, ?] és a foksorozatokhoz hasonló sorozatokra [69, 72, 73] vonatkozó hasonló feladatok megoldásánál.

Érdemes megemlíteni, hogy a fokszámsorozatok számának meghatározásával kapcsolatos nehézségek miatt annak is van irodalma (lásd például [19, 43, 62]), hogy véletlen mintavétellel becsüljük ezeket a számokat.

Melléktermékként javítottuk és bővítettük az EIS [80] adatbázist: egyrészt a benne lévő ismert sorozatok új elemeivel, másrészt a cikkben definiált új sorozatok elemeinek megadásával. Módszerünk az összes jó sorozat gazdaságos előállítására is alkalmas (vesd össze Ruskey [75] 1994-es, valamint Barnes and Savage [6] 1997-es cikkével).

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

2. Klasszikus pontos algoritmusok

2.1. Havel-Hakimi algoritmus

A feladat megoldására az első módszert Vaclav Havel cseh matematikus javasolta 1955-ben [29, 57], majd 1962-ben Louis Hakimi [26] tőle függetlenül publikálta ugyanezt az eredményt, innen adódik a Havel-Hakimi elnevezés.

1. Tétel. (Havel [29]) *Ha* $n \geq 1$, *a* $(b_1, \dots, b_n)$ *n-szabályos sorozat akkor és csak akkor n-jó, ha a* $(b_2 - 1, b_3 - 1, \dots, b_{b_1} - 1, b_{b_1+1} - 1, b_{b_1+2}, \dots, b_n)$ *sorozat* $(n-1, n-1)$*-jó.*

Bizonyítás.: Lásd [29]. □

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

Ha ezen tétel alapján írunk egy rekurzív algoritmust, akkor annak futási ideje a RAM számítási modell [17] szerint legjobb esetben – például az $(n-1, 0, \dots, 0) = (n-1, 0^{n-1})$ bemenetre – $\Theta(1)$, legrosszabb esetben pedig – például a *homogén* $(n-1)^n$ vagy a *tranzitív* $(n-1, n-2, \dots, 1, 0)$ bemenetre – $\Theta(n^2)$. Érdemes megjegyezni, hogy a tétel bizonyítása konstruktív, és a bizonyításon alapuló algoritmus négyzetes idő alatt nem csak ellenőriz, hanem egy megfelelő gráfot is előállít (feltéve persze, hogy létezik megfelelő egyszerű gráf).

Mivel a teljes gráfnak $\binom{n}{2} = \Theta(n^2)$ éle van, bármely – a megvalósítást is biztosító – RAM algoritmus legrosszabb futási ideje nem lehet kisebb, mint négyzetes.

1962-ben Louis Hakimi [26] Haveltől függetlenül publikálta ugyanezt az eredményt, ezért ma a tételt rendszerint *Havel-Hakimi tételnek*, a módszert pedig *Havel-Hakimi algoritmusnak* nevezik.

Az algoritmust később irányított gráfokra [22, 32, 33, 49] is kiterjesztették.

2.2. Erdős-Gallai algoritmus

Időrendben a következő eredmény Erdős Pál és Gallai Tibor alábbi szükséges és elégséges feltétele [21] volt.

Nemnegatív egészek adott $b = (b_1, \dots, b_n)$ sorozata esetén a sorozat első i elemét a sorozat i elemhez tartozó *fejének*, míg a többi elemét a i elemhez tartozó *farkának* nevezzük. A fejelemek összegét H_i, míg a farokelemek összegét T_i jelöli $(i = 1, \dots, n)$.

A $\sum_{k=i+1}^{n} \min(i, b_k)$ összeget pedig C_i-vel jelöljük és a farok *kapacitásának* nevezzük. Ha egy b sorozatra H_n páros, akkor a sorozatot *párosnak,* egyébként *páratlannak* nevezzük.

2. Tétel. (Erdős, Gallai, [21]) *Ha $n \geq 1$, az n-szabályos $(b_1 \ldots, b_n)$ sorozat akkor és csak akkor jó, ha*

$$H_n \quad \textit{páros} \tag{1}$$

és

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

Bizonyítás. Lásd [15, 21, 76, 91]. □

Bár ez a tétel csak ellenőriz, futási ideje a RAM számítási modell szerint a legjobb esetben $\Theta(n)$ és a legrosszabb esetben $\Theta(n^2)$ között változik. A közelmúltban Tripathi et al. [91] publikálták a tételre konstruktív bizonyítást, amely jó bemenet esetén $O(n^3)$ idő alatt egy megoldást is előállít. Az 1. ábrán láthatjuk az n-szabályos (R(n)) és n-páros (E(n)) sorozatok számát és ezek egymáshoz viszonyított arányát (E(n)/R(n)) n=1,...,38. Ha n tart a végtelenbe, akkor ez a hányados az 1/2-hez tart (26).

3. Új pontos algoritmusok

Ebben a részben a nullamentes algoritmusok, az eltoló Havel-Hakimi, a paritásos Havel-Hakimi, a rövidített Erdős-Gallai, ugró Erdős-Gallai, lineáris Erdős-Gallai és a gyors Erdős-Gallai algoritmusokat mutatjuk be.

3.1. Nullamentes algoritmusok

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

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

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

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

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

3.2. Eltoló Havel-Hakimi algoritmus

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

A tétel alapján olyan megvalósítás is lehetséges, hogy a fokszámok redukálását a sorozat monotonitását megőrizve végezzük. Ekkor az Eltoló Havel-Hakimi algoritmust (HHE) kapjuk. A módszer tovább gyorsítható, ha az első nem kielégíthető csúcsnál elutasítjuk a sorozatot és nem végzünk el további felesleges számításokat.

Bemenet: n a csúcsok száma ($n \geq 1$); $b = (b_1, b_2, \ldots, b_n)$ n-szabályos sorozat
Kimenet: L logikai változó
(L = False azt jelzi, hogy b biztosan nem jó, míg L = True azt mutatja, hogy b jó)

```
Eltoló-Havel-Hakimi(n, b, L)
01 FF = b                                  ▷ 01. sor: munkaváltozó inicializálása
02 first = 0                               ▷ 02. sor: ellenőrizendő elem indexének beállítása
03 while FF_first > 0                      ▷ 03–33. sor: sorozat ellenőrzése
04     last = FF_first + first             ▷ 04. sor: utoljára megvizsgálandó elem
05     if FF_last == 0                     ▷ 05–08. sor: gyors teszt
06         L = false                       ▷ 06–07. sor: rossz sorozat elutasítása
07         return
08     end
09     if FF_last == FF_last+1             ▷ 09–30. sor: ellenőrzése 'eltolással'
10         i = first + 1                   ▷ 06. sor: első egyforma elem index
11         if FF_last ≠ FF_i               ▷ 11–15. sor: első egyforma elem keresése
12             while FF_i ≠ FF_last
13                 i = i + 1
14             end
15         end
16         for j = first + 1 to i          ▷ 16–18. sor: eltolás mentes rész csökkentése
17             FF_j = FF_j − 1
18         end
19         i = last − i + 1                ▷ 19. sor: leköthető fokszámigény
20         j = last + 2                    ▷ 20. sor: j inicializálása
21         while FF_last = FF_j            ▷ 21–23 sor: utolsó ugyanolyan elem megkeresése
22             j = j + 1
23         end
24         for ii = 0 to i                 ▷ 24–27. sor: fokszámok csökkentése hátulról haladva
25             j = j − 1
26             FF_j = FF_j − 1
27         end
28         FF_first = 0                    ▷ 28. sor: a kielégített első elem nullázása
29         first = first + 1               ▷ 29. sor: új első elem beállítása
30     else                                ▷ 30–36. sor: ellenőrzés 'eltolás' nélkül
31         for j = first + 1 to last + 1   ▷ 31–33. sor: fejelem igényének kielégítése
32             FF_j = FF_j − 1
```

```
33          end
34          FF_first = 0                  ▷ 34. sor: a kielégített első elem nullázása
35          first = first + 1             ▷ 35. sor: új első elem beállítása
36      end
37 end
38 if FF_n == 0                          ▷ 38. sor: ha az utolsó igényt is sikeresen kielégítettük
39      L = true                          ▷ 39. sor: elfogadó állapot beállítása
40 else
41      L = false                         ▷ 41. sor: elutasító állapot beállítása
42 end
43 return L                              ▷ 43. sor: visszatérés az eredménnyel
```

3.3. Paritásos Havel-Hakimi algoritmus

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

Bemenet: n a csúcsok száma ($n \geq 1$); $b = (b_1, b_2, \ldots, b_n)$ n-szabályos sorozat
Kimenet: L logikai változó
($L =$ False azt jelzi, hogy b biztosan nem jó, míg $L =$ True azt mutatja, hogy b jó)
Munkaváltozók:
$H = (H_1, \ldots, H_n)$: H_i b első i elemének az összege

Havel-Hakimi(n, b, L)

```
01 H_1 = b_1                             ▷ 01. sor: H_1 számítása
02 for i = 2 to n                        ▷ 02–04. sor: H értékek számítása
03      H_i = H_{i-1} + b_i
04 end
05 if H_n páratlan                       ▷ 05–08. sor: paritás ellenőrzés
06      L = false                         ▷ 06–07. sor: hibás sorozat elutasítása
07      return L
08 end
09 FF = b                                ▷ 09. sor: munkaváltozó inicializálása
10 while FF_0 > 0                        ▷ 10. sor: amíg nem értünk a sorozat végére
11      for j = 1 to FF_0 + 1            ▷ 11–13. sor: kielégítjük az első fok igényét
12          FF_j = FF_j − 1
13      end
14      FF_0 = 0                          ▷ 14. sor: nullázzuk az első elemet
```

```
15     sort_descending(FF)           ▷ 15. sor: újrarendezzük a sort
16 end
17 t = 0                             ▷ 09. sor: ideiglenes változó nullázása
18 for i = 0 to n                    ▷ 18–20. sor: munkaváltozó értékeinek összegzése
19     t = t + FF_i
20 end
21 if t == 0                         ▷ 21–25. sor: helyreállíthatóság eldöntése
22     L = true                      ▷ 22. sor: helyreállítható
23 else
24     L = false                     ▷ 24. sor: nem helyreállítható
25 end
26 return L
```

3.4. Rövidített Erdős-Gallai algoritmus

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

Ennél is hasznosabb észrevételt tartalmaz a következő lemma.

Tripathi és Vijay 2003-as cikkében [90] 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 [90]) *Ha $n \geq 1$, az n-szabályos $b = (b_1 \ldots, b_n)$ sorozat akkor és csak akkor n-jó, ha*

$$H_n \quad \textit{páros} \tag{3}$$

és

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

ahol

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

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

Például a $b = (5^{100})$ sorozat esetén (2) jobb oldalát az Erdős-Gallai algoritmus szerint 99-szer, míg a rövidített Erdős-Gallai algoritmus szerint csak 6-szor kell kiszámítani. A javításnak a várható futási időre gyakorolt hatását a 7. részben vizsgáljuk.

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

3.5. Ugró Erdős-Gallai algoritmus

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

A b_i elemet a b sorozat *ugró* elemének nevezzük, ha $i = n$ vagy $1 \leq i \leq n-1$ és $b_i > b_{i+1}$. Ekkor az ugró elemek a $b_{\sigma_1}, b_{\sigma_2}, \ldots, b_{\sigma_q}$ elemek.

5. Tétel. (Tripathi, Vijay [90]) *Az n-szabályos $b = (b_1 \ldots, b_n)$ sorozat akkor és csak akkor n-jó, ha*

$$H_n \quad \textit{páros} \tag{6}$$

és

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

Bizonyítás. Lásd [90]. □

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

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

Bemenet: n a csúcsok száma $(n \geq 1)$; $b = (b_1, b_2, \ldots, b_n)$ n-szabályos sorozat
Kimenet: L logikai változó
($L =$ False azt jelzi, hogy b biztosan nem jó, míg $L =$ True azt mutatja, hogy b jó)
Munkaváltozók:
$H = (H_1, \ldots, H_n)$: H_i b első i elemének az összege

Erdős-Gallai-Ugró(n, b, L)

01 $H_1 = b_1$ ▷ 01. sor: H_1 számítása
02 **for** $i = 2$ **to** n ▷ 02–04. sor: H értékek számítása
03 $\quad$ $H_i = H_{i-1} + b_i$
04 **end**
05 **if** mod $(H_n, 2)$ ▷ 05–08. sor: paritás ellenőrzés
06 $\quad$ $L =$ **false** ▷ 06–07. sor: hibás sorozat elutasítása
07 $\quad$ **return** L
08 **end**
09 **for** $j = 1$ **to** $n - 1$ ▷ 09–23. sor: sorozat ellenőrzése
10 $\quad$ **if** $b_j = b_{j+1}$ ▷ 10–12. sor: eldöntjük, hogy az adott pont ellenőrzőpont-e
11 $\quad\quad$ **continue**
12 $\quad$ **end**

```
13        R = 0                                   ▷ 13. sor: munkaváltozó nullázása
14        for k = j + 1 to n                      ▷ 14–16. sor: a tétel szummájának számítása
15        R = R + min(j, b_k)
16        end
17        if H_j − j · (j − 1) > R                ▷ 17. sor: a tétel feltételének ellenőrzése
18              L = false                         ▷ 18–19. sor: rossz sorozat elutasítása
19              return L
20        end
21  end
22  L = true                                      ▷ 22-23. sor: jó sorozat elfogadása
23  return L
```

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

3.6. Lineáris Erdős-Gallai algoritmus

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

Adott b sorozat esetén legyen $w(b) = (w_1, \ldots, w_{n-1})$, ahol w_i a b sorozat legnagyobb indexű olyan elemére mutat, ameyik legalább akkora, mint i.

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

$$H_n \quad \textit{páros}, \tag{8}$$

és ha $i > w_i$, akkor

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

különben ha i w_i, akkor

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

Bizonyítás. A b sorozat b_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 b_j elemének hozzájárulása csak b_j.

- ha $i \leq w_1$, akkor a farok hozzájárulása két esetet tartalmaz: $C_{i+1}, \ldots, C_{m_i}$ egyenlő i, while $C_j = b_j$ for $j = m_i + 1, \ldots, n$.

Ezért ha $n-1 \geq i > w_i$, akkor

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

ha $1 \leq i \leq w_i$, akkor

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

□

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

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

Kimenet: L logikai változó

(melynek értéke TRUE, ha a bemenet jó, és FALSE egyébként)

Munkaváltozók

$H = (H_1, \ldots, H_n)$: H_i az éppen tesztelt b első i elemének az összege

$m = (m_1, \ldots, m_{n-1})$: m_i a legnagyobb indexű olyan b_j indexe, amely legalább i

$H_0 = 0$: segédváltozó a H sorozat elemeinek kiszámításához;

$b_0 = n-1$: segédváltozó az m sorozat elemeinek kiszámításához;

$w = (w_1, \ldots, w_{n-1})$: w_i a b_i súlypontja, a legnagyobb eleme a b-nek ami nem kisebb mint i

Erdős-Gallai-Lineárisan(n, b, L)

01 $H_0 = 0$ ▷ 01. sor: értékadás
02 **for** $i = 1$ **to** n ▷ 02–03. sor: H számítása
03 $\quad H_i = H_{i-1} + b_i$
04 **if** H_n páratlan ▷ 04–06. sor: paritás tesztelése
05 $\quad L =$ False
06 $\quad$ **return** L
07 $b_0 = n - 1$ ▷ 07. sor: értékadás a segédváltozónak
08 **for** $i = 1$ **to** n ▷ 08–12. sor: súlypont kiszámítása
09 $\quad$ **if** $b_i < b_{i-1}$
10 $\qquad$ **for** $j = b_{i-1}$ **downto** $b_i + 1$
11 $\qquad\quad w_j = i - 1$
12 $\qquad w_{b_i} = i$
13 **for** $j = b_n$ **downto** 1 ▷ 13–14. sor: large weights
14 $\quad w_j = n$
15 **for** $i = 1$ **to** n ▷ 15–23. sor: a b elemeinek tesztelése
16 $\quad$ **if** $i \leq w_i$ ▷ 16–19. sor: test of indices for large w_i's
17 $\qquad$ **if** $H_i > i(i-1) + i(w_i - i) + H_n - H_{w_i}$
18 $\qquad\quad L =$ False
19 $\qquad\quad$ **return** L
20 $\quad$ **if** $i > w_i$ ▷ 20–23. sor: test of indices for small w_i's
21 $\qquad$ **if** $H_i > i(i-1) + H_n - H_i$
22 $\qquad\quad L =$ False
23 $\qquad\quad$ **return** L
24 $L =$ True ▷ 24–25. sor: a program visszatérési értéke True
25 $\quad$ **return** L

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

Bizonyítás. Az első sor $O(1)$, 2–3. sor $\Theta(n)$, 4–6. sor $O(1)$, 07. sor $O(1)$, 08–12. sor $O(1)$, 13–14. sor $O(n)$, 15–23. sor $O(n)$ és 24–25. sor $O(1)$ időben végezhető el, ezért a szükséges futási idő $\Theta(n)$. □

Mivel a jó sorozatok minden elemét megvizsgáljuk a teszt során, ezért a RAM számítási modell alapján a [17] Erdős-Gallai-Linear aszimptotikusan optimális.

3.7. Gyors Erdős-Gallai algoritmus

A Erdős-Gallai-Lineárisan algoritmus egyik lehetséges alkalmazása, hogy meghatározzuk a jó sorozatok számát olyan n értékekre, amelyekre eddig a nagy számolásigény miatt nem volt ismert: Sloane *On-line Encyclopedia of Integer Sequences* című honlapja [80] az $n = 23$ értékig tartalmazza a jó sorozatok számát.

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

Igyekszünk az ellenőrizendő sorozatok számát is csökkenteni.

Ennek egy egyszerű megoldása, hogy eleve csak a páros sorozatokat állítjuk elő. További ötlet, hogy csak a nullamentes sorozatokat vizsgáljuk. A nullát tartalmazó n-jó sorozatok között ugyanis a 14. lemma szerint pontosan $G(n-1)$ jó sorozat van. Igaz, hogy aszimptotikusan a nullát tartalmazó sorozatok a páros sorozatok elhanyagolható részét adják, de az általunk most gyakorlatilag vizsgált $n \in [4, 30]$ tartományban még jelentős a részarányuk.

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

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

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

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

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

Kimenet. G: az n-jó sorozatok száma.

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

változó.

```
Erdős-Gallai-Gyorsan(n, G)
01 H_0 = c = 0                                ▷ 01–03. sor: kezdeti értékek beállítása
02 G = 1
03 ???
04 for i = 1 to n                             ▷ 04–05. sor: első vizsgált sorozat előállítása
05      b_i = n − 1
06      H_i = H_i + b_i                       ▷ 06. sor: H_i-k előállítása
07 while b_2 ≥ 2                              ▷ 07. sor: van-e még vizsgálandó sorozat?
08          ???
09          ???
10          for i = 1 to c                    ▷ 10–15. sor: aktuális sorozat tesztelése
11              if ???
12                if ???
13                   ???
14              else if ???
15                      go to 19              ▷ áttérés új sorozatra
16          G = G + 1                         ▷ 16. sor: jó sorozat esetén G növelése
17          go to 19                          ▷ áttérés új sorozatra
18 return G                                   ▷ 18. sor: program befejezése
19 if    b_n ≥ 3                              ▷ 19–??. sor: új sorozat előállítása
20       b_n = b_n − 2                        ▷ 19–25. sor: ha b_n = 3
21       H_n = H_n − 2
22       if b_n == b_{n−1} − 2
23          c = c + 1
24          C_c = n − 1
25       go to 07
26 else if b_n == 2                           ▷ 26–46. sor: ha b_n = 2
27          if b_{n−1} > 2
28             b_{n−1} = b_{n−1} − 1
29             H_{n−1} = H_{n−1} − 1
30             b_n = b_{n−1}
31             H_n = H_n + b_n − 3
32          if b_{n−1} páros
33             b_n = b_n − 1
34             C_c = n − 2
35             c = c + 1
36             C_c = n − 1
37          if b_{n−2} > b_{n−1} + 1
38             c = c + 1
39             C_c = n − 1
40          else C_c = n − 2
```

41 **else if** $b_{n-1} == b_n$
42 $C_c = n - 2$
43 **else** $c = c + 1$
44 $C_c = n - 1$
45 **go to** 07
46 **else** $j = n - 1$ ▷ 46–??. sor: ha $b_n = 1$
47 **while** $b_j == 1$
48 $j = j - 1$
49 **if** $b_j > 2$
50 **for** $k = j$ **to** n
51 $b_k = b_j - 1$
52 $H_k = H_k - 1 + (k - j)(b_j - 1)$
53 **if** $(n - j)(b_j + 1)$ páros
54 $b_n = b_n - 1$
55 $H_n = H_n - 1$
56 **if** $j > 1$
57 **if** $b_j < b_{j-1} - 1$
58 $c = c - 1$
59 **if** $b_n < b_{n-1}$
60 $c = c + 1$
61 $C_c = n - 1$
62 **else** $C_c = j - 1$
63 **if** $b_n < b_{n-1}$
64 $c = c + 1$
65 $C_c = n - 1$
66 **go to** 07
60 **else** **for** $k = j - 1$ **to** n
61 $b_k = b_{j-1} - 1$
62 $H_k = H_k - 1 + (k - j + 1)(b_{j-1} - 1)$

63 ???
64 ???
65 ???
66 ???
67 ???
68 ???
69 **go to** 07

4. Leszámlálási eredmények

Eddig például Avis és Fukuda [4], Barnes és Savage [5, 6], Burns [13], Erdős és Moser [63], Frank, Savage and Sellers [25], Kleitman és Winston [50], Rødseth, Sellers, Tverberg [74], Ruskey et al. [75], Simion [78], Stanley [87], Winston és Kleitman [94] publikáltak foksorozatok leszámlálására vonatkozó eredményeket. Az általunk vizsgált sorozatok számával kapcsolatos eredmények találhatók Sloane és Ploffe [84], valamint Stanley [86] könyvében és az *On-line Encyclopedia of Integer Sequences* című honlapon [81, 82, 83] is.

Könnyen belátható, hogy ha l, m és u egész számok, továbbá $l \geq u$ és $1 \geq m$, és $l \leq b_i \leq u$ akkor az (l, u, m)-korlátos sorozatok $\kappa(l, u, m)$ száma

$$\kappa(l, u, m) = (u - l + 1)^m. \tag{13}$$

Tudjuk (lásd [41, 65. oldal]), hogy ha l, u és m egészek, továbbá $u \geq l$ és $m \geq 1$, and $u \geq b_1 \geq \cdots \geq b_n \geq l$, akkor a (l, u, m)-szabályos sorozatok száma $R(l, u, m)$.

$$R(l, u, m) = \binom{m + u - l}{m}. \tag{14}$$

Az előző két speciális eset **??**. nagyon hasznos a leszámlálást végző algoritmusok számára.

Ha $n \geq 1$ egész szám, akkor $R(0, n-1, n)$-szabályos sorozatok száma

$$R(0, n-1, n) = R(n) = \binom{2n-1}{n}. \tag{15}$$

Ha $n \geq 1$ egész szám, akkor $R(1, n-1, n)$-szabályos sorozatok száma

$$R(1, n-1, n) = R_z(n) = \binom{2n-2}{n}. \tag{16}$$

1987-ben Ascher az alábbi explicit megoldást adta a páros sorozatok számára vonatkozóan.

8. Lemma. (Ascher [3], Sloane and Pfoffe [84]) *Ha* $n \geq 1$, *akkor az* n*-páros sorozatok száma* $E(n)$

$$E(n) = \frac{1}{2}\left(\binom{2n-1}{n} + \binom{n-1}{\lfloor n \rfloor}\right). \tag{17}$$

Bizonyítás. Lásd [84]. □

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

9. LEMMA. *Ha $n \geq 1$, akkor*

$$\frac{R(n+2)}{R(n+1)} > \frac{R(n+1)}{R(n)}, \tag{18}$$

$$\lim R_{n\to\infty} = 4, \tag{19}$$

továbbá

$$\frac{4^n}{\sqrt{4\pi n}}(1 - \frac{1}{2n}) < R(n) < \frac{4^n}{\sqrt{4\pi n}}(1 - \frac{1}{8n+8}). \tag{20}$$

Bizonyítás. A (47) egyenlőség alapján

$$\frac{R(n+2)}{R(n+1)} = \frac{(2n+3)!(n+1)n!}{(n+2)!(n+1)!(2n+1)!} = \frac{4n+6}{n+2} = 4 - \frac{2}{n+2}, \tag{21}$$

ahonnan (18) és (19) is közvetlenül adódik. □

A 10 felhasználásával megadhatjuk precise asymptotic order of growth of $E(n)$.

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

$$\frac{E(n+2)}{E(n+1)} > \frac{E(n+1)}{E(n)}, \tag{22}$$

$$\lim E_{n\to\infty} = 4, \tag{23}$$

továbbá

$$\frac{4^n}{\sqrt{\pi n}}(1 - \delta_3(n)) < E(n) < \frac{4^n}{\sqrt{\pi n}}(1 - \delta_4(n)), \tag{24}$$

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

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

????????????????????????????????????? A (15) és (**??**) összehasonlítva látjuk, hogy a , that the order of growth of numbers of even and odd sequences is the same, de több páros sorozat van mint páratlan. Az 1. ábra tartalmazza $R(n)$, $E(n)$ és $E(n)/R(n)$ értékeit $n = 1, \ldots,$ 37-ig. 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.

11. KÖVETKEZMÉNY. *Ha $n \geq 1$, akkor*

$$\frac{E(n+1)}{R(n+1)} < \frac{E(n)}{R(n)} \tag{25}$$

és

$$\lim_{n\leftarrow\infty} \frac{E(n)}{R(n)} = \frac{1}{2}. \tag{26}$$

Bizonyítás. Az állítás a (15) és (17) következménye.

□

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.

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

$$\binom{2n-2}{n-1} = \frac{n}{2n-1} R(n). \tag{27}$$

tartalmaz legalább egy nullát.

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

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

□

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

$$\frac{\sum_{i=0}^{n} i\binom{2n-3}{n-i}}{\binom{2n-1}{n}}. \tag{29}$$

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

A pontos algoritmusokról szóló 3.1. részben beláttuk, hogy elég az n-páros sorozatok nullamentes prefixét megvizsgálni ahhoz, hogy eldöntsük, jó-e a vizsgált sorozat. Mivel a 12 páros sorozatoknak aszimptotikusan csak nullmértékű hányada tartalmaz nullát (és ez a hányad a gyakorlat számára legérdekesebb n-ekre sem nagy), konkrét sorozatok vizsgálatánál nem jelentős az időmegtakarítás. Amikor viszont az összes n-páros sorozatot elemezzük (az átlagos futási idő vagy $G(n)$ meghatározása érdekében), nagyon hasznos a következő lemma.

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

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

$$G(n) = \zeta n + G(n-1). \tag{30}$$

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

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

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

15. Lemma. (Móri [64]). *Ha $n \geq 1$, akkor az n-korlátos sorozatok átlagos szivárvány száma*

$$\frac{n^2}{2n-1}. \tag{31}$$

Bizonyítás. □

Megjegyezzük, hogy ez a lemma választ ad egy Kátai Imre által [42] az átfedéses memóriájú számítógépekkel és a rejtvények megoldásait ellenőrző algoritmusokkal kapcsolatban felvetett kérdésre.

16. Lemma. (Móri [64]). *Ha $n \geq 1$, akkor az n-szabályos sorozatok átlagos szivárvány száma*

$$\frac{n}{2} + \frac{1}{4} + \frac{3}{8n-4}. \tag{32}$$

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

$$\frac{\sum_{k=1}^{n} k\binom{n}{k}\binom{n-1}{k-1}}{\binom{2n-1}{n}}. \tag{33}$$

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

$$\frac{n^2}{2n-1} = \frac{n}{2} + \frac{1}{4} - \frac{3}{8n-4}. \tag{34}$$

□

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

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

Bizonyítás. ???? □

A jó sorozatok $G(n)$ számának jellemzésével kapcsolatos kutatások ígéretes iránya a páros számok pozitív összeadandókra való felbontása, és annak vizsgálata, hogy az ilyen felbontások közül melyek jók [5, 6, 13]. Ezek segítségével sikerült a jó sorozatok számára vonatkozó alábbi aszimptotikus korlátokat bizonyítani.

18. LEMMA. (Burns [13]) *Léteznek olyan pozitív c és C állandók, hogy a jó sorozatok $G(n)$ száma a következő korlátok közé esik:*

$$\frac{4^n}{cn} < G(n) < \frac{4^n}{(\log n)^C \sqrt{n}} \ . \tag{35}$$

5. Tesztelő algoritmusok

A minket is érdeklő focisorozatok vizsgálatakor gyakran szükség van egy foksorozat tesztelésére.

Lehetséges mód a tesztelési idő csökkentésére, ha gyors (lineáris) szűrő algoritmusokat futtatunk, amivel a nem jó sorozatok nagy része elutasítható, így a lassú pontos algoritmusokkal csak a visszamaradó sorozatokat kell vizsgálni.

Bemutatunk egy „paritásos", majd egy „pozitív", egy „binomiális" és végül egy „fejfelező" tesztelő/szűrő algoritmust.

5.1. Paritás teszt

Első tesztünk az Erdős-Gallai tétel első szükséges feltételén alapul. Nagyon hatékony teszt, mivel szabályos sorozatoknak körülbelül fele páratlan sorozat (1. ábra, 26.), és a teszt ezekről lineáris idő alatt megállapítja, hogy biztosan nem jó sorozatok.

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

$$H_n \quad \textit{páros.} \tag{36}$$

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

Ezt az állítást a 2. tétel következményeként is megkaphatjuk. A 19. lemmában javasolt tesztet a következő algoritmus végzi.

Bemenet: n a csúcsok száma ($n \geq 1$); $b = (b_1, b_2, \ldots, b_n)$ n-szabályos sorozat.
Kimenet: L logikai változó
(L = FALSE azt jelzi, hogy b biztosan nem jó, míg L = TRUE azt mutatja, hogy a

teszt *nem tudta eldönteni,* hogy b jó-e);
$H = (H_1, \ldots, H_n)$: H_i a b sorozat első i elemének összege;

Paritás-teszt(n, b, L)

$H_1 = 0$
for $i = 2$ **to** n
    $H_i = H_{i-1} + b_i$
if H_n páratlan
    $L =$ False
    return L
$L =$ True
return L

Ennek az algoritmusnak a lépésszáma minden esetben $\Theta(n)$. Az 1. ábra mutatja milyen hatékony a Paritás-teszt. A Paritás-teszt csak egy közelítő algoritmus mivel csak a (1.) szükséges feltételt ellenőrzi.

5.2. Binomiális teszt

Második tesztünk az Erdős-Gallai tétel másik szükséges feltételének ötletét terjeszti ki. Az algoritmus nevét arról kapta, hogy a fej és a farok belső éleinek a számát egy-egy binomiális együttható segítségével becsüljük. Legyen $T_i = b_{i+1} + \cdots + b_n$ $(i = 1, \ldots, n)$.

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

$$H_i \leq i(i-1) + T_i \quad (i = 1, \ldots, n-1). \tag{37}$$

Bizonyítás. A (37) 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 20. lemmában javasolt tesztet végzi el a következő program.

Bemenet: n a csúcsok száma $(n \geq 1)$; $b = (b_1, b_2, \ldots, b_n)$ n-páros sorozat;
p a b sorozat pozitív elemeinek a száma; *Kimenet.* L: logikai változó
($L =$ False azt jelzi, hogy b biztosan nem jó, míg $L =$ True azt mutatja, hogy a teszt *nem tudta eldönteni,* hogy b jó-e). *Munkaváltozó*:
$T = (T_1, \ldots, T_n)$: T_i az utolsó $n - i$ elem összeg b;
$H = (H_1, \ldots, H_n)$: H_i a b sorozat első i elemének összege.

Binomiális-teszt(n, b, H, L)

$T_0 = 0$
for $i = 1$ **to** $n - 1$
    $T_i = H_n - H_i$

```
    if H_j > i(i − 1) + T_i
        L = FALSE
        return L
L = TRUE
return L
```

Az algoritmus futási ideje legrosszabb esetben $\Theta(n)$, míg a legjobb esetben $\Theta(1)$.
A tesztek alapján BINOMIAL-TEST egy hatékony tesztelő algoritmus (lásd 2. ábra).

5.3. Fej felezése

A b sorozat fokpárosító lehetőségeinek az eddigieknél pontosabb becslését kaphatjuk, ha a fejet két részre osztjuk a (37) binomiális együttható alapján. Legyen $\lfloor i/2 \rfloor = h_i$, p a b sorozat pozitív elemeinek a száma. A $(b_1, \ldots, b_{h_i})$ sorozatot az i indexhez tartozó fej *elejének*, a $(b_{h_i+1}, \ldots, b_i)$ sorozatot pedig az i indexhez tartozó fej *végének* nevezzük.

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

$$\begin{aligned} H_i \leq \min(&\min(H_{h_i}, T_n - T_i, h_i(n-i)) \\ &+ \min(H_i - H_{h_i}, T_n - T_i, (i-h_i)(n-i)), T_i) \\ &+ \min(h_i(i-h_i) + \binom{h_i}{2} + \binom{i-h_i}{2} \quad (i = 1, \ldots, n), \end{aligned} \tag{38}$$

továbbá

$$\min(H_{h_i}, T_n - T_i, h_i(n-i)) + \min(H_i - H_{h_i}, T_n - T_i, (i-h_i)(n-i)) \leq T_i. \tag{39}$$

Bizonyítás. Hasonló gondolatmenettel adódik megkapjuk (**??**)-t is.

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

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

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

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

Bizonyítás. Legyen G a b sorozatot megvalósító G gráf.Ekkor az i indexhez tartozó fej H_i fokszámösszegét lekötő éleinek halmazát öt részhalmazra osztjuk: $(S_{i,1})$ a fej eleje és a farok, a fej vége és a farok közötti, $(S_{i,2})$ a fej eleje és vége közötti, $S_{i,3}$ a fej részein belüli élekre, $S_{i,4}$ a fej eleje $S_{i,5}$ a fej vége.

A következő halmazt kaptuk $X_{i,1}, \ \ldots, \ X_{i,5}$.
$X_{i,1}$ 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.

$X_{i,1}$ legfeljebb a fej elemeinek az összege H_{h_i}, legfeljebb a farok elemeinek az összege $T_n - T_i$, és legfeljebb a fej elejéből és a farokból képezhető párok,

$$X_{i,1} \leq \min(H_{h_i}, T_n - T_i, h_i(n-i)). \tag{40}$$

A similar train of thought results ?????????

$$X_{i,2} \leq \min(H_i - H_{h_i}, T_n - T_i, (i-h_i)(n-i)). \tag{41}$$

$X_{i,3}$ is at most $h_i(i-h_i)$ and at most H_i, implying

$$X_{i,3} \leq \min(h_i(i-h_i), H_i). \tag{42}$$

$X_{i,4}$ is at most $\binom{h_i}{2}$ and at most H_{h_i}, implying

$$X_{i,4} \leq \min(\binom{h_i}{2}, H_{h_i}), \tag{43}$$

while $X_{i,5}$ is at most $\binom{i-h_i}{2}$ and at most $H_i - H_{h_i}$, implying

$$X_{i,5} \leq \binom{i-h_i}{2}. \tag{44}$$

A requirement is also, that the tail can overrun its capacity, that is

$$X_{i,1} + X_{i,2} \leq T_i. \tag{45}$$

Summing of (40), (41), (42), (43), and (44) results

$$H_i \leq X_{i,1} + X_{i,2} + X_{i,3} + 2X_{i,4} + 2X_{i,5}. \tag{46}$$

Substituting of (40), (41), (42), (43), and (44) into (46) results (38), while (45) is equivalent with (39). □

A 21. lemmában javasolt tesztet a következő algoritmus végzi.

Bemenet: n a csúcsok száma ($n \geq 1$); $b = (b_1, \ldots, b_n)$: egy n-páros sorozat,a Binomial-Test által előtesztelt;

$H = (H_1, \ldots, H_n)$, H_i a b sorozat első i elemének összege;
$T = (T_1, \ldots, T_n)$, T_i a b sorozat utolsó $n - i$ elemének összege.

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

Munkaváltozók:
h: a h_i aktuális értéke $X = (X_1, X_2, X_3, X_4, X_5)$: X_j az $X_{i,j}$ értéke a fej végének ??????????? of the end of the head.

FEJFELEZŐ-TESZT(n, b, H, T, L)

for $i = 2$ **to** $n - 1$
    $h = \lfloor i/2 \rfloor$
    $X_1 = \min(H_h, T_n - T_i, h(n - i))$
    $X_2 = \min(H_i - H_h, T_n - T_i, (i - h)(n - i))$
    $X_3 = \min(h(i - h)$
    $X_4 = \binom{h}{2}$
    $X_5 = \binom{i-h}{2}$
    if $H_i > X_1 + X_2 + X_3 + 2X_4 + 2X_5$ or $X_1 + X_2 > T_i$
        L = FALSE
        return L
L = TRUE
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 tanúsága szerint ez nem csökkentené a várható futási időt.

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

5.4. Összetett teszt

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

ÖSSZETETT-TESZT(n, b, E, B, F, L)

$E = B = \pi = F = T = 0$
BINOMIÁLIS-TESZT(L, E, B)
FEJFELEZŐ-TESZT(L, F)
L = TRUE
return E, B, F, L

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

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

6. Közelítő algoritmusok hatékonysága és futási ideje

A tesztek elemzésénél a szabályos és páros sorozatokat vettük alapul. A páros sorozatok halmaza a legkisebb olyan halmaz, melynek elemszámát explicit képlettel meg tudjuk adni. Az $n - 1 \geq b_i \geq 1$ feltételeknek eleget tevő *n-korlátos sorozatok* halmazának elemszámát is könnyű megadni, de ezen halmazok elemszáma túl gyorsan nő n növekedtével. A szabályos sorozatok elemzéséhez szerencsére nem kell *minden* korlátos sorozatot előállítani: elegendő a szabályos sorozatokat előállítani, és a rájuk vonatkozó hatékonysági jellemzőket a nekik megfelelő gyakoriságokkal súlyozni. Például egy azonos elemekből álló *homogén* szabályos sorozatnak egyetlen korlátos sorozat felel meg, míg a különböző elemekből álló $(n, n-1, \ldots, 1, 0)$ „szivárvány” sorozatnak $n!$ különböző korlátos sorozat felel meg.

Az alapvető pontos algoritmusokat kétféle módon próbáljuk gyorsítani (azaz várható futási idejüket csökkenteni). Az egyik út, hogy csökkentjük az általuk elvégzendő ellenőrzések számát. A másik út pedig az, hogy gyors (lineáris) előtesztekkel igyekszünk a rossz sorozatok jelentős részét kiszűrni, hogy csak a lehetséges bemenetek kis hányadánál legyen szükség a viszonylag lassú, de pontos alapalgoritmusokra. A második típusra pedig példa, hogy

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

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

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

A következő közelítő algoritmusokat vizsgáljuk:

1) Nullamentesítő-teszt

2) Binomiális-teszt

3) Fejfelező-teszt

A szabályos sorozatoknak aszimptotikusan a fele páros sorozat. Az 1. ábra az n-szabályos sorozatok számát megadó **??**, valamint az n-páros sorozatok számát megadó 8 lemmák alapján mutatja a konvergencia gyorsaságát (ami a képletek alapján

n	$B_z(n)$	$F_z(n)$	$G(n)$	$G(n+1)/G(n)$
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	

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

természetes).

A 2. ábra a NULLAMENTES BINOMIÁLIS-TESZT, NULLAMENTES FEJFELEZŐ-TESZT és az ERDŐS-GALLAI-GYORSAN programok futási eredményeit tartalmazza.

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

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

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

3. ábra. A nullamentes páros sorozatok száma,nullamentes páros, binomiális, fejfelező tesztek és jó sorozatok számának részaránya a szabályos sorozatokhoz viszonyítva.

Ha $n = 2$, akkor (47) szerint $R(n) = \binom{3}{2} = 3$ szabályos sorozat van: $(1, 1)$, $(1, 0)$ és $(0, 0)$. Ha egy szabályos sorozat elemeinek összege páros, akkor *páros sorozatnak* nevük. Az n hosszúságú páros sorozatok számát $\pi(n)$-nel jelöljük. Ezzel a jelöléssel $\pi(2) = 2$. A Binomiális-teszt által elfogadott, n hosszúságú sorozatok számát $B(n)$-nel jelölve $B(2) = 2$. Az n hosszúságú helyreállítható sorozatok számát jelöljük $G(n)$-nel. Ekkor $G(2) = 2$ és a Binomiális-teszt hibája (hatékonysága) $R_{\mathrm{BT}}(2) = 2/2 = 1$.

Ha $n = 3$, akkor a szabályos sorozatok száma $R(3) = 10$. Ezek közül a (2,2,2), (2,2,0), (2,1,1), (2,0,0), (1,1,0) és (0,0,0) páros, azaz $E(3) = 6$. Ezek közül a Binomiális-

n	Binomiális, s	Binomiális, művelet	Fejfelező, s	Fejfelező, 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

4. ábra. Binomiális teszt és a fejfelező teszt futási ideje másodpercben és műveletszáma.

teszt kizárja a (2,2,0) és (2,0,0) sorozatokat, így $B(3) = 4$. A megmaradt 4 sorozat jó, így $F(3) = T(3) = G(3) = 4$.

Ha $n = 4$, akkor a szabályos sorozatok száma $R(4) = 35$. Ezek közül 19 a páros, és a következő 11 jó: (3,3,3,3), (3,3,2,2), (3,2,2,1), (3,1,1,1,), (2,2,2,2), (2,2,2,0), (2,2,1,1), (2,1,1,0), (1,1,1,1), (1,1,0,0) és (0,0,0,0). A 19 páros sorozat közül Binomiális-teszt is kizárja azt a nyolc sorozatot, amelyeket az Erdős-Gallai kizárna, így $B(4) = G(4) = F(4) = G(4) = 11$.

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

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

??????????????????????????????????????Eddig jó A $R(7) = 1716$ 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 jó 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 következő FAROKFELEZŐ-TESZT $i = 4$ mellett legfeljebb $8+2$ fokot tud lekötni a fej eleje és a farok részei között, legfeljebb további $4+0$ fokot a fej vége és a farok részei között, legfeljebb további 8 fokot a fej két része között, és két fokot a fej elején belül. Ez azonban összesen csak $10 + 4 + 8 + 2 = 24$ fok, ami kevesebb a sorozat $H_7 = 26$ összfokszámánál. Tehát $T(7) = G(7) = 342$.

A 2. ábrán 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. $n > 5$ esetén egyre nő a BINOMIÁLIS-TESZT hibája: $n = 6$ esetén még csak egyetlen páros sorozatról nem ismeri fel, hogy rossz, $n = 7$ esetén már ötször hibázik.

POZITÍV-TESZT $n = 5$-ig hibátlan, a FEJFELEZŐ-TESZT $n = 6$-ig, a FAROKFELEZŐ-TESZT pedig $n = 7$-ig.

A 3. ábrán $R(n)$ értéke $n = 24$-ig az EIS A001700 sorozata [81], $E(n)$ értéke $n = 23$-ig az EIS A005654 sorozata [83], $G(n)$ értéke pedig $n = 23$-ig az EIC A0004251-es sorozata [82]. A többi értéket mi határoztuk meg: sem R(25),..., R(38) sem E(24),...,E(38), sem a $B_z(n)$, $\pi(n)$, $G(n)$ és $(T(n)$ értékek nem szerepelnek az EIS-ben.

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

A témakörben vannak párhuzamos eredmények is [67, 77, 85]. Saját párhuzamos eredményeinket a 8. részben ismertetjük.

7. Pontos algoritmusok futási ideje

A pontos algoritmusok által elvégzett műveleteinek a számát n függvényében alábbi 5. ábra tartalmazza. Az algoritmusok csak a páros, nullamentes sorozatokat vizsgálták. A művelet szám nem tartalmazza a sorozatok előállításának a költségét.

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 műveletszámát n függvényében a 5. ábra tartalmazza.

Az EG-LINEÁRISAN-MIND algoritmus teljes és az elemenkénti futási időit tartalmazza a 6. ábra.

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

Az 8. ábrán az látható, hogy az EG-LINEÁRISAN ciklusmagja átlagosan (gyakoriságokkal súlyozva) hányszor futott le, ha csak a rossz sorozatokat vesszük figyelembe,

n	HH	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

5. ábra. Pontos algoritmusok elvégzett műveleteinek a száma.

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	0.000000007121487	5.34132654018
15	38 781 096	5	3 055 289 271	0.000000008595253	5.25219687963
16	150 273 315	23	12 434 367 770	0.000000009565903	5.17156346504
17	583 407 990	79	50 561 399 261	0.000000007965367	5.09797604337
18	2 268 795 980	297	205 439 740 365	0.00000000727258	5.03056202928

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

illetve a ha minden sorozatot figyelembe veszünk (mindkét esetben csak a páros, nullamentes sorozatokat vettük figyelembe).

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

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

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

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

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

8. Grafikus sorozatok száma

A 2. ábra 1-től 29 csúcsig tartalmazza a grafikus sorozatok számát. A táblázat grafikus sorozatok oszlopa úgy készült, hogy párhuzamosítottuk az Erdős-Gallai-Gyorsan algoritmust. 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.

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

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

9. ábra. A jó sorozatok b_1 szerinti eloszlása $n = 1, \ldots, 12$.

22. Következmény. *Ha Ha m és u pozitív egész számok, akkor az $(1, u, m)$-szabályos sorozatok $R(1, u, m)$ száma*

$$R(1, u, m) = \binom{m + u - 1}{m}. \tag{47}$$

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

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

Köszönetnyilvánítás. A szerzők köszönik Móri Tamás docensnek (ELTE TTK) a 15. és a 16. lemmák bizonyítását, Kása Zoltán egyetemi tanárnak (Sapientia Erdélyi Magyar Tudományegyetem) a kézirat javítására vonatkozó javaslatait, Mányoki Ádámnak (TFM World Kereskedelmi és Szolgáltató Kft.) és Sándor Antal programtervező maetmatikusnak pedig a programok nagy processzorigényű futtatásához nyújtott segítséget. A kutatás az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg (a támogatás száma TÁMOP 4.2.1/B-09/1/KMR-2010-0003).

Hivatkozások

[1] Anholcer, M., Babiy, V., Bozóki, S., Koczkodaj, W. W.: *A simplified implementation of the least squares solution for pairwise comparisons matrices.* CEJOR Cent. Eur. J. Oper. Res. **19,** (4) (2011) 439–444. ⇒1

[2] Arikati, S. R.,Maheshwari, A.: *Realizing degree sequences in parallel.* SIAM J. Discrete Math. **9,** (2) (1996) 317–338. ⇒

[3] Ascher, M.: *Mu torere: an analysis of a Maori game.* Math. Mag. **60(2),** (1987) 90–100. ⇒ 16

[4] Avis, D., Fukuda, K.: *Reverse search for enumeration.* Discrete Appl. Math. **2,** (1993) 21–46. ⇒16

[5] Barnes, T. M., Savage, C. D.: *A recurrence for counting graphical partitions.* Electron. J. Combin. **2,** (1995) R11, 10 pp. ⇒16, 20

[6] Barnes, T. M., Savage, C. D.: *Efficient generation of graphical partitions.* Discrete Appl. Math. **78(1–3),** (1997) 17–26. ⇒1, 2, 16, 20, 38

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

[8] 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

[9] Bödei, N.: *Degree sequences of graphs* (Hungarian), Mathematical master thesis (supervisor A. Frank), Dept. of Operation Research of Eötvös Loránd University, Budapest, 2010, 43 oldal. ⇒

[10] 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

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

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

[13] Burns, J. M.: *The number of degree sequences.* PhD Dissertation, MIT, 2007. ⇒2, 16, 20

[14] 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

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

[16] Cooper, J., Lu, L.: *Graphs with asymptotically invariant degree sequences under restriction.* Internet Mathematics **7,** (1) 67–80. ⇒

[17] 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, 12

[18] De Agostino, S. , Petreschi, R.: *Parallel recognition algorithms for graphs with restricted neighbourhoods. Internat. J. Found. Comput. Sci.* **1,** (2) (1990) 123–130. ⇒

[19] 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

[20] Dessmark, A., Lingas, A., Garrido, O.: *On parallel complexity of maximum f-matching and the degree sequence problem.* Mathematical Foundations of Computer Science 1994 (Košice, 1994), LNCS **841,** Springer, Berlin, 1994, 316–325. ⇒

[21] Erdős, P., Gallai, T.: *Gráfok előírt fokú pontokkal.* Mat. Lapok **11,** (1960) 264–274. ⇒1, 3, 4, 38

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

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

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

[25] Frank, D. A., Savage, C. D., Sellers, J. A.: *On the number of graphical forest partitions.* Ars Combin. **65,** (2002) 33–37. ⇒16

[26] 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, 38

[27] Hakimi, S. L.: *On realizability of a set of integers as degrees of the vertices of a linear graph II. Uniqueness.* SIAM J. Appl. Math., **11(1),** (1963) 135–147. ⇒2

[28] Hakimi, S. L.: *On the degrees of the vertices of a graph.* F. Franklin Institute, **279,** (4) (1965) 290–308. ⇒

[29] Havel, V.: *A remark on the existence of finite graphs.* (cseh) Časopis Pěst. Mat. **80,** (1955), 477–480. ⇒1, 3, 38

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

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

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

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

[34] 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

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

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

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

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

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

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

[41] Járai, A.: *Introduction to Mathematics* (Hungarian). ELTE Eötvös Kiadó, Budapest, 2005. ⇒16

[42] Kátai, I.: *Szóbeli közlés.* Budapest, 2010. ⇒19

[43] 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

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

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

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

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

[48] 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

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

[50] Kleitman, D. J., Winston K. J.: *Forests and score vectors.* Combinatorica **1(1),** (1981) 49–54. ⇒16

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

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

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

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

[55] 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

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

[57] 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

[58] Lucz, L., Iványi, A., Sótér, P., Pirzada, S.: *Testing and enumeration of football sequences.* Abstracts of MaCS 2012, ed. by Z. Csörnyei (Siófok, February 9–12). ⇒

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

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

[61] Miklós, I.: *Graphs with prescribed degree sequences.* Lecture in Alfréd Rényi Institute of Mathematics, 16 November 2009. ⇒

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

[63] Moon, J. W.: *Topics on Tournaments.* Holt, Rinehart, and Winston, New York, 1968. ⇒16

[64] Móri, T.: *Szóbeli közlés.* Budapest, 2011. ⇒19

[65] 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. ⇒

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

[67] Pécsy G., Szűcs, L.: *Parallel verification and enumeration of tournaments.* Stud. Univ. Babeş-Bolyai, Inform. **45(2),** (200) 11–26. ⇒29

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

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

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

[71] 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

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

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

[74] 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, 16

[75] Ruskey, F., Cohen, R., Eades, P., Scott, A.: *Alley CAT's in search of good homes.* Congr. Num., **102,** (1994) 97–110. ⇒1, 2, 16, 38

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

[77] Siklósi, B.: *Soros és párhuzamos algoritmusok összehasonlítása sportversenyekkel kapcsolatos problémákban.* Programtervező matematikus diplomamunka. ELTE TTK, Budapest, 2001. 69 oldal. ⇒29

[78] Simion, R.: *Convex polytopes and enumeration.* Advances in Applied Math. **18(2)** (1996) 149–180. ⇒16

[79] Sloane N. J. A., Plouffe S.: *The Encyclopedia of Integer Sequences.* Academic Press, 1995. ⇒16

[80] Sloane N. J. A. (szerk.): *Encyclopedia of Integer Sequences.* 2011. ⇒2, 13

[81] Sloane N. J. A.: *The number of ways to put $n+1$ indistinguishable balls into $n+1$ distinguishable boxes.* In (ed. N. J. A. Sloane): The On-line Encyclopedia of the Integer Sequences. 2011. ⇒1, 16, 29, 38

[82] 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. ⇒16, 29

[83] 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. ⇒16, 29

[84] Sloane, N., J., A., S. Plouffe: *The Encyclopedia of Integer Sequences.* Academic Press, 1995. ⇒16

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

[86] Stanley, R.: *Enumerative Combinatorics. Vol. 2.* Cambridge University Press, Cambridge, 1997. ⇒16

[87] 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. ⇒16

[88] Temesi, J.: *Pairwise comparison matrices and the error-free property of the decision maker.* CEJOR Cent. Eur. J. Oper. Res. **19,** (2) (2011) 239–249. ⇒1

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

[90] Tripathi, A., Vijay, S.: *A note on a theorem of Erdős & Gallai.* Discrete Math. **265(1–3),** (2003) 417–420. ⇒8, 9

[91] 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, 38

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

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

[94] Winston, K. J., Kleitman, D. J.: *On the asymptotic number of tournament score sequences.* J. Combin. Theory Ser. A. **35,** (1983) 208–230. ⇒16

Beérkezett:

IVÁNYI ANTAL
tony@compalg.inf.elte.hu

LUCZ LORÁND
lorand.lucz@gmail.com

SÓTÉR PÉTER
mapoleon@freemail.hu

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

Budapest, 2011. szeptember 7.

LINEAR ERDŐS-GALLAI TEST

ANTAL IVÁNYI, LORÁND LUCZ, PÉTER SÓTÉR

Havel in 1955 [29], Erdős and Gallai in 1960 [21], Hakimi in 1962 [26], Ruskey, Cohen, Eades and Scott 1994-ben [75], Barnes and Savage in 1997 [6], Tripathi, Venugopalan and West in 2010 [91] proposed a method to decide, whether a sequence of nonnegative integers can be the degree sequence of a simple graph (such sequences are called graphical). The running time of their algorithms in worst case is $\Omega(n^2)$. In this paper we propose a new algorithm called Erdős-Gallai-Linear, whose worst running time is linear. Since in the case of a graphical sequence all elements of the investigated sequence are to be tested, in the case of RAM model of computations Erdős-Gallai-Linear is asymptotically optimal. Using this algorithm we determined the number of the degree sequences of the simple graphs for $n = 24,\ 25,\ 26,\ 27$ and $n = 28$ (*Online Encyclopedia of Integer Sequences* [81] contains these numbers only for $n = 1,\ \ldots,\ 23$).