

FOKSOROZATOK PÁRHUZAMOS LESZÁMLÁLÁSA

IVÁNYI ANTAL, KÁSA ZOLTÁN

Az egyszerű gráfok foksorozatainak tesztelése, előállítás és leszámllása gazdag irodalommal rendelkezik. Cikkünkben párhuzamos algoritmusokat ismertetünk, melyek segítségével leszámlltuk az egyszerű gráfok ($G(n)$) és az izolált csúcsot nem tartalmazó egyszerű gráfok ($G_z(n)$) foksorozatait $n = 24, \dots, 31$ csúcs esetén.

1. Bevezetés

A gyakorlatban gyakran előforduló probléma különböző objektumok rangsorolása (példák találhatóak [89, 95]-ben). Az egyik módszer az, hogy az objektumokat páronként összehasonlítjuk, az összehasonlítás alapján – rendszerint különböző, de esetenként azonos számú – pontokat rendelünk az összehasonlított objektumokhoz, és az összegyűjtött pontjaik száma (vagy a nyert és veszített pontok számának különbsége) alapján rangsoroljuk őket. Az azonos pontszámú eset irányítatlan [14, 15, 16, 17, 20, 22, 23, 24, 25, 26, 33, 34, 35, 39, 44, 47, 48, 53, 56, 57, 59, 60, 62, 63, 61, 66, 76, 77, 81, 82, 89, 91, 92, 93, 94, 95, 97, 99, 100, 105, 106, 110, 111, 112, 113, 114, 118, 119, 120, 121, 122, 126, 130, 131, 132, 133, 134, 135, 136, 137, 143, 144, 158, 159, 162, 164, 166, 167, 169, 170, 171, 172, 173, 184, 190, 191], míg a különböző pontszámú eset irányított gráfokkal [1, 4, 6, 7, 8, 9, 10, 11, 12, 13, 18, 36, 21, 27, 28, 29, 30, 31, 36, 37, 38, 40, 43, 50, 51, 52, 55, 58, 64, 67, 68, 69, 70, 71, 72, 73, 75, 80, 83, 84, 85, 86, 87, 88, 88, 90, 97, 98, 104, 103, 108, 109, 117, 123, 124, 129, 137, 138, 139, 140, 141, 142, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 160, 163, 165, 168, 172, 174, 175, 176, 179, 180, 185, 186, 189] kapcsolatos problémákra vezet.

Különösen gazdag irodalommal rendelkezik az az eset, amikor az eredmények egyszerű gráfként reprezentálhatóak (azaz az összehasonlítás eredményeképpen vagy mindkét objektum nulla, vagy mindkettő egy pontot kap), és a probléma a potenciális foksorozatok [71, 187] tesztelése, rekonstruálása és leszámllása. Havel 1955-ben [78], Erdős és Gallai 1960-ban [56], Hakimi 1962-ben [74], Tripathi és munkatársai 2010-ben [183] javasoltak egy-egy algoritmust annak eldöntésére, hogy egy nemnegatív egészekből álló sorozat lehet-e egy egyszerű gráfhoz tartozó foksorozat. Az algoritmusaik futási ideje legrosszabb esetben $\Omega(n^2)$.

2007-ben Takahashi [177], 2009-ben Hell és Kirkpatrick [79], 2011-ben Iványi és munkatársai [95], 2012-ben pedig Király [107] egymástól függetlenül javasolták az Erdős–Gallai-algoritmus lineáris futási idejű változatát, 2012-ben pedig Iványi [87] a Havel–Hakimi-algoritmus tesztelő változatát, melyek futási ideje legrosszabb esetben $\Theta(n)$.

A klasszikus Havel–Hakimi és Erdős–Gallai-tételekre számos új bizonyítás ismert [32, 45, 116, 181, 182, 183]. Kiterjesztéseik $(0, b)$ -gráfokra [46, 145] és (a, b) -gráfokra [84, 85, 87, 97, 151] szintén ismertek. Bemutatjuk a Havel–Hakimi-algoritmus tesztelő változatait [87], és összehasonlítjuk a korábbi lineáris tesztelő algoritmusokkal.

Léteznek korai párhuzamos eredmények is, lásd például [3, 115, 149, 168, 174]. Az Erdős–Gallai-enumerating algoritmus (EGE1) [95] párhuzamos verzióját használtuk fel a 24, \dots , 31 csúcús egyszerű gráfok foksorozatainak leszámolására.

Megemlíttjük, hogy más szerzők is felhasználták algoritmusunkat különböző, foksorozatokkal kapcsolatos problémák megoldására [16, 41, 54].

Legyenek l, m, n, u egész számok, $u \geq l, m \geq 1, n \geq 1$. Ha $l \leq b_i \leq u$ ($i = 1, \dots, m$), akkor az egész számokból álló $b = b_1, \dots, b_m$ sorozatot (l, u, m) -korlátosnak nevezzük. Ha b $(0, n - 1, n)$ -korlátos, akkor n -korlátosnak, ha az $n - 1 \geq b_1 \geq \dots \geq b_n \geq 0$ feltételt is kielégíti, akkor n -szabályosnak nevezzük. Ha egy n -szabályos sorozat elemeinek összege páros, akkor n -páros. Ha létezik olyan n csúcús egyszerű gráf, amelynek b a foksorozata, akkor azt mondjuk, hogy a b sorozat n -gráfós [71, 188]. Ha nincs ilyen gráf, akkor b nemgráfós. Egy n -szabályos b sorozat első i elemét az i -hez tartozó fejnek, az azt követő $n - i$ elemét pedig az i indexhez tartozó farknak nevezzük.

Cikkünk fő célja a lineáris ERDŐS–GALLAI-algoritmus párhuzamos megvalósításának bemutatása. Bár a probléma önmagában is érdekes, számunkra a fő motívációt Frank András [65, Research problem 2.3.1] monográfiájában szereplő kérdés megválaszolása jelenti: „Döntsük el, hogy egy n egész számból álló sorozat lehet-e egy n csapatból álló labdarúgó bajnokság végeredménye.” A potenciális futbalsorozatok tesztelésének és rekonstruálásának fontos részproblémája a döntetlen sorozatok kezelése. Mivel az „Ez a sorozat gráfós?” és az „Ez a sorozat egy labdarúgó döntetlen sorozat?” kérdések ekvivalensek (lásd [87, 99, 115, 121, 165]), ezért a gyors válasz létfontosságú számunkra.

Cikkünk felépítése a következő. A bevezető 1. rész után a 2. részben bemutatjuk a párhuzamos programunk alapjául szolgáló lineáris ugró Erdős–Gallai-algoritmust (EGLJ), majd a 3. részben ismertetjük az EGLJ-algoritmus leszámoló változatát. A 4. részben a feladat szabályos sorozatokon alapuló felbontását, míg az 5. részben a páros sorozatokon alapuló felbontását mutatjuk be. A 6. részben az EGLJ-algoritmus párhuzamos változatát (EGP), a 7. részben pedig a grafikus és nem grafikus sorozatok átugrásának lehetőségeit elemezzük, végül a 8. részben összegezzük az eredményeket.

2. Lineáris ugró Erdős–Gallai-algoritmus (EGLJ)

Korábbi cikkeinkben [87, 89, 95] ismertettük a klasszikus Havel–Hakimi (HH) [74, 78] és Erdős–Gallai-algoritmusokat (EG) [56], valamint ezek különböző javításait. Fontos megjegyezni, hogy HHL lineáris csak teszteli a megvizsgált sorozatokat, de nem állít elő a gráfok sorozatoknak megfelelő gráfot. Ugyanakkor a helyreállító változat mindenképpen $\Omega(n^2)$ időt igényel.

2011-ben [95] bemutattuk a HH-algoritmus (HH) néhány gyorsított változatát, majd 2012-ben [87] a lineáris Havel–Hakimi-algoritmust (HHL).

Jelen cikkünkben táblázatokkal, valamint grafikonokkal összehasonlítjuk a HH-és EG-algoritmusok különböző változatainak futási idejét.

A cikk programjaiban a [49] tankönyvben leírt pszeudokód konvenciókat követjük.

A HH-algoritmus gyorsított változatai közül csak a HHT-algoritmus kódját ismételjük meg, mivel közreadjuk annak elemzését, mennyi esély van arra, hogy a HHT már az első lépésben felismer egy nemgrafikus n -szabályos sorozatot. Itt és a továbbiakban n a sorozat hosszát (a gráf csúcsainak számát) jelöli, $b = b_1, \dots, b_n$ pedig a vizsgálandó $(0, 1, n)$ -szabályos sorozatot.

2.1. Algoritmus. HAVEL-HAKIMI-TESTING (n, b)

Bemenet: n : csúcsok száma ($n \geq 1$);

$b = b_1, \dots, b_n$: a megvizsgálandó n -szabályos sorozat.

Kimenet: 0 vagy 1: 1, ha b gráfok, és 0 egyébként.

Munkaváltozó: i : ciklusváltó.

```

1. for  $i = 1$  to  $n - 1$                                 // 1–6. sor:  $b$  elemeinek tesztelése
2.     if  $i + b_i > n$  vagy  $b_{i+b_i} = 0$                 // 2–3. sor:  $b$  nem grafikus
3.         return 0
4.     for  $j = i + 1$  to  $i + b_i$ 
5.          $b_j = b_j - 1$ 
6.      $b_{i+1}, \dots, b_n$  rendezése nemnövekvő sorrendbe
7. return 1                                             // 7. sor:  $b$  grafikus

```

Ebben a HHT-algoritmusban az eredeti HH-algoritmust kiegészítettük a 3. sorban végzett, a bemenő sorozat minden elemére legfeljebb konstans időt igénylő ellenőrzésével, amely jelzi, ha a sorozat rövidege, vagy a nullák nagy száma miatt már a legnagyobb fokszerű csúcsot sem tudjuk a szükséges számú másik csúccsal összekötni.

Ez az ellenőrzés a bemenetként szóbjövő [89, 95]

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

$(0, n-1, n)$ -szabályos sorozat közül pontosan azokat szűri ki, amelyekben $b_1 = n-1$ és $b_n = 0$, vagy $b_1 = n-2$ és $b_{n-1} = 0$ vagy ... vagy $b_1 = 1$ és $b_2 = 0$. Így az (1) képlet segítségével azt kapjuk, hogy a kiszűrt sorozatok $F(n)$ száma

$$F(n) = \sum_{i=1}^{n-1} F_i(n) = \sum_{i=1}^{n-1} \binom{2i-1}{i-1}.$$

Az 1. táblázat tartalmazza a csapatok n számát, az $n-1$ ($F_{n-1}(n)$), az $n-2$ ($F_{n-2}(n)$) és az $n-3$ ($F_{n-2}(n)$) elemmel kezdődő, valamint az összesen kiszűrt sorozatok $F(n)$ számát $n = 1, \dots, 32$ csapat esetén.

A következő 2.2. algoritmus EG leggyorsabb, általunk ismert soros változata (ez a [125] dolgozatban leírt algoritmus egyszerűsített változata). Az algoritmus alapja a következő állítás.

2.1. TÉTEL. Legyen $n \geq 1$ és $b = b_1, \dots, b_n$ egy n -szabályos sorozat.

A b sorozat akkor és csak akkor gráfos, ha elemeinek összege páros, továbbá ha $i > w$, akkor

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

és ha $i \leq w_1$, akkor

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

Bizonyítás. Lásd [95]. □

2.2. TÉTEL. (Iványi, Lucz, Móri, Sótér, 2011 [95]) Az ERDŐS-GALLAI-LINEAR-JUMPING $\Theta(n)$ idő alatt dönti el, hogy egy n -szabályos $b = b_1, \dots, b_n$ sorozat gráfos-e.

Bizonyítás. Lásd [95]. □

2.2. Algoritmus. ERDŐS-GALLAI-LINEAR-JUMPING (n, F)

Bemenet. n : a foksorozat hosszát adja meg;

b : az ellenőrizendő foksorozat.

Kimenet. ha a b sorozat grafikus, akkor 1, egyébként pedig 0.

1. $H_1 = b_1$ // 1. sor: H_1 számítása
2. **for** $i = 2$ **to** n // 2-3. sor: H értékeinek számítása
3. $H_i = H_{i-1} + b_i$
4. **if** H_n páratlan // 4-5. sor: paritás ellenőrzése
5. **return** 0 // 4. sor: hibás sorozat elutasítása
6. $w = n$ // 5. sor: súlypont kezdeti értékének beállítása
7. $i = 1$ // 7-15. sor: sorozat ellenőrzése

```

8. while  $b_i = b_i + 1$  and  $i \leq n - 1$  // 8–9. sor: ellenőrzőpont ellenőrzése
9.      $i = i + 1$ 
10.    while  $w > 1$  and  $f_i \leq i$  // 10–11. sor: súlypont frissítése
11.         $w = w - 1$ 
12.    if  $w < i$  // 12–13. sor: súlypont ellenőrzőpont előtt van
13.        while  $w > 1$  and  $i < 1$  // 13–14. sor: súlypont frissítése
14.    if  $w > i$  // 14–15. sor: súlypont ellenőrzőpont után van
15.        if  $H_i > H_n - H_i + i(i - 1)$  // 15–16. sor: rossz sorozat elutasítása
16.            return 0
17.        else if  $H_i > H_n - H_w + i(i - 1)$  // 17–18. sor: rossz sorozat elutasítása
18.            return 0
19.    return 1 // 19. sor: jó sorozat elfogadása

```

Egy (szabályos) páros $s = s_1, \dots, s_n$ sorozatot *nullamentesnek* nevezünk, ha $s_n > 0$. A 2. táblázatban láthatjuk a tesztelt nullamentes sorozatok számát ($E_z(n)$), valamint az egy sorozatra jutó átlagos tesztelés idejét mikroszekundumban az EGL ($T_{\text{EGL}}(n)/E_z(n)$), EGLJ ($T_{\text{EGLJ}}(n)/E_z(n)$) és HHL ($T_{\text{HHL}}(n)/E_z(n)$) esetén, ahol $n = 10, \dots, 19$. Az $n = 1, \dots, 9$ értékek nem szerepelnek a táblázatban, mivel a futási idejüket a mérés során programunk nullára kerekítette.

Az 1. ábrán láthatjuk az EGL, EGLJ és HHL átlagos futási idejét a csúcok számának függvényében. A háromszögek mutatják az $(n, T(n))$ párokat a lineáris Erdős–Gallai-algoritmus (EGL) esetében, a négyzetek a lineáris ugró Erdős–Gallai-algoritmus (EGLJ) esetében, míg a gyémántok a lineáris Havel–Hakimi-algoritmust (HHL) jelzik.

A 3. táblázatban található a nullamentes gráfos sorozatok $G_z(n)$ száma, valamint a nullamentes sorozatok ellenőrzésének átlagos műveletszáma az EGL ($O_{\text{EGL}}(n)/E_z(n)$), EGLJ ($O_{\text{EGLJ}}(n)/E_z(n)$) és HHL ($O_{\text{HHL}}(n)/E_z(n)$) esetén, ahol $n = 2, \dots, 19$.

A 2. ábrán láthatjuk az EGL, EGLJ és HHL átlagos műveletszámát a csúcok számának függvényében. A (honlapon lévő változatban zöld) háromszögek mutatják az $(n, O(n))$ párosokat a lineáris Erdős–Gallai-algoritmus (EGL) esetében, a (piros) négyzetek a lineáris ugró Erdős–Gallai-algoritmus (EGLJ) esetén, míg a (kék) gyémántok a lineáris Havel–Hakimi-algoritmust (HHL) jelzik.

Műveletnek számítottunk minden összehasonlítást, összeadást, kivonást, szorzást, osztást, maradékképzést és értékadást. Kivételt képeztek a ciklusok törzsében szereplő indexváltozók. Például a $H[i] - i \cdot (i - 1) > R$ parancs elvégzéséhez három műveletre van szükség: a $H[i] - i \cdot (i - 1)$ kivonásra, az $i \cdot (i - 1)$ szorzásra, valamint a $H[i] - i \cdot (i - 1) > R$ összehasonlításra. Az $i - 1$ típusú kivonásokat *nem* számoltuk, mert az i a hozzá tartozó ciklus indexváltozója.

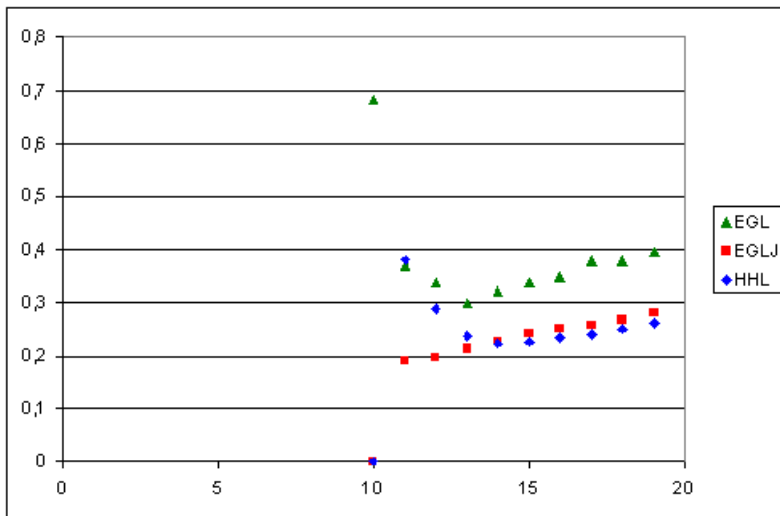
Példaként tekintsük részletesen az $(1, 1)$ nullamentes sorozat tesztelését. A példa az EGL [95], az EGLJ [125], valamint a HHL [87] pszeudokódján alapul.

n	$F_{n-1}(n)$	$F_{n-2}(n)$	$F_{n-3}(n)$	$F(n)$	$\frac{100F(n)}{R(n)}$
2	1	—	—	1	33.3333
3	3	1	—	4	40.0000
4	10	3	1	14	40.0000
5	35	10	3	49	38.8889
6	126	35	10	175	37.8788
7	462	125	35	637	37.1212
8	1716	462	126	2353	36.5657
9	6435	1716	462	8788	36.1497
10	24310	6435	1716	33098	35.8289
11	92378	24310	6435	125476	35.5742
12	352716	92378	24310	478192	35.3671
13	1352078	352716	92378	1830270	35.1955
14	5200300	1352078	352716	7030570	35.0507
15	20058300	5200300	1352078	27088870	34.9269
16	77558760	20058300	5200300	104647630	34.8198
17	300540195	77558760	20058300	405187825	34.7263
18	1166803110	300540195	77558760	1571990935	34.6439
19	4537567650	1166803110	300540195	6109558585	34.5707
20	17672631900	4537567650	1166803110	23782190485	34.5053
21	68923264410	17672631900	4537567650	92705454895	34.4465
22	269128937220	68923264410	17672631900	361834392115	34.3933
23	1052049481860	269128937220	68923264410	1413883873975	34.3450
24	4116715363800	1052049481860	269128937220	5530599237775	34.3008
25	16123801841550	4116715363800	1052049481860	21654401079325	34.2604
26	63205303218876	16123801841550	4116715363800	84859704298201	34.2232
27	247959266474052	63205303218876	16123801841550	332818970772253	34.1889
28	973469712824056	247959266474052	63205303218876	1306288683596310	34.1572
29	3824345300380220	973469712824056	247959266474052	5130633983976530	34.1277
30	15033633249770500	3824345300380220	973469712824056	20164267233747100	34.1003
31	59132290782430700	15033633249770500	3824345300380220	79296558016177800	34.0747
32	232714176627630000	59132290782430700	15033633249770500	312010734643808000	34.0507

1. táblázat. Csapatok n száma, az $n-1$ és $n-2$, valamint $n-3$ elemmel kezdődő, valamint összesen kiszűrt sorozatok $F(n)$ száma $n = 1, \dots, 32$ csapat esetén.

n	$E_z(n)$	$\frac{T_{EGL}(n)}{E_z(n)}$	$\frac{T_{EGLJ}(n)}{E_z(n)}$	$\frac{T_{HHL}(n)}{E_z(n)}$
10	21 942	0.683620	0.000000	0.000000
11	83 980	0.369136	0.190521	0.381083
12	323 554	0.336883	0.194712	0.287433
13	1 248 072	0.299662	0.213128	0.237967
14	4 829 708	0.319895	0.226101	0.222788
15	18 721 080	0.338281	0.241371	0.226643
16	72 714 555	0.348197	0.251665	0.233406
17	282 861 360	0.379355	0.255846	0.240789
18	1 101 992 870	0.377512	0.267014	0.249460
19	4 298 748 300	0.394.319	0.281491	0.261416

2. táblázat. Nullamentes sorozatok száma, valamint az átlagos futási idő mikroszekundumban az EGL, EGLJ, valamint HHL esetén.

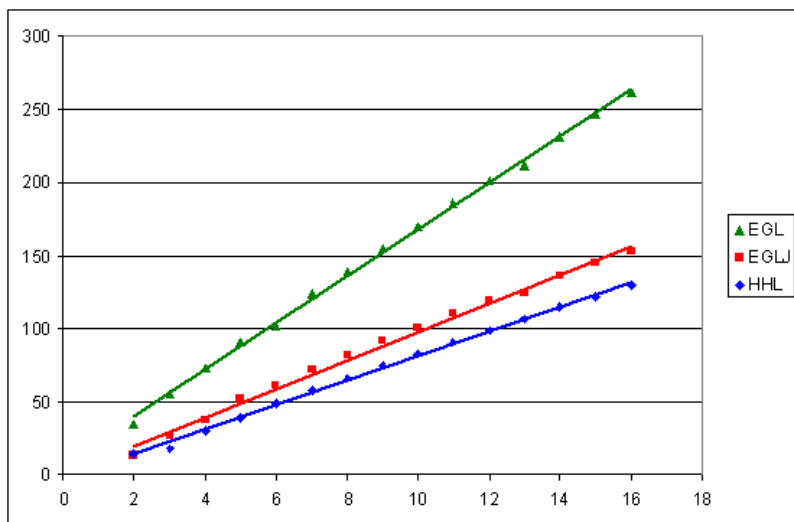


1. ábra. Az EGL, EGLJ és a HHL átlagos futási ideje.

A HHL-nek 14 műveletre van szüksége: 1 összehasonlításra az 1. sorban, 1 összehasonlításra a 3. sorban, 1 értékadásra az 5. sorban, 5 műveletre a 6. és 7. sorban (1 $i = 1$ értékadás, 1 összeadás az i növelésénél, 2 összehasonlítás $i < n$ -re, 1 $H_1 = s_1$ értékadás), 1 maradékképzésre, 1 összehasonlításra a 8. sorban, 1

n	$G_z(n)$	$\frac{O_{EGL}(n)}{E_z(n)}$	$\frac{O_{EGLJ}(n)}{E_z(n)}$	$\frac{O_{HHL}(n)}{E_z(n)}$
2	1	35.000	13.000	14.000
3	2	55.000	26.500	18.000
4	7	73.000	37.667	29.889
5	20	91.000	51.429	39.357
6	71	101.609	61.473	48.591
7	240	123.495	72.480	57.553
8	871	139.162	82.042	66.123
9	3148	154.944	91.751	74.552
10	11655	170.421	100.929	82.749
11	43332	185.885	110.047	90.824
12	162769	201.209	118.930	98.758
13	614718	212.177	124.720	106.591
14	2330537	231.659	136.373	114.739
15	8875768	246.785	144.939	121.976
16	33924858	261.846	153.411	129.552

3. táblázat. Csúcsok n nullamentes gráfos sorozatok $G_z(n)$ száma, valamint az egy nullamentes páros sorozatra jutó átlagos műveletszám az EGL, EGLJ és a HHL algoritmus és $2, \dots, 16$ csúcs esetén.



2. ábra. Az EGL, EGLJ és a HHL átlagos műveletszáma nullamentes páros sorozatok tesztelése esetén.

értékkadásra a 10. sorban, 2 kivonásra és egy értékkadásra a 13. sorban, valamint 1 összehasonlításra a 14–22. sorban.

Az EGLJ-nek 13 műveletre van szüksége: 1 értékadásra az 1. sorban, 5 műveletre a 2–3. sorban (ciklusváltozó kezdeti értékének beállítása és növelése, 1 összehasonlítás, két H_i értékadás), 1 maradékképzésre és 1 összehasonlításra az 5–8. sorban, 1 értékadásra a 9. sorban, 4 műveletre a 10–28. sorban (ciklusváltozó kezdeti értékének beállítása és növelése, 1 összehasonlítás a 11. sorban és 1 összehasonlítás a 17. sorban).

Az EGL-nek 35 műveletre van szüksége: 1 értékadásra az 1. sorban, 9 műveletre a 2–3. sorban (ciklusváltozó kezdeti értékének beállítása és növelése kétszer, összehasonlítása kétszer, 2 összeadás a H_i esetében, 2 értékadás a H_i esetében) 1 maradékképzésre, 1 összehasonlításra a 4. sorban, 1 értékadásra a 7. sorban, 7 műveletre a 8–12. sorban (ciklusváltozó kezdeti értékének beállítása és növelése kétszer, összehasonlítása kétszer, elágazás tesztelése kétszer), 4 műveletre a 13–14. sorban (ciklusváltozó kezdeti i értékének beállítása és csökkentése, 1 összehasonlítás, 1 értékadás), 11 műveletre a 15–23. sorban (ciklusváltozó kezdeti értékének beállítása és növelése, 9 összehasonlítás).

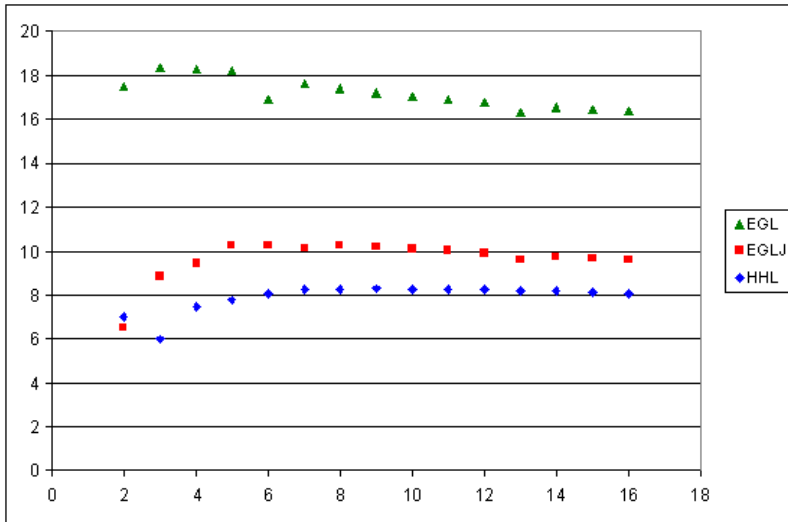
A 4. táblázatban található az amortizált műveletek számának és a nullamentes páros sorozatok $E_z(n)$ számának hányadosa az EGL ($O_{\text{EGL}}(n)/E_z(n)$), az EGLJ ($O_{\text{EGLJ}}(n)/E_z(n)$) és a HHL ($O_{\text{HHL}}(n)/E_z(n)$) algoritmusok esetén, ahol $n = 2, \dots, 16$.

n	$\frac{O_{\text{EGL}}(n)}{E_z(n)}$	$\frac{O_{\text{EGLJ}}(n)}{E_z(n)}$	$\frac{O_{\text{HHL}}(n)}{E_z(n)}$
2	17.500	6.500	7.000
3	18.333	8.833	6.000
4	18.250	9.417	7.472
5	18.200	10.286	7.781
6	16.935	10.246	8.099
7	17.642	10.154	8.222
8	17.395	10.255	8.265
9	17.216	10.195	8.284
10	17.042	10.093	8.275
11	16.899	10.004	8.257
12	16.767	9.911	8.230
13	16.321	9.593	8.199
14	16.547	9.741	8.196
15	16.452	9.663	8.132
16	16.365	9.588	8.097

4. táblázat. Az amortizált műveletek száma a nullamentes sorozatokra nézve az EGL, EGLJ, valamint HHL-algoritmusokra $n = 2, \dots, 16$ csúcs esetében.

A 3. ábrán láthatjuk az EGL, EGLJ és HHL amortizált műveletszámát. A (zöld) háromszögek mutatják az $(n, O(n))$ párosokat a lineáris Erdős–Gallai-algoritmus esetében (EGL), a (piros) négyzetek a lineáris ugró Erdős–Gallai-algoritmus esetében, míg a (kék) gyémántok a lineáris Havel–Hakimi-algoritmust jelzik.

A 4. táblázat és a 3. ábra alapján kijelenthető, hogy az EGL, EGLJ és HHL-algoritmusok CAT (konstans időben amortizált) algoritmusok (lásd [161]).



3. ábra. Az amortizált műveletek száma a nullamentes sorozatokra nézve az EGL, EGLJ, valamint HHL esetében.

3. Leszámláló Erdős–Gallai-algoritmus (EGE)

A gráfelmélet egyik klasszikus problémája különböző gráfok – többek között az egyszerű gráfok – foksorozatának leszámllása. Példaként tekinthető a *The On-Line Encyclopedia of Integer Sequences* [170], amely $n = 1, \dots, 31$ csúcsra tartalmazza az egyszerű gráfok foksorozatának számát (az $n = 20, \dots, 23$ értékeket 2011. júliusában adta meg Nathann Cohen, míg az $n = 24, \dots, 29$ értékek a 2011. november 15-én elért eredményeink [95]).

Az új, gyorsított EGL-algoritmust alkalmaztuk a nagyobb értékű n -ek meghatározására.

A szabályos sorozatok ellenőrzését és a gráfos sorozatok leszámllását tűztük ki célul. Az n -szabályos sorozatok $R(n)$ száma

$$R(n) = \binom{2n-1}{n}, \quad (2)$$

a nullamentes sorozatok $R_z(n)$ száma pedig

$$R_z(n) = \binom{2n-2}{n}. \quad (3)$$

A (2) és a (3) képleteket egyszerűen megkaphatjuk az ismétléses kombinációkra vonatkozó képlettel [19], vagy egyszerű közvetlen bizonyítással [95].

n	$R(n)$	$E(n)$
39	13608507434599516007800	6804253717317430635800
40	53753604366668088230810	26876802183368505747610
41	212392290424395860814420	106196145212266853671620
42	839455243105945545123660	419727621553107337030440
43	3318776542511877736535400	1659388271256207997204920
44	13124252690842425594480900	6562126345421738821981380
45	51913710643776705684835560	25956855321889404891899640
46	205397724721029574666088520	102698862360516845690726160
47	812850570172585125274307760	406425285086296679352517680
48	3217533506933149454210801550	1608766753466582789006321550
49	12738806129490428451365214300	6369403064745230349484448700
50	50445672272782096667406248628	25222836136391079936354733752
51	199804427433372226016001220056	99902213716686176213303828904
52	791532924062974587678774064068	395766462031487417819020269060
53	3136262529306125724764953838760	1568131264653063110341743393432
54	12428892245768720464809261509160	6214446122884360719139487166608
55	49263609265046928387789436527216	24631804632523465167364431087664
56	195295022443578894680165266232892	97647511221789449252255283306556
57	774327632846470705223111406467256	387163816423235356435901003613848
58	3070609578529107968988200404956360	1535304789264553992010916827363440
59	12178349853827309571919303301013360	6089174926913654800993284900277200
60	48307454420181661301946569760686328	24153727210090830680539430271558520

5. táblázat. A szabályos és páros sorozatok száma $n = 39, \dots, 60$ esetén.

Ascher 1987-ben a következő explicit formula felhasználásával határozta meg a páros sorozatok $E(n)$ számát (pontosabban a mu torere nevű maori játék bizonyos állapotainak számát):

3.1. LEMMA. (Ascher [5], Sloane, Pfoffe [172]) *Ha $n \geq 1$, akkor a páros sorozatok $E(n)$ száma*

$$E(n) = \frac{1}{2} \left(\binom{2n-1}{n} + \binom{n-1}{\lfloor n/2 \rfloor} \right). \tag{4}$$

Bizonyítás. Lásd [5]. □

A (2) és (4) képletek felhasználásával meghatároztuk az $R(n)$ és $E(n)$ értékeket $n = 1, \dots, 100$ esetén. Az $n = 1, \dots, 38$ eredményeket [95]-ben publikáltuk, az $n = 39, \dots, 60$ értékek a 5. táblázatban találhatóak, míg a további értékek $n = 100$ -ig az OEIS-ben, valamint az azokat előállító program [127]-ben található.

A következő lemma miatt elegendő a nullamentes páros sorozatok ellenőrzése.

3.2. LEMMA. (Iványi, Lucz, Móri, Sótér [95]) *Ha $n \geq 2$, akkor az n -gráfok sorozatok $G(n)$ száma meghatározható az $(n - 1)$ -gráfok sorozatok $G(n - 1)$ és az*

n -gráfós nullamentes sorozatok $G_z(n)$ számából:

$$G(n) = G(n-1) + G_z(n),$$

és ha $n \geq 1$, akkor

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

Bizonyítás. Lásd [95]. □

A [95] cikkben található 12. és 13. lemmából azt kapjuk, hogy $R(n) = \Theta(4^n/\sqrt{n})$ és $E(n) = \Theta(4^n/\sqrt{n})$. A 14. következmény [95] azt állítja, hogy ha n tart a végtelenhez, akkor $E(n)/R(n)$ $1/2$ -hez tart. Szimulációs kísérleteink alapján úgy gondoljuk, hogy a következő állítás szintén igaz: $E_z(n) = \Theta(4^n/\sqrt{n})$, és ha n tart a végtelenhez, akkor $E_z(n)/R(n)$ $1/4$ -hez tart.

A [42, 95] cikkekben szereplő 22. tétel szerint azonban

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

ahol c és C pozitív konstansok.

A 2., a 3. és a 4. táblázatban lévő adatok szerint ha n nő, akkor a lineáris EGL, EGLJ és HHL-algoritmusok átlagos költsége csökken. A gráfós sorozatok ellenőrzésének ideje hosszabb, ezért ha n tart a végtelenhez, akkor a nemgráfós sorozatok ellenőrzésének átlagos ideje és a $G(n)/E_z(n)$ sorozat nullához tart.

Ezeket az eredményeket figyelembe véve azt kapjuk, hogy elegendő számunkra a szabályos sorozatok negyedének ellenőrzése. A 6. táblázatban található a nullamentes gráfós sorozatok száma, valamint a szabályos sorozatok számával elosztott nullamentes, szabályos sorozatok számával elosztott nullamentes gráfós és a szabályos sorozatok számával elosztott gráfós sorozatok száma. Burns eredményéből [42, 95] következik, hogy az utolsó két oszlopban található sorozatok nullához tartanak.

Az EGE párhuzamosított EGP változatát (lásd a következő fejezetben) felhasználva meghatároztuk $G(n)$ értékeit $n = 29$ -ig. Ezen értékek megtalálhatóak a [95]-ben lévő 2. táblázatban.

Megjegyezzük, hogy $G_z(n)$ megadja az izolált csúcsokat nem tartalmazó egyszerű gráfok számát. 2006-ban Gordon Royle [159] vetette fel a következő kérdést: igaz-e, hogy ha n tart a végtelenhez, akkor $G_z(n+1)/G_z(n)$ 4 -hez tart?

Hasonló kérdés vonatkozik a $G(n)/G(n+1)$ sorozat határértékére. Mindkét kérdést szimulációval vizsgáltuk a [90, Conjecture 12, Conjecture 13, Table 4] cikkben.

n	$G_z(n)$	$E_z(n)/R(n)$	$G_z(n)/R(n)$	$G(n)/R(n)$
1	0	0.000000	0.000000	1.000000
2	1	0.333333	0.333333	0.666667
3	2	0.200000	0.200000	0.400000
4	7	0.257143	0.200000	0.314286
5	20	0.222222	0.158730	0.246032
6	71	0.238095	0.153680	0.220779
7	240	0.230769	0.139860	0.199301
8	871	0.236053	0.135454	0.188500
9	3148	0.235294	0.129494	0.179391
10	11655	0.237524	0.126166	0.173375
11	43332	0.238095	0.122852	0.168260
12	162769	0.239188	0.120384	0.164278
13	614198	0.245769	0.118108	0.160821
14	2330537	0.240783	0.116188	0.157882
15	8875768	0.241379	0.114439	0.155271
16	33924859	0.241946	0.112880	0.152950
17	130038230	0.242424	0.111448	0.150844
18	499753855	0.242860	0.101137	0.148926
19	1924912894	0.243243	0.108920	0.147158
20	7429160296	0.243590	0.107789	0.145521
21	28 723 877 732	0.2439024	0.106729	0.143997
22	111 236 423 288		0.105733	0.142569
23	431 403 470 222		0.104793	0.141228
24	1 675 316 535 350		0.103903	0.139961
25	6 513 837; 679 610		0.103058	0.138762
26	25 354 842 100 894		0.102254	0.137625
27	98 794 053 269 694		0.101486	0.136542
28	385 312 558 571 890		0.100752	0.135509
29	1 504 105 116 253 904		0.100049	0.134521
30	5 876 236 938 019 300		0.100752	0.135509
31	22 974 847 474 172 374		0.100049	0.134521

6. táblázat. A csúcsok n és a nullamentes gráfok sorozatok $G_z(n)$ száma, illetve a nullamentes páros sorozatok $E_z(n)$ számának, a nullamentes gráfok sorozatok $G_z(n)$ számának, valamint a gráfok sorozatok $G(n)$ számának és a szabályos sorozatok $R(n)$ számának hányadosa $n = 1, \dots, 31$ csúcs esetén.

Megmutattuk, hogy a nullamentes gráfok sorozatok száma $n = 30$ csúcs esetén

$$G_z(30) = 5\,876\,236\,938\,019\,300,$$

míg $n = 31$ csúcs esetén

$$G_z(31) = 22\,974\,847\,474\,172\,374.$$

Tripathi és Vijay [95, 182, 6. lemma és 7. tétel] eredményeit felhasználva lényegesen csökkenthetjük a nullamentes páros sorozatok átlagos ellenőrzési idejét. Tripathi és Vijay [182] eredményével kapcsolatban beláttuk [95], hogy a tesztelt sorozatok egyenletes eloszlása esetén az ellenőrző pontok számának várható értéke körülbelül $n/2$.

A 3.3. lemma felhasználásával tovább gyorsíthatjuk az EGE-algoritmust. Ha $b = (b_1, \dots, b_n)$ egy szabályos sorozat, akkor $c = (c_1, \dots, c_n)$ *lexikografikusan i-kisebb, mint b*, ha

$$c_j = b_j \quad \text{for } j = 1, \dots, i,$$

és

$$\sum_{j=i+1}^n c_j < \sum_{j=i+1}^n b_j.$$

3.3. LEMMA. *Legyen i egész szám, melyre $1 \leq i \leq n$. Ha $b = (b_1, \dots, b_n)$ egy nemgráfok sorozat és $c = (c_1, \dots, c_n)$ lexikografikusan i -kisebb, mint b , akkor c szintén nemgráfok.*

Bizonyítás. Lásd [90]. □

A következőekben bemutatandó ERDŐS–GALLAI–ENUMERATING (EGE) az EGL egy leszámoló változata. Az algoritmus lexikografikus sorrendben megvizsgálja a nullamentes páros sorozatokat, ami lehetővé teszi a legtöbb alapvető művelet végrehajtását $O(1)$ átlagos időben.

- H_i (összegzett fokszámok): a legtöbb esetben a b sorozat utolsó eleme az egyetlen, amely változik, így ilyenkor elegendő a H utolsó értékét frissíteni a b utolsó értékének változása szerint.
- C_i (ellenőrző pontok): ha módosítjuk a sorozat i . elemét, akkor az előtte lévő pontok ugyanazok maradnak, tehát az összes ellenőrző pont előtt ugyanazok maradnak, így elegendő csak az i . index előtti elemet frissíteni és az összes többit utána.
- W_i (súlypontok): minden alkalommal, amikor az ellenőrző algoritmus megkap egy sorozatot ellenőrzésre, frissíti a súlypontokat, azonban sosem kezd 1-ről vagy n -ről. Azt az utolsó értéket használja, amelyet akkor használt, amikor a sorozat aktuális indexét ellenőrizte. Különböző súlypontokkal rendelkezik minden pontra minden egyes i index esetén, és csak eltolja ezeket az értékeket balra, vagy jobbra.

Tegyük fel, hogy n , b , H , c , C és W globális változók, így a **return** utasítás nem igényel többletidőt.

Az EGE fontos tulajdonsága, hogy a következő problémákat $\Theta(1)$ átlagos időben megoldja:

- egy nullamentes sorozat generálása;
- egy sorozathoz tartozó H összegzett fokszámok frissítése;
- egy sorozathoz tartozó C ellenőrző pontok frissítése;
- egy sorozathoz tartozó W súlypontok frissítése.

Noha az EGE a részproblémák többségét sorozatonként átlagosan $\Theta(1)$ időben megoldja, az ellenőrző pontoknál lévő munka több időt igényel, ezért a teljes futási idő $\Theta(E(n))$.

Az ERDŐS–GALLAI-ENUMERATING program [95] 8. tételén alapul.

3.1. Algoritmus. ERDŐS–GALLAI-ENUMERATING (n, G_z)

Bemenet. n : csúcsok száma ($n \geq 4$);

$b = b_1, \dots, b_n$: n -szabályos sorozat.

Kimenet. G_z : az n hosszú nullamentes gráfok sorozatok száma.

Munkaváltozók. i és j : ciklusváltozók;

$H = H_1, \dots, H_n$: H_i a tesztelt b sorozat első i elemének összege;

$W = W_1, \dots, W_n$: W_i az aktuális b_i -hez tartozó súlypont, b olyan elemeinek maximális indexe, amelyre igaz, hogy nem kisebbek, mint i ;

y : az aktuális b_i vágópontja, azaz i és w maximuma.

```

1. for  $i = 1$  to  $n$                                      // 1–8. sor: kezdeti értékek beállítása
2.    $b_i = n - 1$ 
3.    $H_i = i(n - 1)$ 
4.    $W_i = n$ 
5.    $C_i = 0$ 
6.  $G_z = 1$ 
7.  $c = 0$ 
8.  $b_{n+1} = -1$ 
9. while  $b_2 \geq 2$  or  $b_1 \geq 3$                          // 9. sor: utolsó sorozat volt?
10.   if  $b_n \geq 3$                                        // 10–14. sor: a következő sorozat előállításá
11.     NEW3( $n, b, H, c, C, W$ )
12.   else if  $b_n = 2$ 
13.     NEW2( $n, b, H, c, C, W$ )
14.   else NEW1( $n, b, H, c, C, W$ )

```

```

15.    CHECK( $n, b, H, c, W, L$ )           // 15. sor: paraméterek frissítése
16.     $G_z = G_z + L$                        // 16. sor:  $G_z$  növelése
17.    return  $G_z$                           // 17. sor: végeredmény

```

Az algoritmus négy eljárást használ. NEW1, NEW2, valamint NEW3 egy új sorozatot generálnak (ahol b_n 1, 2, illetve 3) és frissítik a fő paramétereket, míg a CHECK eldönti, hogy az aktuálisan megvizsgált sorozat gráfos-e vagy sem.

A CHECK eljárásban az EGLJ feltételét [125, (26) egyenlőség] használjuk fel.

CHECK(n, b, H, c, C, W)

```

1.  for  $i = 1$  to  $c$                        // 1-4. sor: ellenőrzőpontok tesztelése
2.     $y = \max(W_{C_i}, i)$                    // 2. sor: aktuális vágópont kiszámítása
3.    if  $H_i > i(y - 1) + H_n - H_y$        // 3-4. sor: EG tesztelés
4.    return 0
5.  return 1                               // 6. sor:  $b$  gráfos

```

NEW3(n, b, H, c, C, W)

```

1.   $b_n = b_n - 2$                            // 1-10. sor: új sorozat előállítás, ha  $b_n = 3$ 
2.   $H_n = H_n - 2$ 
3.  if  $b_n = b_{n-1} - 2$ 
4.     $c = c + 1$ 
5.     $C_c = n - 1$ 
6.     $W_{b_n} = W_{b_n} - 1$ 
7.  if  $b_n \leq b_{n-1}$ 
8.     $W_{b_{n+1}} = n + 1$ 
9.     $W_{b_n} = n + 1$ 
10. return  $H, c$ 

```

NEW2(n, b, H)

```

1.  if  $b_{n-1} = 2$                            // 1-53. sor: új sorozat előállítás, ha  $b_n = 2$ 
2.     $b_n = 1$                                // 1-9. sor: új sorozat előállítás, ha  $b_{n-1} = 2$ 
3.     $b_{n-1} = 1$ 
4.     $H_{n-1} = H_{n-1} - 1$ 
5.     $H_n = H_n - 2$ 
6.     $W_2 = n - 2$ 
7.    if  $b_{n-2} = 2$                        // 7-9. sor: új sorozat előállítás, ha  $b_{n-2} = 2$ 
8.       $c = c + 1$ 
9.       $C_c = n - 1$ 

```



```

10. else if  $b_{n-1} = 3$  // 10–16. sor: új sorozat előállítás, ha  $b_{n-1} = 3$ 
11.      $b_{n-1} = 2$ 
12.      $b_n = 1$ 
13.      $H_{n-1} = H_{n-1}$ 
14.      $H_n = H_n - 2$ 
15.      $W_3 = n - 2$ 
16.      $W_2 = n - 1$ 
17. else  $H_{n-1} = H_{n-1} - 1$ 
18.     if  $b_{n-2} = b_{n-1}$  and  $b_n$  páratlan
19.          $b_{n-1} = b_{n-1} - 1$ 
20.          $b_n = b_{n-1}$ 
21.          $H_n = H_n + b_{n-1} - b_n - 1$ 
22.          $C_c = C_c - 1$ 
23.          $W_{b_{n-2}} = n - 2$ 
24.         for  $i = 1$  to  $b_{n-2}$ 
25.              $W_i = n$ 
26.     if  $b_{n-2} = b_{n-1}$  and  $b_n - 1$  páros
27.          $b_{n-1} = b_{n-1} - 1$ 
28.          $b_n = b_{n-1} - 1$ 
29.          $H_n = H_n + b_{n-1} - b_n - 1$ 
30.          $C_c = C_c - 1$ 
31.          $c = c + 1$ 
32.          $C_c = n - 1$ 
33.          $W_{b_{n-2}} = n - 2$ 
34.          $W_{b_{n-1}} = n - 1$ 
35.         for  $i = 1$  to  $b_{n-2} - 2$ 
36.              $W_i = n$ 
37.     if  $b_{n-2} > b_{n-1}$  and  $b_{n-1}$  páros
38.          $b_{n-1} = b_{n-1} - 1$ 
39.          $b_n = b_{n-1}$ 
40.          $H_n = H_n + b_{n-1} - b_n - 1$ 
41.          $c = c - 1$ 
42.          $W_{b_{n-2}-1} = n - 2$ 
43.          $W_{b_{n-2}-1} = n - 1$ 
44.         for  $i = 1$  to  $b_{n-1} - 1$ 
45.              $W_i = n$ 
46.     if  $b_{n-2} > b_{n-1}$  and  $b_n - 1$  páros
47.          $b_{n-1} = b_{n-1} - 1$ 

```

```

48.           $b_n = b_{n-1} - 1$ 
49.           $H_n = H_n + b_{n-1} - b_n - 1$ 
50.           $W_{b_{n-1}+1} = n - 1$ 
51.          for  $i = 1$  to  $b_{n-1} - 1$ 
52.               $W_i = n$ 
53. return  $H, c, C, W$ 

```

NEW1 hasonló NEW2-höz (bár bonyolultabb, lásd GENERATE-NEW-SEQUENCE a következő részben), ezért azt itt nem részletezzük.

4. A feladat részekre osztása (szeletelés a szabályos sorozatok száma alapján)

A leszámolás párhuzamos megoldásának fontos része a feladat kisebb részekre osztása. Ezt *szeletelésnek* nevezzük.

Felosztási módszereink alapja a következő *szabályos* (l, u, m) -lemma.

Legyenek l , u és m egész számok, továbbá m legyen pozitív. Ekkor az

$$l \leq a_1 \leq \dots \leq a_m \leq u$$

feltételnek eleget tevő $\mathbf{a} = a_1, \dots, a_m$ sorozatokat *nemcsökkenő* (l, u, m) -szabályos, míg az

$$l \leq a_m \leq \dots \leq a_1 \leq u$$

feltételnek eleget tevő sorozatokat *nemnövekvő* (l, u, m) -szabályos sorozatoknak nevezzük.

4.1. LEMMA. (Iványi, Lucz, Móri, Sótér [95, (21) és (22)]) *Legyenek l , u és m egész számok, továbbá m legyen pozitív. Ekkor az*

$$l \leq a_1 \leq \dots \leq a_m \leq u$$

feltételnek eleget tevő (l, u, m) -szabályos sorozatok száma

$$R(l, u, m) = \binom{u-l+m}{m}. \quad (5)$$

Bizonyítás. Jelöljük \mathbf{b} -vel a $b_i = a_i + i - 1$ előírással képzett b_1, \dots, b_m sorozatot. A \mathbf{b} sorozatok száma megegyezik az $\mathbf{a} = a_1, \dots, a_m$ sorozatok számával. Másrészt a \mathbf{b} sorozatok száma annyi, ahány módon $2m-1$ különböző elem közül kiválaszthatunk m elemet, azaz az (5) egyenlőségben szereplő binomiális együttható valóban megadja a keresett számosságot. \square

1	1	1	1
1	2	3	4
1	3	6	10
1	4	10	20
1	5	15	35

7. táblázat. A felosztáshoz használt mátrix $n = 5$ esetén.

2	2	2	2	2
3	2	2	2	2
3	3	3	3	3
4	2	2	2	2
4	3	3	3	3
4	4	3	3	3
4	4	4	4	4

8. táblázat. $n = 5$ -ös felosztás határsorozatai.

Az (5) képlet segítségével kiszámítható, hány olyan sorozat van, amelynek hossza, valamint első és utolsó eleme adott. Ezt felhasználva megadható egy mátrix, amely megadja, hány olyan sorozat van, amely az f értékkel kezdődik és a g értékkel végződik. Ezt a táblázatot felhasználva a következő módon adható meg a felosztás:

1. válasszuk meg a maximális szeletméretet (ezt a konkrét számítások során úgy választottuk meg, hogy esetünkben ez akkora szeletet jelentett, amelyek egy éjszaka alatt kiszámíthatóak);
2. induljunk el a mátrix alsó sorának első elemétől, és kezdjük a mátrix további sorait kiolvasni és összegezni mindaddig, amíg az aktuális kapacitás vagy a kiolvasandó értékek el nem fogytak;
3. ha egy érték túl nagy az adott maximális mérethez viszonyítva, akkor lépünk egy sorral feljebb, és kezdjük el azt a sort kiolvasni addig az oszlopig, ahonnan felléptünk;
4. folytassuk ezt az algoritmust, amíg az utolsó sor végére nem értünk.

A korábban ismertetett algoritmus szerint $n = 5$ -re a 7. táblázatból kiolvasható a 8. táblázatban látható felosztás. A táblázatban látható sorozatok a következő

módon értendők: az első szelet az 1; 1; 1; 1; 1 sorozattól a 2; 2; 2; 1; 1 sorozatig tart, a második pedig a 2; 2; 2; 2; 2 sorozattól a 3; 2; 2; 1; 1 sorozatig és így tovább. Ez a módszer ugyan nem garantálja, hogy a részfeladatok hossza pontosan ugyanakkora legyen, azonban megszünteti a korábbiakban tapasztalt „mamut” hatást.

Ezt a módszert használtuk fel $G_z(29)$ kiszámításakor. A teljes számítást összesen több mint 15 000 részfeladatra felosztva végeztük. A teljes felosztás generálásának első lépése a 3. táblázathoz hasonló mátrix előállítás a megfelelő n értékhez. A mátrix feltöltését végzi el az alábbi GENERATE-MATRIX program.

4.1. Algoritmus. GENERATE-MATRIX ($n, MaxSize$)

Bemenet. n : a csúcsok száma ($n \geq 4$);

MaxSize: a maximális megengedett szeletméret

(nullmentes szabályos sorozatok maximális száma a szeletben).

Kimenet. A képernyőre írjuk a szeleteket határoló sorozatokat.

```

1. for  $j = n - 1$  downto 1           // 1-2. sor: a mátrix első sorának kitöltése
2.    $M_{1j} = 1$ 
3. for  $i = n$  downto 2               // 3-5. sor: a mátrix feltöltése
4.   for  $j = 1$  to  $n$ 
5.      $M_{i,j} = \binom{i+j-2}{i-1}$ 
6. GENSEQUENCES( $M, n, n, 1, n - 1, MaxSize, 0$ ) // 6. sor: szelet generálása
```

Miután megvan a mátrixunk, már csak ki kell olvasnunk belőle a szeletek határsorozatokat (a 8. táblázathoz hasonló alakban) az alábbi GENERATE-SEQUENCES algoritmus segítségével.

4.2. Algoritmus. GENERATE-SEQUENCES($M, n, i, j, j_m, MaxSize, J$)

Bemenet. M : a felosztáshoz tartozó mátrix, amit a GENERATE-MATRIX algoritmussal kaptunk;

n : a csúcsok száma ($n \geq 1$);

i, j, J : segédparaméterek;

MaxSize: a maximális megengedett szeletméret (nullmentes szabályos sorozatok maximális száma a szeletben).

Kimenet. A szeleteket határoló sorozatok (rekurzívan).

```

1.  $C = 0$                                // 1. sor: a szelet kezdő méretének beállítása
2. while  $j < j_m + 1$ 
3.   if  $C + M_{ij} \leq MaxSize$            // 3. sor: ha bővíthető a szelet
4.      $C = C + M_{i,j}$                        // 4. sor: bővítés
5.     if  $j \leq j_m$                        // 5-6. sor: tovább lépünk a mátrixban
```

```

6.            $j = j + 1$ 
7.   else                                     // 7. sor: nem bővíthető a szelet
8.       if  $C \neq 0$                              // 8. sor: a szelet nem üres
9.           for  $k = 2$  to  $size(J, 2)$            // 9–14. sor: kiírás
10.               $print J_k$ 
11.           for  $k = 2$  to  $size(J, 2)$ 
12.               $print j - 1$ 
13.               $print newline$                    // 13. sor: sortörés
14.               $C = 0$ 
15.           if  $M_{i,j} > MaxSize$  és  $j \leq j_m$  // 15. sor: felbonthatóság ellenőrzése
16.              GENERATE-SEQUENCES( $M, n, i - 1, 1, j, MaxSize, [J, j]$ )
17.               $j = j + 1$ 
18.   if  $C \neq 0$                                // 18. sor: utolsó szelet nem üres
19.       for  $k = 2$  to  $size(J, 2)$              // 19–23. sor: utolsó szelet kiírása
20.           $print(J_k)$ 
21.       for  $k = 1$  to  $n - size(J, 2) - 1$ 
22.           $print(J, size(J, 2))$ 
23.        $print newline$ 

```

5. A feladat részekre osztása (szeletelés a páros sorozatok száma alapján)

Az $n \geq 32$ esetben a következő tételre – melyet *páros* (l, u, m)-tételnek (röviden: páros tételnek) nevezünk – alapuló felosztási módszert alkalmaztunk.

Legyenek l , u és m egész számok, melyekre $u \geq l$ és $m \geq 1$. Legyen $\mathbf{a} = a_1, \dots, a_m$ egész számok olyan sorozata, amelyre teljesül

$$u \geq a_m \geq \dots \geq a_1 \geq l \quad (6)$$

és

$$a_1 + \dots + a_m \text{ páros.} \quad (7)$$

Jelöljük $E(l, u, m)$ -mel a (6) és (7) feltételeknek eleget tevő sorozatok számát. Ascher tételének [5] következő általánosítása megadja $E(l, u, m)$ -et.

5.1. TÉTEL. *Ha l , u és m olyan egész számok, melyekre $u \geq l$ és $m \geq 1$, továbbá $p = 1$, ha $l(u - l + 1)$ páratlan, és $p = 0$ egyébként, akkor*

$$E(l, u, m) = \sum_{i=0}^{\lfloor m/2 \rfloor + p} \binom{s+2i}{2i} \cdot \binom{s+m-2i}{m-2i}, \quad (8)$$

ahol $s = \lfloor u - l + 1 \rfloor / 2 + p$.

Bizonyítás. a) Először tegyük fel, hogy $u - l + 1$ páros. Ekkor $p = 0$ és az $[u, l]$ intervallum $(u-l+1)/2$ páratlan és $(u-l+1)/2$ páros számot tartalmaz. A páratlan számok közül páros sokat – azaz $0, 2, \dots, (u-l+1)/2$ elemet kell kiválasztanunk. A még hiányzó elemeket pedig a páros elemek közül kell kiválasztanunk.

Mivel n elem k -ad osztályú ismétléses variációinak száma $\binom{n+k-1}{k}$, ezért ebben az esetben az

$$E(l, u, m) = \sum_{i=0}^{\lfloor m/2 \rfloor} \binom{(u-l+1)/2 + 2i}{2i} \cdot \binom{(u-l+1)/2 + m - 2i}{m - 2i}$$

eredményt kapjuk, ami a $p = 0$ esetben megfelel (8)-nak.

b) Most tegyük fel, hogy mind l , mind $u - l + 1$ páratlan. Ekkor $p = 1$. Ekkor az $[l, u]$ intervallumban páratlan számú páratlan elem van, amelyek közül most is páros számút kell kiválasztani, ezért ebben az esetben az

$$E(l, u, m) = \sum_{i=0}^{\lfloor m/2 \rfloor + 1} \binom{(u-l+1)/2 + p + 2i}{2i} \cdot \binom{(u-l+1)/2 + p + m - 2i}{m - 2i}$$

eredményt kapjuk, ami a $p = 1$ esetben megfelel (8)-nak. \square

A 5.1. tételnek mind a páros sorozatok $E(n)$ számát megadó páros lemma [90, Lemma 6], mind pedig a nullamentes páros sorozatok $E_z(n)$ számát megadó nullamentes páros lemma [90, Lemma 3] speciális esete. Ehhez érdemes hozzátenni, hogy a következőkben szereplő képletek egyszerűbbek, mint a korábban publikált lemmákban lévők: négy képlet helyett csak egyet kell használni. Ez annak köszönhető, hogy most közvetlenül a foksorozatokat vizsgáltuk a monotonitást biztosító transzformáció nélkül.

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

$$E(n) = \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{\lfloor n/2 \rfloor - 1 + 2i}{2i} \cdot \binom{\lfloor n/2 \rfloor - 1 + n - 2i}{n - 2i}. \quad (9)$$

Bizonyítás. A 5.1. tételben végezzük el az $l = 0$, $u = n - 1$, $p = 0$ és $m = n$ helyettesítést. \square

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

$$E_z(n) = \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{\lfloor (n-1)/2 \rfloor - 1 + 2i}{2i} \cdot \binom{\lceil (n-1)/2 \rceil - 1 + n - 2i}{n - 2i}. \quad (10)$$

Bizonyítás. A 5.1. tételben végezzük el az $l = 1$, $u = n - 1$ és $m = n$ helyettesítést. \square

Tekintsünk néhány példát.

1. Először számítsuk ki $E(4)$ -et. (9) szerint

$$E(4) = \binom{1}{0} \cdot \binom{5}{4} + \binom{3}{2} \cdot \binom{3}{2} + \binom{5}{4} \cdot \binom{1}{0} = 1 \cdot 5 + 3 \cdot 3 + 5 \cdot 1 = 19.$$

2. Számítsuk ki $E_z(4)$ -et. (10) szerint

$$E_z(4) = \binom{4}{0} \cdot \binom{4}{4} + \binom{3}{2} \cdot \binom{2}{2} + \binom{5}{4} \cdot \binom{0}{0} = 1 \cdot 1 + 3 \cdot 1 + 5 \cdot 1 = 9.$$

3. Számítsuk ki $E(5)$ -öt (9) szerint

$$E(5) = \binom{1}{0} \cdot \binom{7}{2} + \binom{3}{2} \cdot \binom{5}{2} + \binom{5}{4} \cdot \binom{3}{1} = 1 \cdot 21 + 3 \cdot 10 + 5 \cdot 3 = 66.$$

4. Számítsuk ki $E(6)$ -ot.

$$\begin{aligned} E(6) &= \binom{2}{0} \cdot \binom{8}{2} + \binom{4}{2} \cdot \binom{6}{2} + \binom{6}{2} \cdot \binom{4}{2} + \binom{8}{2} \cdot \binom{2}{0} = \\ &= 28 + 90 + 90 + 28 = 236 = 1 \cdot 21 + 3 \cdot 10 + 5 \cdot 3 = 66. \end{aligned}$$

A 9. táblázat tartalmazza az $R(n)/R(n+1)$, $R_z(n)/R_z(n+1)$, $E(n)/R(n)$, $E(n)/E(n+1)$, $E_z(n)/E_z(n+1)$ és $E_z(n)/R_z(n)$ hányadosokat $n = 1, \dots, 32$ csúcs esetén.

$R(n)$ értékét a (2), az $R_z(n)$ értékét pedig a (3) egyenlőségek, $E(n)$ értékét az 5.1., míg $E_z(n)$ értékét az 5.2. következmény alapján számítottuk.

Érdemes megjegyezni, hogy $R(101)/R(102)$ és $R_z(101)/R_z(102)$ első kilenc decimális számjegye megegyezik.

A 9. táblázatban lévő adatok azt mutatják, hogy ha $1 \leq n \leq 32$, és n páratlan, akkor $E_z(n)/R_z(n) = 0, 5$. Ez a tulajdonság n nagyobb értékei esetén is jellemző a hányadosra.

- 5.1. LEMMA. Ha $1 \leq k \leq 600$, akkor

$$\frac{E_z(2k-1)}{R_z(2k-1)} = 0, 5.$$

Bizonyítás. Lásd $R_z(n)$ és $E_z(n)$ pontos értékeit [128]. □

A 10. táblázat az $E_z(n)/G_z(n)$ hányadosokat tartalmazza $n = 1, \dots, 31$ csúcs esetén, míg a 12. táblázat a $G_z(n)/T(n)$ hányadosokat $n = 30$ és $n = 31$ csúcs esetén.

n	$\frac{R(n)}{R(n+1)}$	$\frac{R_z(n)}{R_z(n+1)}$	$\frac{E(n)}{R(n)}$	$\frac{E(n)}{E(n+1)}$	$\frac{E_z(n)}{E_z(n+1)}$	$\frac{E_z(n)}{R_z(n)}$
1	0.333333	0.000000	1.000000	0.000000	0.000000	-- --
2	0.300000	0.250000	0.666667	0.500000	0.500000	1.000000
3	0.287714	0.266667	0.600000	0.222220	0.222222	0.500000
4	0.277778	0.257857	0.487179	0.321427	0.321429	0.600000
5	0.270562	0.266667	0.523810	0.254545	0.254545	0.500000
6	0.269231	0.265151	0.510823	0.277778	0.277778	0.523810
7	0.266667	0.263736	0.505828	0.260698	0.260698	0.500000
8	0.264706	0.262500	0.502720	0.265559	0.265559	0.505828
9	0.263158	0.261437	0.501440	0.260687	0.260687	0.500000
10	0.261905	0.260526	0.500682	0.261276	0.261276	0.501440
11	0.260870	0.259740	0.500357	0.259555	0.259555	0.500000
12	0.260000	0.259058	0.500171	0.259243	0.259243	0.500357
13	0.259259	0.258461	0.500089	0.258415	0.258416	0.500000
14	0.258621	0.257937	0.500043	0.257982	0.257982	0.500089
15	0.258065	0.257471	0.500022	0.257460	0.257460	0.500000
16	0.257578	0.257056	0.500011	0.257068	0.257068	0.500022
17	0.257143	0.256684	0.500005	0.256682	0.256682	0.500000
18	0.256757	0.256349	0.500003	0.256352	0.256352	0.500006
19	0.256410	0.256046	0.500001	0.256045	0.256045	0.500000
20	0.256098	0.255769	0.500001	0.255770	0.255770	0.500001
21	0.255814	0.255517	0.50000034	0.255517	0.255517	0.500000
22	0.255556	0.255285	0.50000016	0.255286	0.255286	0.50000034
23	0.255319	0.255072	0.50000009	0.255072	0.255072	0.50000000
24	0.255102	0.254876	0.50000004	0.254876	0.254876	0.50000000
25	0.254902	0.254694	0.50000002	0.254694	0.254694	0.50000009
26	0.254717	0.254525	0.50000001	0.254525	0.254525	0.50000000
27	0.254545	0.254368	0.50000001	0.254368	0.254368	0.50000000
28	0.254386	0.254221	0.50000000	0.254221	0.254221	0.50000000
29	0.254237	0.254083	0.50000000	0.254083	0.254083	0.50000000
30	0.254098	0.253854	0.50000000	0.253955	0.253955	0.50000000
31	0.253968	0.253834	0.50000000	0.253834	0.253834	0.50000000
32	0.253846	0.253720	0.50000000	0.253720	0.253720	0.50000000

9. táblázat. Az $R(n)/R(n+1)$, $R_z(n)/R_z(n+1)$, $E(n)/R(n)$, $E(n)/E(n+1)$, $E_z(n)/E_z(n+1)$ és $E_z(n)/R_z(n)$ hányadosok $n = 1, \dots, 32$ csúcs esetén.

A 10. táblázat adatai azt mutatják, hogy a $G_z(n)/E_z(n)$ sorozat csökkenő. Feltesszük, hogy a sorozat monoton csökkenve nullához tart, ha n tart a végtelenhez (hasonlóan ahhoz, ahogy a $G(n)/E(n)$ sorozat tart nullához [95, page 260, Corollary 23]).

A 11. táblázat tartalmazza $G_z(n)$ és $G_n(n)$ értékét $n = 1, \dots, 30$ esetén, valamint $G_z(n+1)/G_z(n)$ és $G(n+1)/G(n)$ értékét $n = 1, \dots, 31$ esetén.

A 12. táblázat tartalmazza a $G_z(n)$, $T(n)$ és $G_z(n)/T(n)$ értékeit $n = 30$ és $n = 31$ csúcs esetén: a gráfos és tesztelt sorozatok számának aránya lényegesen

n	$G_z(n)$	$E_z(n)$	$G_z(n)/E_z(n)$
01	0	0	---
02	1	1	1.000000
03	2	2	1.000000
04	7	9	0.777778
05	20	28	0.714286
06	71	110	0.645455
07	240	396	0.606061
08	871	1 519	0.573404
09	3 148	5 720	0.550350
10	11 655	21 942	0.531173
11	43 332	83 980	0.515980
12	162 769	323 554	0.503149
13	614 198	1 248 072	0.492117
14	2 330 537	4 829 708	0.482542
15	8 875 768	18 721 080	0.474106
16	33 924 859	72 714 555	0.466548
17	130 038 230	282 861 360	0.459724
18	499 753 855	1 101 992 870	0.453500
19	1 924 912 894	4 298 748 300	0.447784
20	7 429 160 296	16 789 046 494	0.442500
21	28 723 877 732	65 641 204 200	0.437589
22	111 236 423 288	256 895 980 068	0.433002
23	431 403 470 222	1 006 308 200 040	0.428699
24	1 675 316 535 350	3 945 186 233 014	0.424648
25	6 513 837, 679 610	15 478 849 767 888	0.420821
26	25 354 842 100 894	60 774 332 618 300	0.417197
27	98 794 053 269 694	238 775 589 937 976	0.413752
28	385 312 558 571 890	938 702 947 395 204	0.410473
29	1 504 105 116 253 904	3 692 471 324 505 040	0.407344
30	5 876 236 938 019 300	14 532 512 180 224 216	0.404351
31	22 974 847 474 172 100	57 224 797 531 384 560	0.401484
32	---	225 441 858 758 287, 187	---
33	---	888 545 038 032 771 168	---
34	---	3 503 546 152 385 412 870	---
35	---	13 820 048 716 545 422 988	---
36	---	54 534 996 163 146 555 910	---

10. táblázat. $G_z(n)$, $E_z(n)$ és $G_z(n)/E_z(n)$ $n = 1, \dots, 31$ csúcs esetén.

magasabb, mint kisebb n -ek esetén, és ez a sorozat növekvő. A drasztikus változás annak köszönhető, hogy EGE2 sok nemgráfes sorozat tesztelését átugorja.

A 13. táblázat negyedik oszlopa alátámasztja Royle Gordon következő sejtését.

5.1. SEJTÉS. (Royle, 2012 [159]). *Ha n tart a végtelenbe, akkor a $G_z(n + 1)/G_z(n + 1)$ sorozat tart a 4-hez.*

n	$G_z(n)$	$G(n)$	$\frac{G_z(n+1)}{G_z(n)}$	$\frac{G(n+1)}{G(n)}$
1	0	1	---	0.500000
2	1	2	2.000000	2.000000
3	2	4	3.500000	3.750000
4	7	11	2.857143	2.818182
5	20	31	3.550000	3.290323
6	71	102	3.380282	3.352941
7	240	342	3.629167	3.546784
8	871	1213	3.614237	3.595218
9	3148	4361	3.702351	3.672552
10	11655	16016	3.717889	3.705544
11	43332	59348	3.756323	3.742620
12	162769	222117	3.773434	3.786674
13	614198	836315	3.794439	3.802710
14	2330537	3166852	3.808465	3.817067
15	8875768	12042620	3.822189	3.828918
16	33924859	45967479	3.833125	3.839418
17	130038230	176005709	3.843130	3.848517
18	499753855	675759564	3.851172	3.856630
19	1924912894	2600672458	3.859479	3.863844
20	7429160296	10029832754	3.866369	3.870343
21	28723877732	38753710486	3.872612	3.876212
22	111236423288	149990133774	3.878257	3.881553
23	431403470222	581393603996	3.883410	3.886431
24	1675316535350	2256710139346	3.888124	3.890907
25	6513837679610	8770547818956	3.894458	3.895031
26	25354842100894	34125389919850	3.895503	3.897978
27	98794053269694	132919443189544	3.900159	3.898843
28	385312558571890	518232001761434	3.903597	3.902238
29	1504105116253904	2022337118015338	3.906814	3.905666
30	5876236938019300	7898574056034638	3.909789	3.908734
31	22974847474172100	30873429530206738	---	---

11. táblázat. $G_z(n)$, $G(n)$, $G_z(n+1)/G_z(n)$ és $G(n+1)/G(n)$ értéke $n = 1, \dots, 31$ csúcs esetén.

n	$G_z(n)$	$T(n)$	$G_z(n)/T(n)$
31	5 876 236 938 019 300	6 790 865 476 867 340	86,531487
32	22 974 847 471 172 100	26 507 499 250 791 700	86,673010

12. táblázat. $G_z(n)$, $T(n)$ és $G_z(n)/T(n)$ $n = 30$ és $n = 31$ csúcs esetén.

Azt gondoljuk, hogy a következő sejtés [90] is igaz.

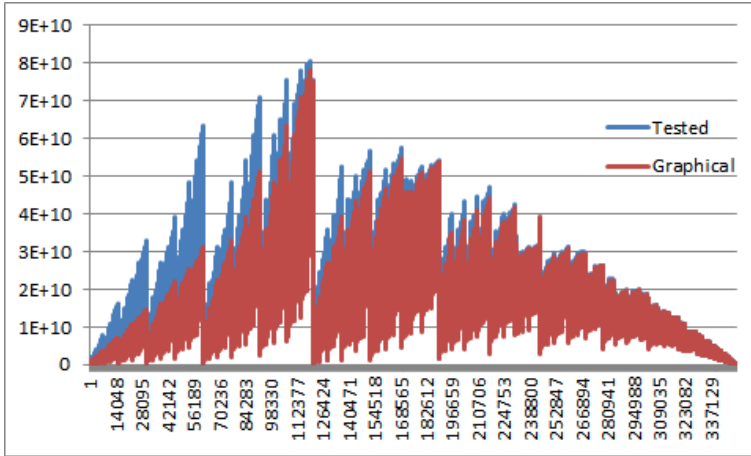
5.2. SEJTÉS. *Ha n tart a végtelenbe, akkor $G(n+1)/G(n)$ tart a 4-hez.*

A 12. táblázat szerint az $n = 30$ esetben a tesztelt potenciális gráfos sorozatok 85.4 százaléka, míg az $n = 31$ esetben a sorozatok 86.67 százaléka gráfos. Ezért a futási idő tovább csökkenthető, ha a lineáris időt igényelő tesztelés nélkül, konstans idő alatt meg tudjuk állapítani, hogy a vizsgált sorozat gráfos.

n	$G_z(n)$	$G(n)$	$\frac{G_z(n+1)}{G_z(n)}$	$\frac{G(n+1)}{G(n)}$
1	0	1	0.000000	0.500000
2	1	2	2.000000	2.000000
3	2	4	3.500000	3.750000
4	7	11	2.857143	2.818182
5	20	31	3.550000	3.290323
6	71	102	3.380282	3.352941
7	240	342	3.629167	3.546784
8	871	1 213	3.614237	3.595218
9	3 148	4 361	3.702351	3.672552
10	11 655	16 016	3.717889	3.705544
11	43 332	59 348	3.756323	3.742620
12	162 769	222 117	3.773434	3.786674
13	614 198	836 315	3.794439	3.802710
14	2 330 537	3 166 852	3.808465	3.817067
15	8 875 768	12 042 620	3.822189	3.828918
16	33 924 859	45 967 479	3.833125	3.839418
17	130 038 230	176 005 709	3.843130	3.848517
18	499 753 855	675 759 564	3.851172	3.856630
19	1 924 912 894	2 600 672 458	3.859479	3.863844
20	7 429 160 296	10 029 832 754	3.866369	3.870343
21	28 723 877 732	38 753 710 486	3.872612	3.876212
22	111 236 423 288	149 990 133 774	3.878257	3.881553
23	431 403 470 222	581 393 603 996	3.883410	3.886431
24	1 675 316 535 350	2 256 710 139 346	3.888124	3.890907
25	6 513 837, 679 610	8 770 547 818 956	3.894458	3.895031
26	25 354 842 100 894	34 125 389 919 850	3.895503	3.897978
27	98 794 053 269 694	132 919 443 189 544	3.900159	3.898843
28	385 312 558 571 890	518 232 001 761 434	3.903597	3.902238
29	1 504 105 116 253 904	2 022 337 118 015 338	3.906814	3.905666
30	5 876 236 938 019 300	7 898 574 056 034 638	3.909789	3.908734
31	22 974 847 474 172 100	30 873 429 530 206 738	— — —	— — —

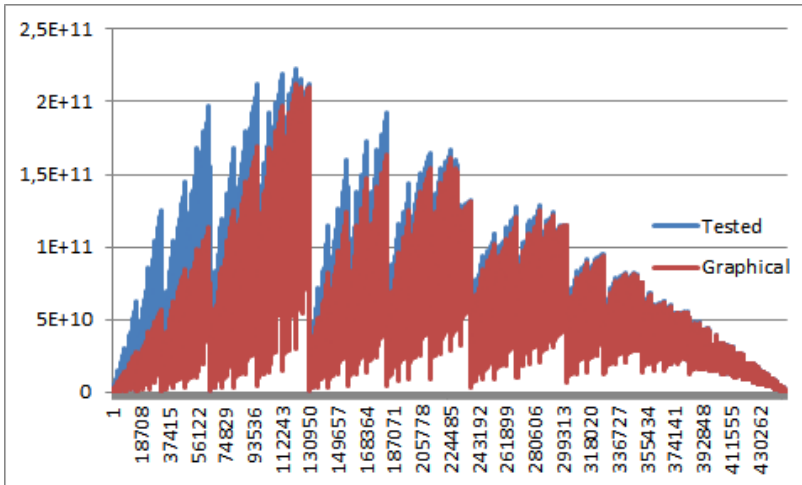
13. táblázat. Nullamentes gráfos sorozatok $G_z(n)$ száma és gráfos sorozatok $G(n)$ száma, továbbá a $G_z(n)/G_z(n+1)$ és $G(n)/G(n+1)$ hányadok $n = 1, \dots, 30$ csúcs esetén.

A 4. ábra a tesztelt és a gráfok sorozatok számát mutatja a szeletek sorszáma függvényében $n = 30$ csúcs esetén.



4. ábra. A tesztelt és a gráfok sorozatok száma a szeletek sorszáma függvényében $n = 30$ csúcs esetén.

Az 5. ábra a hasonló adatokat mutatja $n = 31$ csúcs alapján.



5. ábra. A tesztelt és a gráfok sorozatok száma a szeletek sorszáma függvényében $n = 31$ csúcs esetén.

Megjegyezzük, hogy a folyóirat honlapján a 4. és 5. ábrák színesek (a grafikus sorozatokat piros, míg a tesztelt sorozatokat kék szín jelzi).

6. Párhuzamos Erdős–Gallai-algoritmus (EGP)

A $G(n)$ értékek kiszámítása hosszú ideig tart, ha szekvenciális programot használunk, ezért az EGE gyorsított párhuzamos változatát használtuk fel. A felhasznált processzorok és a $G_z(n)$ kiszámításához szükséges idő megközelítőleg fordítottan arányos, így minél több processzort használunk fel, annál kevesebb időre lesz szükségünk.

A lineáris algoritmusunk használhatósága érdekében szükség van egy algoritmusra, amely képes az ellenőrzendő sorozatok egy részén dolgozni.

Az ERDŐS–GALLAI-PARALLEL algoritmus felhasználásával kiszámítottuk ezeket a számokat $n = 31$ -ig. Az értékek [95] 2. táblázatában, valamint az OEIS-ben [96] találhatóak.

Alkalmazásunk két részre bontható: egy szerverre és egy kliensre. A szerver tárolja a kliensek közti feladatok felosztásához szükséges információkat, valamint összegyűjti az eredményeket. A kliens tárolja a szerver IP címét és portját az új feladatok kéréséhez.

A párhuzamos algoritmus egyik legkritikusabb része a probléma közel azonos méretű részfeladatokra osztása. A következő egyenlet megadja, hogy közelítőleg hány azonos fejjel kezdődő sorozatok léteznek. Ezen számok ismeretében korlátozott méretű feladatok generálására vagyunk képesek, más szóval egyik feladat sem lesz nagyobb egy megadott maximumnál.

Könnyen belátható [95, (22) egyenlőség], hogy az (l, u, m) -szabályos sorozatok $Q(l, u, m)$ száma

$$R(l, u, m) = \binom{u-l+m}{m}. \quad (11)$$

(11) alapján készítettük el a feladatokat generáló algoritmust.

6.1. Algoritmus. GENERATE-MATRIX(n, ms, M)

Bemenet. n : a sorozat hossza;

ms : egy feladat maximális mérete.

Kimenet. M : a feladatok paramétereit tartalmazó mátrix.

Munkaváltozók. i, j ciklusváltozók.

1. **for** $i = n$ **downto** 2 // 1–3. sor: a mátrix kitöltése
2. **for** $j = 1$ **to** $n - 1$
3. $M_{i,j} = \binom{i+j-2}{i-1}$
4. **for** $j = n - 1$ **downto** 1 // 4–5. sor: a mátrix első sorának kitöltése

```

5.      $M_{1,j} = 1$ 
6.  GENERATE-NEW-SEQUENCES( $M, n, n, 1, n - 1, ms, 0$ )
                                           // 6. sor: új feladat előállítás
7.  return  $M$                                // 7. sor: eredmény visszaadása

```

Az algoritmus egy, az egyenlőség felhasználásával kapott értékekkel feltöltött mátrixot ad vissza. Ezek után képesek leszünk a sorozatok generálására. Induljunk el a mátrix alsó sorának első elemétől, és kezdjük el az értékeket kiolvasni és összegezni. Abban az esetben, amennyiben az érték túl nagy a maximális mérethez viszonyítva, lépünk feljebb egy sorral, és kezdjük el kiolvasni és összegezni az elemeket az első sortól kezdve addig az oszlopig, ahonnan felléptünk. Addig folytassuk az algoritmust, amíg el nem érjük a szükséges méretet. A következő (rekurzív) algoritmus a fent említett eljárás szerint működik.

6.2. *Algoritmus.* GENERATE-NEW-SEQUENCE(M, n, i, j, j_m, ms, J)

Bemenet. n : a sorozat hossza;

ms : egy feladat maximális mérete.

Kimenet. M : a feladatok paramétereit tartalmazó mátrix.

Munkaváltozók. i, j : ciklusváltozók.

```

1.   $S = 0$                                      // 1. sor: aktuális szelet méretének beállítása
2.  while  $j < j_m + 1$ 
3.      if  $S + M_{i,j} \leq ms$                  // 3. sor: ha további sorozatot tudunk hozzáadni
4.           $S = S + M_{i,j}$                      // 4. sor: további sorozatot adunk a szelethez
5.          if  $j \leq j_m$                      // 5-6. sor: átlépés a mátrix következő oszlopához
6.               $j = j + 1$ 
7.          else if  $S \neq 0$                    // 7. sor: szelet nem üres
8.              for  $k = 2$  to  $size(J, 2)$       // 8-13. sor: nyomtatás
9.                   $print(J_k)$ 
10.             for  $k = 1$  to  $n - size(J, 2) + 1$ 
11.                  $print(j - 1)$ 
12.              $print$  newline                 // 13. sor: új sor
13.              $S = 0$ 
14.             if  $M_{i,j} > ms$  and  $j \leq j_m$  // 14. sor: ha felbonthatatlan
15.                 GENERATE-NEW-SEQUENCE( $M, n, i - 1, 1, j, ms, [J, j]$ )
16.                  $j = j + 1$ 
17. if  $S \neq 0$                                // 17. sor: utolsó szelet nem üres
18.     for  $k = 2$  to  $size(J, 2)$              // 18-22. utolsó szelet nyomtatása
19.          $print(J_k)$ 
20.     for  $k = 1$  to  $n - size(J, 2) + 1$ 
21.          $print(J(size(J, 2)))$ 
22.      $print$  newline

```

Most már rendelkezünk a probléma kisebb méretű részfeladataival. Ezt követően elkezdhető a szerver program segítségével a különböző számítógépek közötti szétosztásuk. A DISTRIBUTING-JOBS algoritmus megmutatja, hogy a szerver mi-képp küldi el a feladatokat a klienseknek. Az algoritmusban csak a feladatok szétosztására koncentráltunk, így nem tartalmazza a hálózati kommunikációval foglalkozó kódot, kivéve a nagyon fontos hálózati primitíveket (számítógépes háló-zatokról bővebben olvashatunk [178]-ban).

6.3. Algoritmus. DISTRIBUTING-JOBS(n, N, M, G_z)

Bemenet. n : a sorozat hossza;

N : a feladatok becsült száma;

M : a feladatok paramétereit tartalmazó mátrix.

Kimenet. G_z : az n -szabályos nullamentes gráfos sorozatok száma.

Munkaváltozók. $S = S_0, \dots, S_n$: a feladatok állapotát tároló vektor;

fj : befejezett feladatok száma;

aj : a kliensnek elküldött utolsó feladat száma;

ji : a bejövő eredmény feladatának indexe;

cl : kliens azonosító (hálózati kommunikációhoz szükséges);

msg : klientsől jövő üzenet (csak hálózati kommunikációhoz szükséges);

S : aktuális feladat mérete;

$time$: aktuális feladat futási ideje másodpercben;

al : alsó korlát;

bu : felső korlát.

```

1.  $S_0 = \text{TRUE}$  // 1-4. sor: szelet állapot kezdeti beállítása
2.  $S_{N+1} = \text{TRUE}$ 
3. for  $j = 1$  to  $N + 1$ 
4.      $S_j = \text{FALSE}$ 
5.  $G_z = 0$  // 5. sor:  $G_z$  kezdeti értékének beállítása
6. while  $fj < N$  // 6. sor: amíg van befejezetlen szelet
7.     accept( $cl$ ) // 7. sor: klienssel való kapcsolat elfogadása
8.     recv( $cl, msg$ ) // 8. sor: kliens üzenetének fogadása
9.     if  $msg = 0$  // 9. sor: kliens új szeletet kér
10.          $aj = aj + 1$  // 10. sor: utoljára küldött szelet indexének növelése
11.         for  $i = M_{aj-1,0}$  to  $n$  // 11-12. sor: kezdő sorozat frissítése
12.              $b_i = n + M_{aj-1,1}$ 
13.         while  $S_{aj} = \text{TRUE}$  or  $aj > N$  // 13-22. sor: befejezetlen szelet?
14.              $aj = aj + 1$ 
15.             if  $aj > N$  // 15. sor: átléptük a maximális indexet
16.                  $aj = 1$  // 16. sor: index beállítása 1-re
17.             for  $i = M_{aj-1,0}$  to  $n$  // 17-18. kezdő sorozat frissítése

```

```

18.           $b_i = n + M_{aj-1,1}$ 
19.  if  $a_j < N$           // 19–30. sor: utolsó szeletet jellemző sorozatok
                                beállítása
20.           $al = M_{aj,0}$ 
21.           $b = n + M_{aj,1}$ 
22.  else  $al = 1$ 
23.           $bu = 1$ 
24.  send( $c, b, al, bu$ )          // 24. sor: szelet küldése a kliensnek
25.  else recv( $c, j_i, F_{init}, F_{last}, Z_{n,m}, time$ ) // 25. sor: eredmény fogadása
26.  if  $S_{j_i} = \text{FALSE}$           // 26. sor: kapott eredmény új
27.           $S_j = \text{TRUE}$  // 27. sor: befejezett szelet állapotának beállítása
28.           $fj = fj + 1$  // 28. sor: befejezett szeletek számának növelése
29.           $G_z = G_z + Z_{n,m}$           // 29. sor:  $G_z$  frissítése
30.  close( $cl$ )          // 30. sor: hálózati kapcsolat zárása
31. return  $G_z$           // 31. sor: eredmény

```

A kliens készítésekor fontos szempont volt az egyszerűség. Egy olyan programot szeretnénk volna elkészíteni, amelynek nincs szüksége semmilyen felhasználói interakcióra. Elegendő, ha a felhasználó csak elindítja, és attól a pillanattól kezdve a program önállóan képes futni a háttérben. Ez azért fontos, mert a programot annyi helyen akartuk kis részekre osztva futtatni, amennyi helyet fel tudtunk használni számítógépes laboratóriumunkban, mivel nem volt elég időnk és emberünk, akik részt vettek volna a program futtatásában.

Egy másik fontos ötlet az volt, hogy nem akartuk újraindítani a programot, amikor a számítás $G_z(n)$ -ről átvált $G_z(n+1)$ -re. Miután a kliens végzett feladattal, és a szerver nem tudott neki újat adni, a háttérben várakozott – mindaddig amíg nem kapott új feladatot – jelentős erőforrás-felhasználás nélkül.

A kliens program szálként működik. Ennek egyszerű oka van: a programot feltöltöttük a publikus honlapunkra, így bárki csatlakozhatott a számításokhoz. Ezzel az volt a célunk, hogy egyetlen felhasználót se veszítsünk el amiatt, hogy a programunk lefoglalja az erőforrásait.

Harmadik szempontként egy igazán gyors programot szeretnénk volna készíteni, mivel a futási idő óriási mértékben megnövekedhet az n értékétől függően. Ezen okból az ANSI C nyelvet használtuk programunk elkészítésekor. Kísérleteink során kiderült, hogy programunk ANSI C-ben írt változata százszor gyorsabb volt, mint a MATLAB-ban készült változat. A hálózati kommunikációhoz Berkeley-socketeket használtunk.

A kliens a következőképp működik:

- Miután létrehoztuk a csatlakozáshoz szükséges csatlakozót (socket), megpróbálunk kapcsolódni a szerverhez. Ha ez nem lehetséges, várunk egy ideig, és megkétszerezzük ezt a várakozási időt. Ezt az eljárást addig ismételjük, amíg nem tudunk csatlakozni a szerverhez, vagy el nem érünk egy megadott időkorlátot. Ezután már nem növeljük a várakozási időt. Könnyen belátható, hogy a várakozási idő exponenciálisan nő.
- A szerverre való csatlakozás után elkérünk egy részfeladatot, majd miután megkaptuk, lebontjuk a hálózati kapcsolatot.
- Kiszámítjuk a $G_z(n)$ részleges eredményét, majd visszaküldjük a szervernek az első lépésben bemutatott csatlakozási eljárás szerint.

A klienseken futó PARALLEL ERDŐS–GALLAI-algoritmus két részre osztható: CHECK és ENUMERATING részre. Az első csak ellenőrzi a sorozatokat, mást nem csinál. A második generálja a sorozatokat, a H értékeket és az ellenőrző pontokat.

A CHECK-ben a lineáris Erdős–Gallai-algoritmus módosított változatát használjuk.

6.4. Algoritmus. CHECK(b, H, c, C)

Bemenet. b : bemenő sorozat;

$H = H_1, \dots, H_n$: a b sorozat elemeinek összege;

c : ellenőrző pontok száma;

$C = C_1, \dots, C_{n-1}$: ellenőrző pontok.

Kimenet. L : logikai érték. Ha a megvizsgált sorozat gráfos, akkor $L = 1$, különben $L = 0$.

Munkaváltozó. p : aktuális ellenőrző pont.

```

1.  $i = 1$  // 1. sor:  $i$  kezdeti értékének beállítása
2. while  $i \leq c$  and  $H_{C_i} > C_i(C_i - 1)$  // 2–10. sor: sorozatok tesztelése
3.    $p = C_i$  // 3. sor:  $p$  kezdeti értékének beállítása
4.   while  $J_p < n$  and  $b_{J_{p+1}} > p$  // 4–7. sor:  $J_p$  frissítése
5.      $J_p = J_p + 1$ 
6.   while  $J_p > p$  and  $b_{J_p} \leq p$ 
7.      $J_p = J_{p-1}$ 
8.   if  $H_p > H_n - H_{J_p} + p(J_p - 1)$  // 8. sor: tesztelés
9.      $L = 0$  // 9–10. sor:  $b$  nem gráfos
10.   return  $L$ 
11.    $i = i + 1$ 
12.  $L = 1$  // 12–13. sor:  $b$  gráfos
13. return  $L$ 

```

6.5. Algoritmus. ENUMERATING($n, b, last_index, last_value$)

Bemenet. n : a sorozat hossza;

b : első sorozat;

$last_index$: az utolsó ellenőrizendő sorozat elérésekor elem indexe;

$last_value$: az utolsó ellenőrizendő sorozat elérésekor elem értéke.

Kimenet. G_z^p : Az első és utolsó ellenőrizendő sorozat közötti n -szabályos nulla mentes gráfok sorozatok száma.

```

1.  $H_1 = b_1$  // 1. sor:  $H_1$  beállítása
2. for  $i = 2$  to  $n$  // 2–3. sor:  $H$  elemeinek számítása
3.    $H_i = H_{i-1} + b_i$ 
4. if  $b_n \neq n - 1$  // 4. sor: ha a gráf nem teljes
5.   if  $H_n$  páratlan // 5–10. sor: sorozat frissítése
6.      $b_n = b_n - 3$ 
7.      $H_n = H_n - 3$ 
8.   else  $b_n = b_n - 2$ 
9.      $H_n = H_n - 2$ 
10. for  $i = 1$  to  $n$  // 10–11. sor: súlypontok kezdeti értékeinek beállítása
11.    $J_i = n - 1$ 
12. for  $i = 1$  to  $n - 2$  // 12–15. sor: ellenőrző pontok számítása
13.   if  $b_i \neq b_{i+1}$  and  $b_i \neq b_n$ 
14.      $c = c + 1$ 
15.      $C_c = i$ 
16.  $L = \text{CHECK}(b, H, c, C)$  // 16. sor: első sorozat tesztelése
17.  $G_z^p = G_z^p + L$ 
18. while  $b_{last\_index} > last\_value$  // 18. sor: szelet utolsó sorozatáig
19.    $k = n$  // 19. sor: munkaváltozó kezdeti értékének beállítása
20.   if  $b_k = 1$  // 20. sor: ha a sorozat utolsó eleme 1
21.      $j = n - 1$ 
22.     while  $b_j \leq 1$ 
23.        $j = j - 1$ 
24.     if  $b_j = 2$  // 24. sor: ha az utolsó elem 2
25.        $b_{j-1} = b_{j-1} - 1$  // 25. sor: sorozat frissítése
26.        $H_{j-1} = H_{j-1} - 1$  // 26. sor:  $H$  elemeinek frissítése
27.       if  $j > 2$  // 27–36. sor: ellenőrző pontok frissítése
28.         if ( $c \leq 2$  or ( $c > 2$  and  $C_{c-2} \neq j - 2$ )) and
           ( $c > 1$  and  $C_{c-1} \neq j - 2$ )
29.         if  $c > 1$  and  $C_{c-1} > j - 2$ 

```

```

30.           $C_{c+1} = C_c$ 
31.           $C_c = C_{c-1}$ 
32.           $C_{c-1} = j - 2$ 
33.           $c = c + 1$ 
34.      else  $C_{c+1} = C_c$ 
35.           $C_c = j - 2$ 
36.           $c = c + 1$ 
37.  for  $k = j$  to  $n$ 
38.       $b_k = b_{j-1}$           // 39. sor:  $b$  utolsó részének frissítése
39.       $H_k = H_{k-1} + b_k$           // 40. sor:  $H$  frissítése
40.  while  $c > 1$  and  $C_c > j - 1$  // 42–43. sor: ellenőrző pontok
                                     frissítése
41.       $c = c - 1$ 
42.  if  $H_n$  páratlan          // 42. sor: ha a paritás páratlan
43.       $b_n = b_{n-1} - 1$           // 43. sor:  $b$  frissítése
44.       $H_n = H_{n-1} + b_n$           // 44. sor:  $H$  frissítése
45.       $c = c + 1$           // 45–46. sor: ellenőrző pontok frissítése
46.       $C_c = n - 1$ 
47.  else  $b_j = b_j - 1$           // 47. sor:  $b$  frissítése
48.       $H_j = H_j - 1$           // 48. sor:  $H$  frissítése
49.  if  $j > 1$           // 49–50. sor: ellenőrző pontok frissítése
50.      if ( $c = 1$  and  $C_c \neq j - 1$ ) or ( $c > 1$  and  $C_{c-1} \neq j - 1$ )
51.          if  $c > 0$  and  $C_c > j - 1$ 
52.               $C_{c+1} = C_c$ 
53.               $C_c = j - 1$ 
54.               $c = c + 1$ 
55.  for  $k = j + 1$  to  $n$ 
56.       $b_k = b_j$           // 56. sor:  $b$  frissítése
57.       $H_k = H_{k-1} + b_k$           // 57. sor:  $H$  frissítése
58.  while  $c > 1$  and  $C_c > j - 1$  // 58–59. sor: ellenőrző pontok
                                     frissítése
59.       $c = c - 1$ 
60.  if  $H_n$  páratlan          // 60. sor: paritás ellenőrzése
61.       $b_n = b_n - 1$           // 61. sor:  $b$  frissítése
62.       $H_n = H_n - 1$           // 62. sor:  $H$  frissítése
63.       $c = c + 1$           // 63. sor: ellenőrző pontok frissítése
64.       $C_c = n - 1$  // 64. sor: új ellenőrző pont listához fűzése
65.  else if  $b_k = 2$ 

```

```

66.           $b_{k-1} = b_{k-1} - 1$                                 // 66. sor:  $b$  frissítése
67.           $H_{k-1} = H_{k-1} - 1$                                 // 67. sor:  $H$  elemeinek frissítése
68.          if ( $c = 1$  and  $C_c \neq n - 2$ ) or ( $c > 1$  and  $C_{c-1} \neq n - 2$  and
               $C_c \neq n - 2$ ) // 68–73. sor: ellenőrző pontok frissítése
69.          if  $c > 0$  and  $C_c > n - 2$ 
70.               $C_{c+1} = C_c$ 
71.               $C_c = n - 2$ 
72.          else  $c = c + 1$ 
73.               $C_c = n - 2$ 
74.          if  $b_{k-1}$  odd // 74. sor: paritás tesztelése
75.               $b_k = b_{k-1}$  // 75. sor:  $b$  frissítése
76.          if  $c > 0$  and  $C_c = n - 1$ 
77.               $c = c - 1$  // 77. sor: ellenőrzőpont frissítése
78.          else  $b_k = b_{k-1} - 1$  // 78. sor:  $b$  frissítése
79.           $H_k = H_{k-1} + b_k$  // 79. sor:  $H$  elemének kiszámítása
80.          else  $b_k = b_k - 2$  // 80. sor:  $b$  frissítése
81.           $H_k = H_k - 2$  // 81. sor:  $H$  elemének kiszámítása
82.          if  $c < 1$  or  $C_c \neq n - 1$  // 82–84. sor: ellenőrző pontok
                                                    frissítése
83.               $c = c + 1$ 
84.               $C_c = n - 1$ 
85.           $G_z^p = G_z^p + \text{CHECK}(b, H, c, C)$  // 85. sor:  $G_z^p$  frissítése

```

A *The On-Line Encyclopedia of Integer Sequences* [170] tartalmazza az n csúcsú egyszerű gráfok foksorozatainak számát. Az $n = 24, \dots, 29$ értékeket november 16-án töltöttük fel. A [95] cikk 2. táblázatában megtalálható az összes $G(n)$ érték, amelyet az OEIS adatbázis jelenleg tartalmaz.

Számításaink során több mint 200 számítógépet használtunk fel. Számítási teljesítményüket összegezve elmondható, hogy számítási kapacitásunk elméleti maximuma – beleértve a magánszemélyek teljesítményét is – elérte a 6 TFLOPS-ot.

A gráfok sorozatok kiszámításának futási idejét a 14. táblázat tartalmazza. Észrevehető, hogy a futási idő növekedése nem ugyanolyan arányú különböző n értékek esetén. Ez a felhasznált processzorok típusával magyarázható. A korai számítások idején (például $n = 25$ esetén) néhány erős géppel rendelkezünk, azonban a növekvő n -ek során fellépő nagyobb komplexitás miatt inkább sok kevésbé erős gépet használtunk. A számítások teljes ideje kevesebb, ha néhány erős gépet használunk, de a valós futási idő nőni fog. Több mint 200 gép felhasználásával a G_{29} valós futási ideje több, mint két hét volt.

n	Futási idő (nap)	Szeletek száma
25	26	435
26	70	435
27	316	435
28	1130	2 001
29	6733	15 119

14. táblázat. A mért futási idők összege, valamint a feldolgozott szeletek száma.

7. A program további gyorsításának lehetőségei

Leszámláló programunk futási idejét két fő módszerrel csökkenthetjük: az egyik módszer a tesztelendő sorozatok számát, a másik pedig az egyes sorozatok ellenőrzésének idejét csökkenti.

Ebben a részben az előbbi módszer három érdekes elemét, a rossz és jó sorozatok egy részének átugrását elemezzük.

7.1. A nemgrafikus sorozatok egy részének átugrása

Ezt a módszert az $n = 30$ -as projektben vezettük be.

Az $n = 23, \dots, 29$ esetben a foksorozatokat a páros sorozatok között kerestük. Ekkor a 7 516 816 644 943 560 páros sorozat között a 2 022 337 118 015 338 grafikus sorozat 26,90 százalékot képviselt. Ismert (lásd a [95] cikkben a 13. lemmát), hogy $E(n) = \Theta(4^n/\sqrt{n})$, míg Burns tétele (lásd az előbbi cikkben a 22. tételt) $G(n)$ szerint van olyan pozitív C , amelyre $G(n) = O(4^n/\sqrt{n}/(\log n)^C)$, ahonnan következik, hogy a grafikus sorozatok a páros sorozatok között aszimptotikusan nullmértékű jelentős részét részalmazt képviselnek.

Az $n = 30$ és $n = 31$ esetben a rossz sorozatok jelentős részét átugrottuk. Ennek köszönhetően az $n = 30$ esetben a tesztelt sorozatok 85,40%; míg az $n = 31$ esetben 86,67 százaléka volt grafikus. Mivel a futási idők még nagyszámú processzor felhasználása mellett is jelentősek, kulcsfontosságú mind a nemgrafikus, mind pedig grafikus sorozatok minél nagyobb hányadának átugrása.

A nemgrafikus sorozatok jelentős részének tényleges ellenőrzését a következő lemma alapján ugrottuk át az $n = 30$ és $n = 31$ esetben.

Ha $b = (b_1, \dots, b_n)$ $(0, n-1, n)$ -szabályos sorozat, akkor a $c = (c_1, c_2, \dots, c_n)$ sorozatot b -nél lexikografikusan (i, j) -kisebbnek nevezzük, ha vannak olyan $1 \leq i < j \leq n$ indexek, melyekre

$$\text{ha } k = 1, \dots, i, \text{ akkor } c_k = b_k,$$

és

$$\sum_{k=i+1}^n c_k \leq \sum_{k=i+1}^n b_k$$

és

$$\sum_{k=i+1}^j c_k < \sum_{k=i+1}^j b_k.$$

7.1. LEMMA. Ha $b = b_1, \dots, b_n$ nemgrafikus sorozat, és vannak olyan i és j indexek és egy olyan $c = c_1, c_2, \dots, c_n$ sorozat, amely lexikografikusan (i, j) -kisebbséget képez, mint b , akkor c nemgrafikus.

Bizonyítás. Lásd [88]. □

A lemma alkalmazásával mintegy 30 százalékkal sikerült az EGE1 program futási idejét csökkenteni.

7.2. A kis grafikus sorozatok átugrása

Az előző definícióhoz és lemmához hasonló eszközökkel kezeljük a grafikus sorozatok egy részének átugrását.

Ha $b = (b_1, \dots, b_n)$ szabályos sorozat, akkor a $c = (c_1, \dots, c_n)$ sorozatot b -nél lexikografikusan (i, j) -nagyobbknak nevezzük, ha vannak olyan $1 \leq i < j < n$ indexek, melyekre

$$\text{ha } k = 1, \dots, i, \text{ akkor } c_k = b_k$$

és

$$\sum_{k=i+1}^j c_k > \sum_{k=i+1}^j b_k,$$

továbbá

$$\sum_{k=i+1}^n c_k \geq \sum_{k=i+1}^n b_k.$$

7.2. LEMMA. Ha $b = b_1, b_2, \dots, b_n$ nemgrafikus sorozat, és vannak olyan i és j indexek és egy olyan $c = c_1, c_2, \dots, c_n$ sorozat, amely lexikografikusan (i, j) -nagyobb, mint b , akkor c nemgrafikus.

Bizonyítás. Lásd [88]. □

7.3. A gráfok sorozatok komplementereinek átugrása

Ha n pozitív egész szám, és a $b = b_1 \leq \dots \leq b_n \leq n - 1$ nemcsökkenő n -szabályos sorozat, akkor a $b' = n - 1 - b_n \leq \dots \leq n - 1 - b_1$ sorozatot a b sorozat *nemcsökkenő komplementerének* nevezzük. Ehhez hasonlóan definiáljuk a b nemnövekvő n -szabályos sorozatok *nemnövekvő komplementerét* is.

A gráfok sorozatok átugrásának alapja lehet az az egyszerű észrevétel is, hogy egy gráfok sorozat komplementere is gráfok sorozat.

7.3. LEMMA. Ha egy nemnövekvő szabályos $b = b_1, b_2, \dots, b_n$ sorozat gráfos, akkor a $b' = n - 1 - b_n, n - 1 - b_{n-1}, \dots, n - 1 - b_1$ sorozat is gráfos.

Bizonyítás. Ha b gráfos sorozat, akkor létezik olyan G egyszerű gráf, amelynek nemnövekvően rendezett foksorozata $b = b_1, b_2, \dots, b_n$. Ennek a gráfnak van komplementere, és abban a foksámok nemnövekvően rendezett sorozata $b' = n - 1 - b_1, n - 1 - b_2, \dots, n - 1 - b_n$. Tehát a b' sorozat szintén gráfos. \square

Ha egy nemnövekvő (vagy nemcsökkenő) n -szabályos sorozat nemnövekvően (nemcsökkenően) rendezett komplementer sorozata ugyancsak b , akkor akkor a b sorozatot *félpalindrom sorozatnak* nevezzük. A névválasztás oka, hogy a palindrom kifejezés már foglalt [2, 101, 102] az olyan sorozatokra, amelyek megegyeznek a saját megfordításukkal.

A $b = b_1, \dots, b_n$ sorozat *szimmetrikus elempárjainak* a $b_1 - b_n, b_2 - b_{n-1}, \dots, b_{\lfloor (n+1)/2 \rfloor} - b_{\lceil (n+1)/2 \rceil}$ elempárokat nevezzük. Megjegyezzük, hogy ez a definíció azt is jelenti, hogy páratlan n esetén az utolsó szimmetrikus párban kétszer szerepel a sorozat középső eleme.

7.4. LEMMA. A $b = b_1, \dots, b_n$ n -szabályos sorozat akkor és csak akkor félpalindrom sorozat, ha elempárjainak összege $n - 1$.

Bizonyítás. Az állítás a félpalindrom sorozatok definíciójából következik. \square

Ha egy n -szabályos sorozat nem félpalindrom sorozat, akkor szimmetrikus elempárjainak elemei közül a lexikografikusan kisebbet (az adott rendezés szerint előbb sorra kerülőt) *erősnek*, a másik elemet pedig *gyengének* nevezzük.

7.3.1. Komplementerek átugrása szabályos sorozatok tesztelése során

Ha a potenciális gráfok sorozatok tesztelése során egy b sorozatról kiderül, hogy gráfos, akkor a b' sorozat is gráfos. Ha egy gráfos b sorozat b' komplementere nem azonos b -vel, akkor b' -t nem kell tesztelni. Ezért például majd az $n = 33$ csúcsú gráfok foksorozatainak leszámllálása során elegendő a legalább 16-tal kezdődő potenciális sorozatokat vizsgálni, mert ha egy b nemcsökkenő gráfos sorozatban $b_1 \leq 16$, akkor a nemcsökkenő b' sorozatban $b'_1 \geq 16$, azaz ennek a sorozatnak a komplementerét már korábban megvizsgáltuk.

Egy legalább 16-tal kezdődő b gráfos sorozat egyúttal félpalindrom sorozat, akkor *egy* gráfos sorozatot képvisel, egyébként azonban kettőt. Ezért a megtalált gráfos sorozatokat ellenőrizni kell, hogy félpalindrom sorozatok-e. A FÉLPALINDROM algoritmus elvégzi ezt az ellenőrzést.

7.1. Algoritmus. FÉLPALINDROM(n, b)

Bemenet. n : csúcsok száma ($n \geq 1$);

$b = b_1, \dots, b_n$: a megvizsgálandó n -szabályos sorozat.

Kimenet. 0 vagy 1 (1, ha a vizsgált sorozat félpalindrom, egyébként 0).

Munkaváltozó. i: ciklusváltó.

1. **if** n páratlan **and** $(n + 1)/2 \neq (n + 1)/2$ // 1-6. sor: tesztelés
2. **return** 0 // 1-5. sor: b nem félpalindrom sorozat
3. **for** $i = 1$ **to** $\lceil n/2 \rceil$
4. **if** $b_i + b_{n-i+1} \neq n - 1$
5. **return** 0
6. **return** 1 // 6. sor: b félpalindrom sorozat

FÉLPALINDROM futási ideje legrosszabb esetben (például ha b félpalindrom sorozat) $\Theta(n)$, azonban az átlagos futási ideje ennél lényegesen kisebb, mivel a gráfos sorozatok között kevés félpalindrom sorozat van.

Egy nemnövekvő 1-szabályos sorozat van: 0 – ez félpalindrom sorozat. Három nemnövekvő 2-szabályos sorozat van: 1, 1; 0, 1 és 0, 1. Közülük az 1, 1 erős, a 0, 0 gyenge, a 0, 1 pedig félpalindrom sorozat. Tíz 3-szabályos sorozat van: 2, 2, 2; 2, 2, 1; 2, 2, 0; 2, 1, 1; 2, 1, 0; 2, 0, 0; 1, 1, 1; 1, 1, 0; 1, 0, 0 és 0, 0, 0. Ezek közül az 1, 1, 1 és a 0, 1, 2 félpalindrom, a többiek közül 4 erős és 4 gyenge. Harmincöt 4-szabályos sorozat van:

(3, 3, 3, 3); (3, 3, 3, 2); (3, 3, 3, 1); (3, 3, 3, 0); (3, 3, 2, 2); (3, 3, 2, 1); (3, 3, 2, 0);
 (3, 3, 1, 1); (3, 3, 1, 0); (3, 3, 0, 0); (3, 2, 2, 2); (3, 2, 2, 1); (3, 2, 2, 0); (3, 2, 1, 1);
 (3, 2, 1, 0); (3, 2, 0, 0); (3, 1, 1, 1); (3, 1, 1, 0); (3, 1, 0, 0); (3, 0, 0, 0); (2, 2, 2, 2);
 (2, 2, 2, 1); (2, 2, 2, 0); (2, 2, 1, 1); (2, 2, 1, 0); (2, 2, 0, 0); (2, 1, 1, 1); (2, 1, 1, 0);
 (2, 1, 0, 0); (2, 0, 0, 0); (1, 1, 1, 1); (1, 1, 1, 0); (1, 1, 0, 0); (1, 0, 0, 0); (0, 0, 0, 0).

A százhuszonhat 5-szabályos sorozat közül (4, 4, 2, 0, 0); (4, 3, 2, 1, 0); (4, 2, 2, 2, 0); (3, 3, 2, 1, 1); (3, 2, 2, 2, 1) és (2, 2, 2, 2, 2) félpalindrom, a többieknek pedig fele erős, fele pedig gyenge. A 6-szabályos sorozatok között 10, a 7-szabályos sorozatok között pedig 20 félpalindrom sorozat van.

A következő állítás megadja az n -szabályos sorozatok között lévő félpalindrom, erős és gyenge sorozatok számát.

7.5. LEMMA. *Ha $n \geq 1$, akkor a nemcsökkenő n -szabályos sorozatok között*

$$P(n) = \binom{n-1}{\lfloor n/2 \rfloor}$$

n -félpalindrom sorozat, valamint

$$S(n) = W(n) = (E(n) - P(n))/2$$

erős és gyenge sorozat van.

Bizonyítás. Ha n páros, akkor a sorozat első $\lfloor n/2 \rfloor$ elemét ismétléssel választhatjuk ki a $[0, \lfloor (n-1)/2 \rfloor]$ intervallum $\lfloor n/2 \rfloor$ eleme közül. Ha pedig n páratlan, akkor a

n	$R(n)$	$P(n)$	$S(n) = W(n)$	$W(n)/R(n)$
1	1	1	0	0.0000000000
2	3	1	1	0.3333333333
3	10	2	4	0.4000000000
4	35	3	16	0.4571428571
5	126	6	60	0.4761904762
6	462	10	226	0.4891774891
7	1 716	20	848	0.4941724942
8	6 435	35	3 200	0.4972804973
9	24 310	70	12 120	0.4985602633
10	92 378	126	46 126	0.4993180194
11	352 716	252	176 232	0.4996427721
12	1 352 078	462	675 808	0.4998291519
13	5 200 300	924	2 599 688	0.4999111590
14	20 058 300	1 716	10 028 292	0.4999572247
15	77 558 760	3 432	38 777 664	0.4999977875
16	300 540 195	6 435	150 266 880	0.4999989295
17	1 166 803 110	12 870	583 395 120	0.4999944849
18	4 537 567 650	24 310	2 268 771 670	0.4999973213
19	17 672 631 900	48 620	8 836 291 640	0.4999986244
20	68 923 264 410	92 378	34 461 586 016	0.4999993298
21	269 128 937 220	184 756	134 564 376 232	0.4999996568
22	1 052 049 481 860	352 716	526 024 564 572	0.4999998324
23	4 116 715 363 800	705 432	2 058 357 329 184	0.4999999143
24	16 123 801 841 550	1 352 078	8 061 900 244 736	0.4999999581
25	63 205 303 218 876	2 704 156	31 602 650 257 360	0.4999999786
26	247 959 266 474 052	5 200 300	123 979 630 636 876	0.4999999891
27	973 469 712 824 056	10 400 600	486 734 851 211 728	0.4999999947

15. táblázat. Csúcsok n , nemnövekvő szabályos sorozatok $R(n)$, félpalindrom sorozatok $P(n)$, erős sorozatok $S(n)$ és gyenge sorozatok $W(n)$ száma, valamint az átugrott és a szabályos sorozatok számának $W(n)/R(n)$ hányadosa $1, \dots, 27$ csúcs esetén.

sorozat középső eleme $\lfloor n/2 \rfloor$, az $\lfloor n/2 \rfloor$ kezdő elemet pedig ismétléssel választhatjuk ki $\lfloor n/2 \rfloor$ elem közül.

Az erős és gyenge sorozatok kölcsönösen egyértelműen megfeleltethetőek egymásnak, ezért az első egyenlőség igaz. A szabályos sorozatok halmazában háromféle sorozat van: félpalindrom, erős és gyenge. Ezért a második egyenlőség is igaz. □

A 15. táblázatban összefoglaljuk a nemnövekvő szabályos sorozatok bizonyos adatait $n = 1, 2, \dots, 27$ csúcs esetén.

A táblázat alapján úgy tűnik, hogy a hányadosok sorozata monoton növekedve 0.5-höz tart, azaz aszimptotikusan a sorozatok felét nem kell tesztelnünk. A 7.1. tétel bizonyítja, hogy ez igaz.

7.1. TÉTEL. *Ha n tart a végtelenbe, akkor*

$$\lim_{n \rightarrow \infty} \frac{S(n) + P(n)}{R(n)} = \frac{1}{2}. \quad (12)$$

Bizonyítás. Két eset van. Ha n páros, akkor a 7.5. lemma szerint

$$\begin{aligned} \frac{S(n) + P(n)}{R(n)} &= \frac{R(n)/2 - P(n)/2 + P(n)}{R(n)} = \frac{R(n) + P(n)}{2R(n)} = \frac{1}{2} + \frac{P(n)}{2R(n)} = \\ &= \frac{1}{2} + \frac{(n-1)!n!(n-1)!}{2(n/2)!(n/2)!(2n-1)!}. \end{aligned}$$

A Stirling-formula

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + O\left(\frac{1}{n}\right)\right)$$

alakját felhasználva

$$\frac{n^{3n-2}e^{3n-1}}{n^{3n-1}e^{3n-2}} \frac{\sqrt{2\pi(n-1)^2 2\pi n}}{\sqrt{2(n\pi)^3}} \frac{(1 + O(1/(n-1)))^2 O(1 + 1/n)}{(1 + O(2/n))^2 (1 + O(1/(2n-1)))},$$

ahonnan

$$\frac{S(n) + P(n)}{R(n)} = \frac{1}{2} + \frac{e}{n} \Theta(1),$$

amiből következik (12).

Ha n páratlan, akkor a gondolatmenet és az eredmény hasonló. \square

Ha a potenciális gráfok sorozatokat lexicografikusan nemcsökkenő sorrendben teszteljük, a félpalindrom sorozatok továbbra is félpalindrom sorozatok lesznek, míg az erős és gyenge sorozatok szerepet cserélnek. Eközben a különböző típusú sorozatok száma nem változik.

7.3.2. Komplementerek átugrása páros sorozatok tesztelése során

Most vizsgáljuk meg azt az esetet, amikor a gráfok sorozatokat a páros sorozatok között keressük.

Egy nemnövekvő páros 1-sorozat van: 0 – ez félpalindrom sorozat. Két nemnövekvő páros 2-szabályos sorozat van: 1, 1 és 0, 0. Közülük az 1, 1 erős, a 0, 0 pedig gyenge. Hat 3-páros sorozat van: 2, 2, 2; 2, 2, 0; 2, 1, 1; 2, 0, 0; 1, 1, 0 és 0, 0, 0. Ezek között nincs félpalindrom, viszont 3 erős és 3 gyenge sorozat van. Tizenkilenc 4-páros sorozat van:

(0, 0, 0, 0); (0, 0, 0, 2); (0, 0, 1, 1); (0, 0, 1, 3); (0, 0, 2, 2); (0, 0, 3, 3); (0, 1, 1, 2); (0, 1, 2, 3); (0, 2, 2, 2); (0, 2, 3, 3); (1, 1, 1, 1); (1, 1, 1, 3); (1, 1, 2, 2); (1, 1, 3, 3); (1, 2, 2, 3); (1, 3, 3, 3); (2, 2, 2, 2); (2, 2, 3, 3) és (3, 3, 3, 3).

n	$E(n)$	$P_e(n)$	$S_e(n) = W_e(n)$	$W_e(n)/E(n)$
1	1	1	0	0.000000000000000
2	2	0	1	0.500000000000000
3	6	0	3	0.500000000000000
4	19	3	8	0.421052631578947
5	66	6	30	0.454545454545455
6	236	10	113	0.478813559322034
7	868	0	434	0.500000000000000
8	3235	35	1600	0.494590417310665
9	12190	70	6060	0.497128794093519
10	46252	126	23063	0.498637896739600
11	176484	0	88242	0.500000000000000
12	676270	462	337904	0.499658420453369
13	2600612	924	1299844	0.499822349508500
14	10030008	1716	5014146	0.499914456698340
15	38781096	0	19390548	0.500000000000000
16	150273315	6435	75133440	0.499978589012959
17	583407990	12870	291697560	0.499988969983082
18	2268795980	24310	1134385835	0.499994642532820
19	8836340260	0	4418170130	0.500000000000000
20	34461678394	92378	17230793008	0.499998659699639
21	134965161188	184756	67482488216	0.499999315541884
22	526024917288	352716	263012282286	0.499999664734513
23	2058358034616	0	1029179017308	0.500000000000000
24	8061901596814	1352078	4030950122368	0.499999916143978
25	31602652961516	2704156	15801325128680	0.499999957216313
26	123979635837176	5200300	61989815318438	0.499999979027604
27	486734861612328	0	243367430806164	0.500000000000000

16. táblázat. Csúcsok n , nemnövekvő páros sorozatok $E(n)$, páros félpalindrom sorozatok $P_e(n)$, páros erős sorozatok $S_e(n)$ és páros gyenge sorozatok $W_e(n)$ száma, valamint az átugrott sorozatok és a páros sorozatok számának $W_e(n)/E(n)$ hányadosa $1, \dots, 27$ csúcs esetén.

Ezek közül a $(3, 3, 0, 0)$; $(3, 2, 1, 0)$ és $(2, 2, 1, 1)$ félpalindrom, a többiek között pedig 16 erős és 16 gyenge sorozat van. A hatvanhat 5-szabályos sorozat közül a $(4, 4, 2, 0, 0)$; $(4, 3, 2, 1, 0)$; $(4, 2, 2, 2, 0)$; $(3, 3, 2, 1, 1)$; $(3, 2, 2, 2, 1)$ és a $(2, 2, 2, 2, 2)$ félpalindrom, a többieknek pedig fele erős, fele pedig gyenge.

7.6. LEMMA. *Ha $n \geq 1$, akkor a nemnövekvő n -páros sorozatok között páros n és $4k + 1$ alakú páratlan n esetén*

$$P(n) = \binom{n-1}{\lfloor n/2 \rfloor},$$

$4k + 3$ alakú páratlan n esetén pedig

$$P(n) = 0$$

n -félpalindrom sorozat és

$$W_e(n) = S_e(n) = \frac{E(n) - P_z(n)}{2}$$

erős és gyenge sorozat van.

Bizonyítás. Lásd [88]. □

7.3.3. Komplementerek átugrása nullamentes páros sorozatok tesztelése során

Most vizsgáljuk meg azt az esetet, amikor a gráfok sorozatokat a nullamentes páros sorozatok között keressük. Ezek kezelése bonyolultabb, mint a szabályos és páros sorozatoké volt.

Egy nullamentes gráfok sorozat izolált csúcsot tartalmazó gráfhoz tartozik. Egy nullamentes sorozatban vagy van $n - 1$, vagy nincs. Az $n - 1$ -et tartalmazó sorozatok komplementerében van nulla, ezért az ilyen sorozatok komplementerét nem generáljuk, és ha gráfok, csak egyszeres súllyal vesszük őket figyelembe. Az ilyen sorozatokat *félerősnek* nevezzük és számukat $M'_z(n)$ -nel jelöljük az $(n - 1)$ -et nem tartalmazó esetben, és $M''_z(n)$ -nel a másikban.

Először foglalkozunk az $(n - 1)$ -et nem tartalmazó sorozatokkal. A nullamentes 1-páros és a 2-páros sorozatok között nincs megfelelő sorozat. A tizenkilenc 4-páros sorozat közül most csak 3 felel meg: 2, 2, 2, 2; 2, 2, 1, 1 és 1, 1, 1, 1. Közülük a 2, 2, 1, 1 félpalindrom, a 2, 2, 2, 2 erős és az 1, 1, 1, 1 a gyenge párja, azaz $P'_z(4) = 1$ és $S'_z(4) = W'_z(4) = 1$. A hatvanhat 5-páros sorozat közül most csak a következő 9 felel meg:

(3, 3, 3, 3, 2); (3, 3, 3, 2, 1); (3, 3, 2, 2, 2); (3, 3, 2, 1, 1); (3, 2, 2, 2, 1); (3, 2, 1, 1, 1);
(2, 2, 2, 2, 2); (2, 2, 2, 1, 1); (2, 1, 1, 1, 1).

Ezek közül a 3, 3, 2, 1, 1; 3, 2, 2, 2, 1 és 2, 2, 2, 2, 2 a félpalindrom, és van további 3 erős és 3 gyenge.

A 236 páros 6-sorozat közül a következő ötven 0-mentes és ugyanakkor 5-mentes is:

(4, 4, 4, 4, 4, 4); (4, 4, 4, 4, 4, 2); (4, 4, 4, 4, 3, 3); (4, 4, 4, 4, 3, 1); (4, 4, 4, 4, 2, 2);
(4, 4, 4, 4, 1, 1); (4, 4, 4, 3, 3, 2); (4, 4, 4, 3, 2, 1); (4, 4, 4, 2, 2, 2); (4, 4, 4, 2, 1, 1);
(4, 4, 3, 3, 3, 3); (4, 4, 3, 3, 3, 1); (4, 4, 3, 3, 2, 2); (4, 4, 3, 3, 1, 1); (4, 4, 3, 3, 2, 1);
(4, 4, 3, 2, 2, 1); (4, 4, 3, 1, 1, 1); (4, 4, 2, 2, 2, 2); (4, 4, 2, 2, 1, 1); (4, 4, 1, 1, 1, 1);
(4, 3, 3, 3, 3, 2); (4, 3, 3, 3, 3, 1); (4, 3, 3, 3, 2, 2); (4, 3, 3, 3, 2, 1); (4, 3, 3, 2, 2, 2);
(4, 3, 3, 2, 1, 1); (4, 3, 2, 2, 2, 1); (4, 3, 2, 1, 1, 1); (4, 2, 2, 2, 2, 2); (4, 2, 2, 2, 1, 1);
(4, 2, 1, 1, 1, 1); (3, 3, 3, 3, 3, 3); (3, 3, 3, 3, 3, 1); (3, 3, 3, 3, 2, 2); (3, 3, 3, 3, 1, 1);
(3, 3, 3, 3, 2, 2); (3, 3, 3, 3, 2, 2); (3, 3, 3, 3, 1, 1); (3, 3, 3, 2, 2, 1); (3, 3, 3, 1, 1, 1);
(3, 3, 2, 2, 2, 2); (3, 3, 2, 2, 1, 1); (3, 3, 1, 1, 1, 1); (3, 2, 2, 2, 2, 1); (3, 2, 2, 1, 1, 1);
(3, 1, 1, 1, 1, 1); (2, 2, 2, 2, 2, 2); (2, 2, 2, 2, 1, 1); (2, 2, 1, 1, 1, 1); (1, 1, 1, 1, 1, 1).

Ezek között nincs félpalindrom, és 25 erős, valamint 25 gyenge van.

A 17. táblázatban összefoglaljuk a nemnövekvő nullamentes és $(n - 1)$ -mentes páros sorozatok bizonyos adatait.

n	$E_z(n)$	$E'_z(n)$	$P'_z(n)$	$S'_z(n) = W'_z(n)$	$\frac{W'_z(n)}{E'_z(n)}$
1	0	0	0	0	—
2	1	0	0	0	0.00000
3	2	0	0	0	0.00000
4	9	3	1	1	0.11111
5	28	9	3	3	0.10714
6	110	44	0	22	0.20000
7	396	160	0	80	0.20202
8	1 519	651	15	318	0.20934
9	5 720	2 485	35	1 225	0.21416
10	21 942	9 752	0	4 876	0.22222
11	83 980	37 728	0	18 864	0.22462
12	323 554	147 070	210	73 430	0.22695
13	1 248 072	571 802	462	285 670	0.22889
14	4 829 708	2 229 096	0	1 114 548	0.23077
15	18 721 080	8 691 072	0	4 345 536	0.23212
16	72 714 555	33 933 459	3 003	16 965 228	0.23331
17	282 861 360	132 588 045	6 435	66 290 805	0.23436
18	1 101 992 870	518 584 880	0	259 292 440	0.23529
19	4 298 748 300	2 029 952 320	0	1 014 976 160	0.23611
20	16 789 046 494	7 952 706 234	43 758	3 976 331 238	0.23684
21	65 641 204 200	31 179 525 806	92 378	15 589 716 714	0.23750
22	256 895 980 068	122 331 419 080	0	61 165 709 540	0.23810
23	1 006 308 200 040	480 283 282 752	0	240 141 641 376	0.23864
24	3 945 186 233 014	1 886 828 198 398	646 646	943 413 775 876	0.23913
25	15 478 849 767 888	7 416 948 171 074	1 352 078	3 708 473 409 498	0.23958
26	60 774 332 618 300	29 171 679 656 784	0	14 585 839 828 392	0.24000
27	238 775 589 937 976	114 795 954 100 800	0	57 397 977 050 400	0.24038
28	938 702 947 395 204	451 968 085 782 876	9 657 700	225 984 038 062 588	0.24074

17. táblázat. Csúcsok n , nemnövekvő nullamentes páros sorozatok $E_z(n)$, nullamentes, páros, $(n - 1)$ -et nem tartalmazó sorozatok $E'_z(n)$, félpalindrom sorozatok $P'_z(n)$, erős sorozatok $S'_z(n)$ és gyenge sorozatok $W'_z(n)$ száma, valamint a tesztelés nélkül átugrott sorozatok és a nullamentes páros sorozatok számának $W'_z(n)/E'_z(n)$ hányadosa $1, \dots, 28$ csúcs esetén.

A táblázatban lévő adatokat a következő képletekkel számítottuk.

A nullamentes páros sorozatok $E_z(n)$ számát megadja az (5.2) képlet. A következő 7.7. lemma alapján számítottuk $E'_z(n)$ értékét.

7.7. LEMMA. Ha $n \geq 1$, akkor

$$E'_z(n) = \sum_{i=0}^{\lfloor n/2 \rfloor} \binom{\lceil (n-2)/2 \rceil - 1 + 2i}{2i} \cdot \binom{\lfloor (n-2)/2 \rfloor - 1 + n - 2i}{n-2i}.$$

Bizonyítás. Mivel páros sorozatokat vizsgálunk, a bennük lévő páratlan elemek száma $0, 2, \dots, \lfloor n/2 \rfloor$ lehet. A sorozatok lehetséges elemei most $1, 2, \dots, n-2$, ezek között $\lceil (n-2)/2 \rceil$ páratlan és $\lfloor (n-2)/2 \rfloor$ páros elem van, amelyek közül $0, 2, \dots, \lfloor n/2 \rfloor$ páratlan elemet kell kiválasztanunk, a maradék helyeket pedig páros elemekkel kell feltölteniünk. \square

A félpalindrom, erős, gyenge és félerős sorozatok számát ebben az esetben a 7.8. lemma adja meg.

7.8. LEMMA. a) Ha $k \geq 0$ és $n = 4k + 2$, akkor

$$P'_z(n) = 0.$$

b) Ha $k \geq 0$ és $n = 4k$, akkor

$$P'_z(n) = \binom{(n-2)/2 - 1 + n/2}{n/2}.$$

c) Ha $k = 4k + 3$, akkor

$$P'_z(n) = 0.$$

d) Ha $n = 4k + 1$, akkor

$$P'_z(n) = \binom{(n-1)/2 - 1 + (n-1)/2}{(n-1)/2} = \binom{n-2}{(n-1)/2}.$$

e) Ha $n \geq 1$, akkor

$$S'_z(n) = W'_z(n) = \frac{E'_z(n) - P'_z(n)}{2}.$$

f) Ha $n \geq 1$, akkor

$$S''_z(n) = W''_z(n) = \frac{E'_z(n)}{2}.$$

Bizonyítás. a) Mivel $n = 4k + 2$, egy félpalindrom sorozatban $n/2 = 2k + 1$ szimmetrikus elempár lenne, melyek elemei összeadva $(n-1 = 4k+1)$ -et adnak, így a sorozat elemeinek összege $(2k+1)(4k+1)$ lenne, ami páratlan szám.

b) Ha $n = 4k$, akkor $n/2$ szimmetrikus pár van, és ha mindegyikben az előírt $4k-1$ az elemek összege, a sorozat páros. Az első $n/2$ elem az $[1, \lfloor (n-1)/2 \rfloor] = [1, (n-2)/2]$ intervallumba tartozik és innen kell ismétléssel $n/2$ elemet kiválasztani.

c) Ha $n = 4k + 3$, akkor a kételemű szimmetrikus párok elemeinek összege $4k + 2$, viszont a középső elemnek a $2k + 1$ értéket kellene felvennie, ami páratlan szám, és így a sorozat elemeinek összege is páratlan lenne.

d) A középső elemnek a páros $2k = (n - 1)/2$ értéket kell felvennie, ezért a szóba jövő sorozatok elemeinek összege páros. Az első $(n - 1)/2$ elemet az $[1, (n - 1)/2]$ intervallum elemei közül kell ismétléssel kiválasztanunk és ezek az elemek egyértelműen meghatározzák az utolsó $(n - 1)/2$ elemet.

e) A palindrom sorozatokon kívül csak erős és gyenge sorozatok vannak, amelyek kölcsönösen egyértelműen megfeleltethetőek egymásnak.

f) Ebben az esetben minden erős sorozatnak van gyenge párja, palindrom és félerős sorozat nincs. \square

Most nézzük az $(n - 1)$ -et tartalmazó sorozatokat. $E_z''(n) = E_z(n) - E_z'(n)$ ilyen sorozat van. Mivel $n - 1$ eleme a vizsgált sorozatoknak, viszont a nulla nem, ezért itt félpalindrom sorozat nincs, továbbá erős, valamint gyenge sorozat sincs – minden sorozat félerős, ezért mindegyiket tesztelni kell.

Korábbi szimulációs eredményeink [90] szerint aszimptotikusan a páros sorozatok fele nullamentes. A 17. táblázat adatai pedig alátámasztják azt a sejtést, hogy aszimptotikusan a nullamentes páros sorozatok negyede gyenge, azaz átugorható, ezért ettől az algoritmustól a futási idő közel 25 százalékos csökkenését várjuk.

8. Összefoglalás

A naplófájlok és programok forráskódja megtalálható a

<http://people.inf.elte.hu/lulsaai/Holz hacker> és

<http://people.inf.elte.hu/tomintt/DegreeSeq>

címeken [120, 128].

Köszönetnyilvánítás.

A szerzők köszönik Lucz Lorándnak, Matuszka Tamásnak és Szabados Kristófnak a számítógépes szimulációhoz nyújtott segítséget.

Hivatkozások

- [1] ACOSTA, P., BASSA, A., CHAIKIN, A., RIEHL, A., TINGSTAD, A. ZHAO, L., KLEITMAN, D. J.: *On a conjecture of Brualdi and Shen on block transitive tournaments*, Journal of Graph Theory, **44(3)**, (2003) 215–230.

- [2] ANISIU, M.-C., ANISIU, V., KÁSA, Z.: *Total palindrome complexity of finite words*, Discrete Mathematics, **310**, (2010) 109–114.
- [3] ARIKATI, S. R., MAHESHWARI, A.: *Realizing degree sequences in parallel*, SIAM Journal on Discrete Mathematics, **9(2)**, (1996) 317–338.
- [4] AO, S., HANSON, D.: *Score vectors and tournaments with cyclic chromatic number 1 or 2*, Ars Combinatoria, **49**, (1998) 185–191.
- [5] ASCHER, M.: *Mu torere: An analysis of a Maori game*, Mathematical Magazine, **60(2)**, (1987,) 90–100.
- [6] AVERY, P.: *Score sequences of oriented graphs*, Journal of Graph Theory, **15**, (1971) 251–257.
- [7] AVERY, P.: *Condition for a tournament score sequence to be simple*, Journal of Graph Theory, **4(2)**, (1980) 157–164.
- [8] BAGGA, K. S., BEINEKE, L. W., HARARY, F.: *Two problems on coloring tournaments*, Vishwa Internat Journal of Graph Theory, **1(1)**, (1992) 83–94.
- [9] BANG, C. M., SHARP, H.: *An elementary proof of Moon's theorem on generalized tournaments*, Journal of Combinatorial Theory, Series B, **5**, (1977) 299–301.
- [10] BANG, C. M., SHARP, H., JR.: *Score vectors of tournaments*, Journal of Combinatorial Theory, Series B, **26(1)**, (1979) 81–84.
- [11] BANG-JENSEN, J., GUTIN, G.: *Generalizations of tournaments – a survey*, Journal of Graph Theory, **28**, (1998) 178–202.
- [12] BANG-JENSEN, J., BESSY, S.: *Arc-disjoint flows in networks*, Theoretical Computer Science, **526**, (2014), 28–40.
- [13] BANG-JENSEN, J., HUANG, J., PRISNER, E.: *In-tournament digraphs*, Journal of Combinatorial Theory, Ser. B, **59(2)**, (1993) 267–287.
- [14] BARNES, T. M., SAVAGE C. D.: *A recurrence for counting graphical partitions*, Electronic Journal of Combinatorics, **2**, (1995) #P11, 10 oldal.
- [15] BARNES, T. M., SAVAGE, C. D.: *Efficient generation of graphical partitions*. Discrete Applied Mathematics, **78(1–3)**, (1997) 17–26.
- [16] BARRUS, M. D.: *Hereditary unigraphs and Erdős-Gallai inequalities*, Discrete Mathematics **313(21)**, (2013) 2469–2481.
- [17] BARRUS, M. D., HARTKE, S. G., JAO, K. F., WEST, D. B.: *Length thresholds for graphic lists given fixed largest and smallest entries and bounded gaps*, Discrete Mathematics, **312(9)**, (2012) 1294–1501.
- [18] BEASLEY, L. B., BROWN D. E., REID, K. B.: *Extending partial tournaments*. Mathematical Computer Modelling **50(1)**, (2009) 287–291.
- [19] BEGE, A., KÁSA, Z.: *Algoritmikus kombinatorika és számelmélet*, Presa Universittară Clujeană, 2006, 215 oldal.
- [20] BEHRENS, S., ERBES, C., FERRARA, M., HARTKE, H. S., REINIGER, B., SPINOZA, H., THOMLINSON, C.: *New results on degree sequences of uniform hypergraphs*, *Electron. J. Comb.*, **20(4)**, (2013) #P14, 18 oldal.

- [21] BEINEKE, L. W., REID, K. B.: *Tournaments*, in: Selected Topics in Graph Theory, Academic Press (Harcourt Brace Jovanovich Publishers), London, 1978, 169–204.
- [22] BELL, J. P., BENDER, E. A., CAMERON, P. J., RICHMOND, L. B.: *Asymptotics for the probability of connectedness and the distribution of number of components*, Electronic Journal of Combinatorics, **7** (2000), #R33, 33 oldal.
- [23] BENDER, E. A., CANFIELD, E. R.: *The asymptotic number of labeled graphs with given degree sequences*. Journal of Combinatorial Theory, Series A **24**(3), (1978) 296–307.
- [24] BENDER, E. A., CANFIELD, E. R., MCKAY, B. D.: *The asymptotic number of labeled connected graphs with a given number of vertices and edges*. Random Structures & Algorithms, **1**(2), (1990) 127–170.
- [25] BENDER, E. A., CANFIELD, E. R., MCKAY, B. D.: *Asymptotic properties of labeled connected graphs*. Random Structures & Algorithms, **3**(2), (1992) 183–202.
- [26] BENDER, E. A., GAO, Z.: *Asymptotic enumeration of labelled graphs by genus*, Electronic Journal of Combinatorics, **18**(1), (2011) #P13, 28 oldal.
- [27] BERGE, C.: *Graphs and Hypergraphs*, North Holland, 1973.
- [28] BERGE, C.: *Graphs* (harmadik kiadás), North Holland, 2001 (első kiadás: 1989).
- [29] BERGER, A.: *A note on the characterization of digraph sequences*, arXiv, arXiv:1112.1215v1 [math.CO] (6 December 2011), 6 oldal.
- [30] BERGER, A., MÜLLER-HANNEMANN, M.: *Uniform sampling of digraphs with a fixed degree sequence*, in: (szerk. D. M. Thilikos) WG2010, LNCS **6410**, (2010) 220–231.
- [31] BERGER, A., MÜLLER-HANNEMANN, M.: *How to attack the NP-complete dag realization problems in practice*, arXiv, arXiv:1203.36v1, 2012, 20 oldal.
- [32] BÖDEI, N.: *Gráfok foksorozatai*, Diplomamunka (témavezető Frank András), ELTE Operációkutatási Tanszék, Budapest, 2010, 43 oldal.
- [33] BOLLOBÁS, B.: *The distribution of the maximum degree of a random graph*, *Discrete Mathematics*, **32**(2), (1980) 201–203.
- [34] BOLLOBÁS, B.: *A probabilistic proof of an asymptotic formula for the number of labelled regular graphs*, European Journal of Combinatorics, **1**(4) 4 (1980) 311–316.
- [35] B. BOLLOBÁS: *Degree sequences of random graphs*, *Discrete Mathematics*, **33**(1), 1 (1981) 1–19.
- [36] BRUALDI, A. R., FRISTCHER, E.: *Tournaments associated with multigraphs*, *Discrete Mathematics*, 2014, 17 oldal **526**(2), (2014) 28–40.
- [37] BRUALDI, A. R., KIERNAN, K.: *Landau's and Rado's theorems and partial tournaments*, Electronic Journal of Combinatorics, **16**, #N2, 2009, 6 oldal.
- [38] BRUALDI, A. R., LI, Q.: *The interchange graph of tournaments with the same score vector*. in: Progress in Graph Theory (Waterloo, ON, 1982), Academic Press, Toronto, ON, 1984, 129–151.
- [39] BRUALDI, R. A., MICHAEL, T. S.: *The class of 2-multigraphs with a prescribed degree sequence*, Linear and Multilinear Algebra, **24**(2), (1989) 81–102.

- [40] BRUALDI, A. R., SHEN, J.: *Landau's inequalities for tournament scores and a short proof of a theorem on transitive sub-tournaments*, Journal of Graph Theory, **38**, (2001) 244–254.
- [41] BRUNSON, J. C.: *The S -metric, the Beichl-Croteaux approximation and preferential attachment*, *arXiv:1308.4067v1 [math.CO]* 19 August 2013, 14 oldal.
- [42] BURNS, J. M.: *The number of degree sequences of graphs*, PhD disszertáció, The MIT, 2007, 62 oldal.
- [43] BUSCH, A. H., CHEN, G., JACOBSON, M. S.: *Transitive partitions in realizations of tournament score sequences*, Journal of Graph Theory, **64(1)**, (2010) 52–62.
- [44] BUSCH, A. H., FERRARA, M. J., HARTKE, S. G., JACOBSON, M. S., KAUL, H., *Packing of graphic n -tuples*, Journal of Graph Theory, **70**, (2012) 29–39.
- [45] CHOUDUM, S. A., *A simple proof of the Erdős-Gallai theorem on graph sequences*, Bulletin of the Australasian Mathematical Society, **33**, (1986) 67–70.
- [46] CHUNGPHAISAN, V.: *Conditions for sequences to be r -graphic*, Discrete Mathematics, **7**, (1974) 31–39.
- [47] CODISH, M., MILLER, A., PROSSER, P.: *Breaking symmetries in graph representation*, in: Proceedings of Twenty Third Int. Join Conf. on Art. Int. (IJCAI), 2013, 510–516.
- [48] COHEN, N.: *Number of distinct degree sequences among all n -vertex graphs with no isolated vertices: new values for $n = 20, 21, 22$, and 23*, in: ed. by N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences, 2012, <http://oeis.org/A095268>.
- [49] CORMEN, T. H., LEISERSON, CH. E., RIVEST, R. L., STEIN, C.: *Introduction to Algorithms* (harmadik kiadás), The Press McGraw Hill, Cambridge/New York, 2009.
- [50] CRUSE, A. B.: *On linear programming duality and Landau's characterization of tournament scores*, Manuscript, based on the talk presented on the 57th Annual Meeting of the American Mathematical Society, Southeastern Section, held March 31–April 1, 1978, at Clemson University, Clemson, SC, 18 oldal.
- [51] CRUSE, A. B.: *On linear programming duality and Landau's characterization of tournament scores*, Acta Universitatis Sapientiae, Informatica **6(1)**, (2014) 21–32.
- [52] DANIELS, H. E.: *Round-robin tournament scores*, Biometrika, **56** (1969) 295–299.
- [53] 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), 7 oldal.
- [54] DIMITROV, D., *Efficient computation of trees with minimal atom-bound connectivity index*, *arXiv:1305.1155v2 [cs.DM]* 4 October 2013, 13 oldal.
- [55] ENTRINGER, R. C., TOLMAN, L. K.: *Characterizations of graphs having orientations satisfying local degree restrictions*, Czechoslovak Mathematical Journal, **28(1)**, (1978) 108–119.
- [56] ERDŐS, P., GALLAI, T.: *Gráfok előírt fokszámú pontokkal*, Matematikai Lapok, **11**, (1960) 264–274.
- [57] ERDŐS, P., RICHMOND L. B.: *On graphical partitions*. Combinatorica **13(1)**, (1993) 57–63.

- [58] ERDŐS, P., MOON, P. J. W.: *On sets of consistent arcs in a tournament*. Canadian Mathematical Bulletin, **8**, (1965) 269–271.
- [59] ERDŐS, L. P., KIRÁLY, Z., MIKLÓS, I.: *On the swap-distances of different realizations of a graphical degree sequence*, arXiv, arXiv:1205.2842v1 [math.CO] 13 May 2012, 20 oldal.
- [60] ERDŐS, L. P., KISS, S. Z., MIKLÓS, I.: *On the swap-distances of different realizations of a graphical degree sequence*, Combinatorial Probability and Computations, **22(3)**, (2013) 366–383.
- [61] ERDŐS, L. P.: KIRÁLY, Z., MIKLÓS, I., SOUKUP, L.: *Constructive sampling and counting graphical realizations of restricted degree sequences*, arXiv arXiv:13017523v3[math.CO], 24 oldal.
- [62] ERDŐS, L. P.: MIKLÓS, I., TOROCZKAI, Z.: *A simple Havel-Hakimi type algorithm to realize graphical degree sequences of directed graphs*, Electronic Journal of Combinatorics, **17(1)**, (2010), R66, 10 oldal.
- [63] ERDŐS, L. P., MIKLÓS, I., TOROCZKAI, Z.: *A decomposition based proof for fast mixing of a Markov chain over balanced realizations of a joint degree matrix*, arXiv, arXiv:1307.5295v1 [math.CO], 2013, 18 oldal.
- [64] FORD, L. R., SELMER, S. M.: *A tournament problem*, The American Mathematical Monthly, **66(5)**, (1959) 387–389.
- [65] FRANK, A.: *Connections in Combinatorial Optimization*, Oxford University Press, Oxford, 2011.
- [66] GARG, A., GOEL, A., TRIPATHI, A.: *Constructive extensions of two results on graph sequences*. Discrete Applied Mathematics, **159(17)**, (2011) 2170–2174.
- [67] GERVACIO, S. V.: *Score sequences: lexicographic enumeration and tournament construction*, Discrete Mathematics, **72(1–3)**, (1988) 151–155.
- [68] GERVACIO, S. V.: *Construction of tournaments with a given score sequence*, Southeast Asian Bulletin of Mathematics, **17(2)**, (1993) 151–155.
- [69] GRAHAM, R. L., SPENCER, J. H.: *A constructive solution to a tournament problem*, Canadian Mathematical Bulletin, **14** (1971) 45–48.
- [70] GRIGGS, J. R., REID, K. B.: *Landau’s theorem revisited*, Australasian Journal of Combinatorics, **20**, (1999) 19–24.
- [71] GROSS, J. L., YELLEN, J., ZHANG, P.: *Handbook of Graph Theory*, CRC Press, Boca Raton, FL, 2013, XIX + 1610 oldal.
- [72] GUIDULI, B., GYÁRFÁS, A., THOMASSÉ, S., WEIDL, P.: *2-partition-transitive tournaments*, Journal of Combinatorial Theory, Series B, **72(2)**, (1998), 181–196.
- [73] GYÁRFÁS, A.: *Transitive tournaments and self-complementary graphs*, Journal of Combinatorial Theory, Series B, **38(2)**, (1998) 111–112.
- [74] HAKIMI, S. L.: *On the realizability of a set of integers as degrees of the vertices of a simple graph*. Journal of SIAM Applied Mathematics, **10**, (1962) 496–506.
- [75] HAKIMI, S. L.: *On the degrees of the vertices of a graph*, Journal of the Franklin Institute, **279(4)**, (1965) 290–308.

- [76] HARARY, F., MOSER, L.: *The theory of round robin tournaments*, The American Mathematical Monthly, **73**, (1966) 231–246.
- [77] HARARY, F., PALMER, E. M.: *Graphical Enumeration*, Academic Press, New York and London, 1973, 271 oldal.
- [78] HAVEL, V.: A remark on the existence of finite graphs (cseh), *Časopis Pěstovaný Matematika* **80**, (1955) 477–480.
- [79] HELL, P., KIRKPATRICK, D.: *Linear-time certifying algorithms for near-graphical sequences*, Discrete Mathematics **309(18)**, (2009) 5703–5713.
- [80] HEMASINHA, R.: *An algorithm to generate tournament score sequences*, Mathematical and Computer Modelling, **37(1–3)**, (2003), 377–382.
- [81] HOLSHOUSER, A., MOON, J. W., REITER, H.: *Win-loss sequences for generalized roundrobin tournaments*, Missouri Journal of Mathematical Science, **23(2)**, (2011), 103–207.
- [82] G. Isaak, D. B. West, The edge-count criterion for graphic lists, *Electronic Journal of Combinatorics*, **17** (2010) N#36, 5 oldal.
- [83] IVÁNYI, A.: *Versenyek pontsorozatai*, in: 25. Magyar Operációkutatási Konferencia (Debrecen, 2001. október 17–20), 52–52.
- [84] IVÁNYI, A.: *Reconstruction of complete interval tournaments*, Acta Universitatis Sapientiae, Informatica, **1(1)**, (2009) 71–88.
- [85] IVÁNYI, A., *Reconstruction of complete interval tournaments II*, Acta Univ. Sapientiae, Mathematica, **2(1)**, (2010) 47–71.
- [86] 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, 2012. <http://lemon.cs.elte.hu/egres/open/>.
- [87] IVÁNYI, A., *Degree sequences of multigraphs*, Annales Universitatis Scientiarum de Rolando Eötvös Nominatae, Section Computatorica **37**, (2012) 195–214.
- [88] IVÁNYI, A., ELEK J.: *Score sets in multitournaments II. Simulation results*, **LIX(1)**, (2014) 150–164.
- [89] IVÁNYI, A., LUCZ, L., *Multigráfok foksorozatai*, Alkalmazott Matematikai Lapok, **29**, (2012) 1–54.
- [90] IVÁNYI, A., LUCZ, L., GOMBOS, G., MATUSZKA, T.: *Parallel enumeration of degree sequences of simple graphs II*, Acta Universitatis Sapientiae, Informatica **5(2)**, (2013) 245–270.
- [91] IVÁNYI, A., LUCZ, L., MATUSZKA, PIRZADA, S.: *On the Erdős-Gallai and Havel-Hakimi algorithms*, Acta Universitatis Sapientiae, Informatica **3(2)**, (2011) 230–268.
- [92] IVÁNYI, A., LUCZ, L., GOMBOS, G., MATUSZKA, T.: *$C(2n + 1, n + 1)$: number of ways to put $n + 1$ indistinguishable balls into $n + 1$ distinguishable boxes = number of $(n + 1)$ -st degree monomials in $n + 1$ variables = number of monotone maps from $1 \dots n + 1$ to $1 \dots n + 1$* , in (ed. N. J. A. Sloane): The On-Line Encyclopedia of the Integer Sequences, 2013. <http://oeis.org/A001700>.

- [93] IVÁNYI, A., LUCZ, L., GOMBOS, G., MATUSZKA, T.: *The number of degree-vectors for simple graph*. in (ed. N. J. A. Sloane): The On-Line Encyclopedia of the Integer Sequences, 2013. <http://oeis.org/A004251>.
- [94] IVÁNYI, A., LUCZ, L., MATUSZKA, T., PIRZADA, S.: *Parallel enumeration of degree sequences of simple graphs*, Acta Universitatis Sapientiae, Informatica, **4(2)**, (2012) 260–288.
- [95] IVÁNYI, A., LUCZ, L., MÓRI, T. F., SÓTÉR, P., *On the Erdős-Gallai and Havel-Hakimi algorithms*, Acta Universitatis Sapientiae, Informatica **3(2)**, (2011) 230–268.
- [96] IVÁNYI, A., LUCZ, L., MÓRI F. T., SÓTÉR, P.: *The number of degree-vectors for simple graphs*, in: ed. by N. J. A. Sloane, The On-Line Encyclopedia of Integer Sequences, 2011. <http://oeis.org/A004251>.
- [97] IVÁNYI, A., PIRZADA, S.: *Comparison based ranking*, in: Algorithms of Informatics, Vol. 3 (ed. A. Iványi), AnTonCom, Budapest 2011, 1209–1258.
- [98] IVÁNYI, A., PIRZADA, SHAH, N. A.: *Imbalances of bipartite multitournaments*, Annales Univ. Sci. Budapest., Sect. Comput. **37** (2012), 215–228.
- [99] IVÁNYI, A., SCHOENFIELD, J. E.: *Deciding football sequences*. Acta Universitatis Sapientiae, Informatica **4(1)**, (2012) 130–183.
- [100] JOHNSON, R. H.: *Properties of unique realizations—a survey*, Discrete Mathematics, **3(1)**, (1980) 185–192.
- [101] KÁSA, Z., ANISIU, M.-C.: *Szavak bonyolultsága*, in: Informatikai algoritmusok, 3. kötet, Mondat Kft., Vác, 2013, 1697–1739.
- [102] KÁSA, Z., ANISIU, M.-C., ANISIU, V.: *Properties of palindromes in finite words* Pure Mathematics and Applications, **17(3-4)**, (2006) 183–195.
- [103] KEMNITZ, A., DULFF, S.: *Score sequences of multitournaments*, Congressus Numerantium, **127**, (1997) 85–95.
- [104] KHAN, M. A.: *Equal sum sequences and imbalance sets of tournaments*, Arxiv, arxiv:14022.2456v1 math.CO 11 February 2014, 18 oldal.
- [105] KIM J. H., PITTEL, B.: *Confirming the Kleitman-Winston conjecture on the largest coefficient in a q -Catalan number*, Journal Combinatorial Theory, Series A, **99(2)**, (2000) 197–206.
- [106] KIM, J. H., TOROCZKAI, Z., MIKLÓS, I., ERDŐS, P. L., SZÉKELY, L. A.: *Degree-based graph construction*. Journal of Physics: Mathematical Theory A **42(39)**, (2009) 392–401.
- [107] KIRÁLY, Z.: *Recognizing graphic degree sequences and generating all realizations*, Egres Technical Reports, TR-2011-11, April 23, 2012, 12 oldal.
- [108] KIRÁLY, Z., SZIGETI, Z.: *Simultaneous well-balanced orientations of graphs*, J. Combin. Theory Ser. B **96** (2006) 684–692.
- [109] KLEITMAN, J. *The number of score sequences for a large number of players*, in: Combinatorial Structures and Their Applications (ed. R. K. Guy et al.), Gordon and Breach, New York (1970), 209–213.
- [110] KLEITMAN, D. J., WANG, D. L.: *Algorithms for constructing graphs and digraphs with given valencies and factors*. Discrete Mathematics **6**, (1973) 79–88.

- [111] KLEITMAN, D. J., WINSTON K. J.: *Forests and score vectors*, *Combinatorica* **1(1)**, (1981) 49–54.
- [112] KNUTH, D. E.: *The Art of Computer Programming. Volume 4A, Combinatorial Algorithms*, Addison-Wesley, Upper Saddle River, 2011, XVI + 883 oldal.
- [113] KOHNERT, A.: *Dominance order and graphical partitions*. *Electronic Journal of Combinatorics*, **11(1)**, (2004) No. 4. 17 oldal.
- [114] KOREN, M.: *Sequences with a unique realization by simple graphs*, *Journal of Combinatorial Theory, Series B*, **21(3)**, (1976) 235–244.
- [115] KOVÁCS, G. ZS., PATAKI, N.: *Fokszorozatok elemzése*, TDK dolgozat, ELTE, Informatikai Kar, Budapest 2002. 38 oldal.
- [116] LAMAR, M. D.: *Algorithms for realizing degree sequences of directed graphs*. arXiv, 2010. <http://arxiv.org/abs/0906.0343>.
- [117] LANDAU, L. G., *On dominance relations and the structure of animal societies. III. The condition for a score sequence*, *Bulletin of Mathematics of Biophysics* **15**, (1953) 143–148.
- [118] LIBRANDI, V.: *Number of bracelets (turn over necklaces) with n red, 1 pink and $n - 1$ blue beads for $n = 1, \dots, 1000$* , in (szerk. N. J. A. Sloane): *The On-Line Encyclopedia of the Integer Sequences*. 2012. <http://oeis.org/A005654/b005654.txt>.
- [119] LU, X, BRESSAN, S.: *Generating random graph sequences*, in (ed. J. X. Yu, M. H. Kim, R. Unland): *DASFAA2011, Part I*, LNCS **6587**, Springer-Verlag, Berlin/Heidelberg, 2011, 570–579.
- [120] LUCZ, L.: *Párhuzamos Erdős-Gallai algoritmus*. TDK dolgozat, ELTE IK, Budapest (2011). Elérhető: <http://people.inf.elte.hu/lulsaai/Holz hacker/TKD/>.
- [121] LUCZ, L.: *Gráfok fokszorozatainak elemzése*, Programtervező informatikus diplomamunka, ELTE IK, Budapest, 2012. Elérhető: <http://people.inf.elte.hu/lulsaai/diploma>.
- [122] LUCZ, L.: *Párhuzamos Erdős-Gallai algoritmus*, TDK dolgozat, ELTE Informatikai Kar, Budapest, 2013. 31 oldal.
- [123] LUCZ, L.: *Football league numbers: the possible point series for a league of n teams playing each other twice*. OEIS, A064422 számú sorozat. Elérhető: <http://oeis.org/A064422>.
- [124] LUCZ, L.: *Football league numbers with distinct point totals*. OEISA209467 számú sorozat, Elérhető: <http://oeis.org/A209467>.
- [125] LUCZ, L., SÓTÉR, P.: *Fokszorozatok ellenőrző algoritmusok*. TDK dolgozat. ELTE IK, Budapest, 2011. Elérhető: <http://people.inf.elte.hu/lulsaai/Holz hacker/TKD/>.
- [126] MAHMOODIAN, E. S., *A critical case method of proof in combinatorial mathematics*, *Bulletin of the Iranian Mathematical Society*, **8**, (1978) 1L–26L.
- [127] MATUSZKA, T.: *Fokszorozatokkal kapcsolatos programok és eredmények*, 2012. <http://people.inf.elte.hu/tomintt/DegreeSeq>.
- [128] MATUSZKA: *Programs and Results Connected with Degree Sequences*. ELTE IK, Budapest, 2013. Elérhető: <http://people.inf.elte.hu/tomintt/DegreeSeq>.

- [129] MCKAY, B. D., WANG, X.: *Asymptotic enumeration of tournaments with a given score sequence*, Journal of Combinatorial Theory, Series A, **73(1)**, (1996) 77–90.
- [130] MEIERLING, D., VOLKMANN, L.: *A remark on degree sequences of multigraphs*. Mathematical Methods of Operations Research **69(2)**, (2009) 369–374.
- [131] METROPOLIS, N., STEIN, P. R.: *The enumeration of graphical partitions*. European Journal of Combinatorics, **1(2)**, (1980) 139–153.
- [132] MIKLÓS, I., ERDŐS, P. L., SOUKUP, L., *Towards random uniform sampling of bipartite graphs with given degree sequence*, Electronic Journal of Combinatorics **20(1)**, (2013), Article 16, 51 oldal.
- [133] MIKLÓS, I., PODANI, J.: *Randomization of presence-absence matrices: comments and new algorithms*, Ecology, **85(1)**, (2004) 86–92.
- [134] MILLER, J. W. *Reduced criteria for degree sequences*, Discrete Mathematics, **313(4)** (2013) 550–562.
- [135] MILLER, A., PROSSER, P.: *Diamond free degree sequences*, Acta Universitatis Sapientiae, Informatica, **4(2)**, (2012) 189–200.
- [136] MILO, R., KASHTAN, N., ITZKOVITZ, S., NEWMAN, M. E. J., ALON, U.: *On the uniform generation of random graphs with prescribed degree sequences*, *arXiv*, arXiv:cond-mat/0312028v2 [cond-mat.stat-mech], 2004, 4 oldal.
- [137] MOON, J. W.: *On the score sequence of an n -partite tournament*, Canadian Mathematical Bulletin, **5**, (1962) 51–58.
- [138] MOON, J. W.: *An extension of Landau's theorem on tournaments*, Pacific Journal of Mathematics, **13**, (1963) 1343–1345.
- [139] MOON, J. W.: *Topics on Tournaments*, Holt, Rinehart and Wilson, New York, 1968, 104 oldal.
- [140] MOON, J. W., MOSER, L.: *Almost all tournaments are irreducible*, Canadian Mathematical Bulletin, **5**, (1962) 61–65.
- [141] NARAYANA, T. V., BENT, D. H.: *Computation of the number of score sequences in round-robin tournaments*, Canadian Mathematical Bulletin **7(1)**, (1964) 133–136.
- [142] NARAYANA, T. V., MATSEN, R. M., SARANGI, J.: *An algorithm for generating partitions and its applications*, Journal of Combinatorial Theory, **11**, (1971) 54–61.
- [143] NOE, T. D.: *Table of $a(n)$ for $n = 1, \dots, 100$* , in (ed. N. J. A. Sloane): The On-Line Encyclopedia of the Integer Sequences. 2010. <http://oeis.org/A001700>.
- [144] NOE, T. D.: *Table of binomial coefficients $C(2n - 1, n)$* , in (ed. N. J. A. Sloane): The On-Line Encyclopedia of the Integer Sequences. 2012, <http://oeis.org/A001791>.
- [145] ÖZKAN, S.: *Generalization of the Erdős-Gallai inequality*, Ars Combinatoria **98**, (2011) 295–302.
- [146] PÁLVÖLGYI, D.: *Deciding soccer scores and partial orientations of graphs*, Acta Universitatis Sapientiae, Mathematica, **1(1)**, (2009) 35–42.

- [147] PARKER, E. T., REID, K. B.: *Disproof of a conjecture of Erdős and Moser on tournaments*, Journal of Combinatorial Theory, **9**, (1970) 225–238.
- [148] PATRINOS, A. N., HAKIMI, S. L.: *Relations between graphs and integer-pair sequences*. Discrete Mathematics, **15(4)**, (1976) 347–358
- [149] PÉCSY, G., SZŰCS, L.: *Parallel verification and enumeration of tournaments*, Studia Universitatis Babeş-Bolyai, Informatica, **45(2)**, 2 (2000) 11–26.
- [150] PEHLIVAN, H., NABIYEV, V. N.: *Score calculation from final tournament tables*, Computers & Operations Research, **36(3)**, (2009) 936–950.
- [151] PIRZADA, S.: *An Introduction to Graph Theory*. Universities Press, Hyderabad, 2012, VIII + 396 oldal.
- [152] PIRZADA, S., IVÁNYI, A.: *Minimal digraphs with given imbalance sequence*. Acta Universitatis Sapientiae, Mathematica, **4(1)**, (2012), 86–101.
- [153] PIRZADA, S., NAIKOO, T. A., SHAH, N. A., *Score sequences in oriented graphs*, Journal of Applied Mathematics & Computing, **23(1–2)**, (2007) 257–268.
- [154] PIRZADA, S., NAIKOO, T. A., SAMEE, U., IVÁNYI, A.: *Imbalances in multidigraphs*, Acta Universitatis Sapientiae, Mathematica **2(2)**, (2010) 133–145.
- [155] REID, K. B.: *Tournaments: scores, kings, generalizations and special topics*, Congressus Numerantium, **115**, (1996) 171–211.
- [156] REID, K. B.: *Tournaments*, in Handbook of Graph Theory (harmadik kiadás, szerk. Gross, J. L., Yellen, J., Zhang, P.), CRC Press, Boca Raton, 2013, 196–225.
- [157] REID, K. B., ZHANG, C. Q.: *Score sequences of semicomplete digraphs*, Bulletin of Institute of Combinatorial Applications, **24**, (1998) 27–32.
- [158] RØDSETH, Ø. J., SELLERS, J. A., TVERBERG, H.: *Enumeration of the degree sequences of non-separable graphs and connected graphs*. European Journal of Combinatorics **30(5)**, 1309–1319.
- [159] ROYLE, G.: *Is it true that $a(n+1)/a(n)$ tends to 4?* In (szerk. N. J. A. Sloane): The On-Line Encyclopedia of the Integer Sequences. 2006. <http://oeis.org/A095268>.
- [160] RUBINSTEIN, A.: *Ranking the participants in a tournament*, SIAM Journal on Applied Mathematics, **38(1)**, (1980) 108–111.
- [161] RUSKEY, F., COHEN, F. R., EADES, P., SCOTT, A.: *Alley CATs in search of good homes*. Congressus Numerantium, **102**, (1994) 97–110.
- [162] RYSER, H. J.: *Combinatorial properties of matrices of zeros and ones*, Canadian Journal of Mathematics, **9**, (1957), 371–377.
- [163] RYSER, H. J.: *Matrices of zeros and ones in combinatorial mathematics*, in: Recent Advances in Matrix Theory (Proceedings of an Advanced Seminar, Mathematics Research Center, U.S. Army, University of Wisconsin, Madison, WI, 1963), University of Wisconsin Press, Madison, WI, 1964, 103–124.
- [164] SAH, P., SINGH, L. O., CLAUSET, A., BANSAL, S.: *Exploring community structure in biological networks with random graphs*, bioRxiv, 2013, 19 oldal.

- [165] SCHOENFIELD, J. E., The number of football score sequences, in: ed. by N. J. A. Sloane, *The On-Line Encyclopedia of Integer Sequences*, 2008. <http://oeis.org/A064626>.
- [166] SHUO-YEN, R. L. *Graphic sequences with unique realization*, Journal of Combinatorial Theory, Series B, **19**, (1975) 42–68.
- [167] SIERKSMA, G., HOOGEVEEN, H.: *Seven criteria for integer sequences being graphic*, Journal of Graph Theory, **15(2)**, (1991) 223–231.
- [168] SIKLÓSI, B.: *Sportproblémákat megoldó soros és párhuzamos algoritmusok összehasonlítása*. Programtervező matematikusi diplomamunka. ELTE, Informatikai Kar, Budapest, 2001.
- [169] SLOANE, N. J. A.: *The number of ways to put $n + 1$ indistinguishable balls into $n + 1$ distinguishable boxes*, in (szerk. N. J. A. Sloane): *The On-line Encyclopedia of the Integer Sequences*. (ed. by N. J. A. Sloane). <http://oeis.org/A004251>.
- [170] SLOANE, N. J. A.: *The number of degree-vectors for simple graphs*. In: *The On-Line Encyclopedia of the Integer Sequences* (szerk. N. J. A. Sloane). <http://oeis.org/A004251>.
- [171] SLOANE: *The number of bracelets with n red, 1 pink and $n - 1$ blue beads*, in (szerk. N. J. A. Sloane): *The On-Line Encyclopedia of the Integer Sequences*. (szerk. N. J. A. Sloane). <http://oeis.org/AA005654>.
- [172] SLOANE, N. J. A., PLOUFFE, S.: *The Encyclopedia of Integer Sequences*, Academic Press, 1995.
- [173] SOROKER, S.: *Fast Parallel Algorithms for Graphs and networks*, PhD értekezés, University of California, Berkeley, CA, 1987, 104 oldal.
- [174] SOROKER, S.: *Optimal parallel construction of prescribed tournaments*, Discrete Applied Mathematics, **29(1)**, (1990) 113–125.
- [175] SPENCER, J.: *Optimal ranking of tournaments*, Networks, **1**, (1971/1972) 135–138.
- [176] STOCKMEYER, P. K.: *The falsity of the reconstruction conjecture of tournaments*, Journal of Graphs, **1(1)**, (1977) 19–25.
- [177] TAKAHASHI, M., *Optimization Methods for Graphical Degree Sequence Problems and their Extensions*, PhD thesis, Waseda University, Tokyo, 2007. <http://hdl.handle.net/2065/28387>.
- [178] TANENBAUM A. S., WETHERALL, D. J.: *Computer Networks* (5. kiadás), Prentice Hall, 2010.
- [179] TETALI, P. *Unique tournaments*, Journal of Combinatorial Theory, Series B, **72(1)**, (1998), 157–159.
- [180] THOMASSEN, C.: *Landau's characterization of the tournament score sequences*, in: *The Theory and Applications of Graphs* (Kalamazoo, MI, 1980, szerk. G. Chartrand et al.) Wiley, New York, 1981, 589–591.
- [181] TRIPATHI, A., TYAGY, H.: *A simple criterion on degree sequences of graphs*. Discrete Applied Mathematics **156(18)**, 18 (2008) 3513–3517.
- [182] TRIPATHI, A., VIJAY, S.: *A note on a theorem of Erdős & Gallai*. Discrete Mathematics **265(1–3)**, (2003) 417–420.

- [183] TRIPATHI, A., VENUGOPALAN, S., WEST, D. B.: *A short constructive proof of the Erdős-Gallai characterization of graphic lists*. Discrete Mathematics **310**(4), (2010) 833–834.
- [184] TYSHKEVICH, R. I., MELNIKOV O. I., KOTOV, V. M.: *On graphs and degree sequences: a canonical decomposition (orosz)*, *Kibernetika* **6**, (1981) 5–8.
- [185] VOLKMAN, L.: *Strong subtournaments of multipartite tournaments*, Australoasian Journal of Combinatorics, **20**, (1999) 189–196.
- [186] VOLKMAN, L.: *Multipartite tournaments: A Survey*, Discrete Mathematics, **307**, (2007) 3097–3129.
- [187] WEISSTEIN, E. W.: *Degree Sequence*, From MathWorld—Wolfram Web Resource, 2011. MathWorld—Wolfram Web Resource, 2014.
- [188] WEISSTEIN, E. W.: *Graphic Sequence*, From MathWorld—Wolfram Web Resource, 2014.
- [189] WINSTON, K. J., KLEITMAN, D. J.: *On the asymptotic number of tournament score sequences*, Journal of Combinatorial Theory, Series A. **35**, (1983) 208–230.
- [190] YIN, J.-H, LI, J.-S.: *Two sufficient conditions for a graphic sequence to have a realization with prescribed clique size*, Discrete Mathematics, **301**(2–3), (2005), 218–227.
- [191] ZVEROVICH, I. E., ZVEROVICH, V. E.: *Contributions to the theory of graphic sequences*, *Discrete Mathematics*, **105**(1–3), (1992) 293–303.

(Beérkezett: 2014. március 8.)

IVÁNYI ANTAL

tony@inf.elte.hu

Eötvös Loránd Tudományegyetem

Informatikai Kar

1117 Budapest, Pázmány Péter sétány 1/C

KÁSA ZOLTÁN

kasa@ms.sapientia.ro

Sapientia Erdélyi Magyar Tudományegyetem

Matematikai és Informatikai Tanszék

Marosvásárhely/Koronka, Segesvári út 1/C

PARALLEL ENUMERATION OF DEGREE SEQUENCES

ANTAL IVÁNYI, ZOLTÁN KÁSA

The problem of testing, reconstruction and enumeration of the degree sequences of simple graphs has reach bibliography. In this paper we report on the parallel enumeration of the degree sequences of simple graphs resulting the number of sequences for $n = 24, \dots, 31$ vertices.

Alkalmazott Matematikai Lapok (2014)