

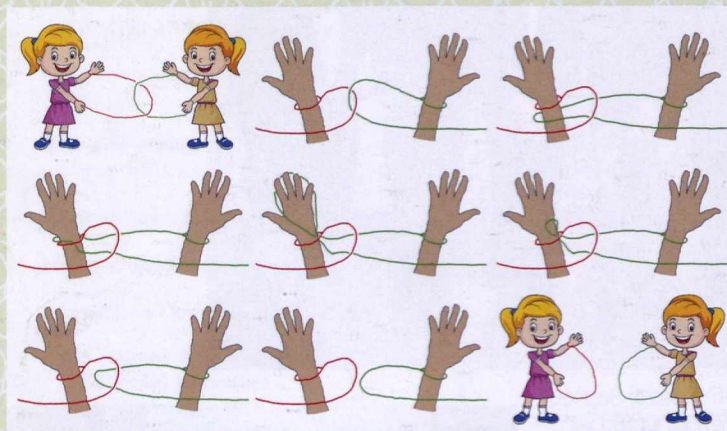
Középiskolai Matematikai és Fizikai Lapok

Informatika rovattal

— 2024 JAN. 17



Kép a G. 838-as feladathoz



Képek a decemberi ördöglakat feladvány megoldásához

Beszámoló a 2023. évi Eötvös-versenyről | Matematika és
informatika – kéz a kézben | Matematika és fizika
képzések az ELTE TTK-n | Rátz Tanár Úr Életműdíj 2023
Ördöglakat az erdőből



KÖNYV

74. évfolyam
1. szám
2024.
január

KÖZÉPISKOLAI MATEMATIKAI ÉS FIZIKAI LAPOK

INFORMATIKA ROVATTAL BŐVÍTVE

ALAPÍTOTTA: ARANY DÁNIEL 1894-ben

74. évfolyam 1. szám

Budapest, 2024. január

Megjelenik évente 9 számban, januártól májusig és szeptembertől decemberig havonta 64 oldalon. ÁRA: 1250 Ft

TARTALOMJEGYZÉK

<i>Makay Géza: Matematika és informatika – kéz a kézben, avagy egy 2023-as Schweitzer példa előzménye</i>	2
Rátz Tanár Úr Életműdíj 2023	9
<i>Kós Géza: Rejtvények, Ördöglakatok – Ördöglakatok az erdőből</i>	10
<i>Tatár Zsuzsanna Mária: Gyakorló feladatsor emelt szintű matematika érettségire</i>	12
<i>Kozma Katalin Abigél: Megoldásvázlatok a 2023/9. szám emelt szintű matematika gyakorló feladatsorához</i>	14
Matematika feladatok megoldása (5305.)	21
A K pontversenyben kitűzött gyakorlatok (794–798.)	27
A C pontversenyben kitűzött gyakorlatok (797–798., 1793–1797.)	28
A B pontversenyben kitűzött feladatok (5358–5365.)	29
Az A pontversenyben kitűzött nehezebb feladatok (869–871.)	30
Matematikai képzések az ELTE TTK-n	31
Matematikatanár-képzés az ELTE TTK-n	32
Informatikából kitűzött feladatok (611–614.) ..	33
Fizika alapszak és fizikatanár-képzés az ELTE TTK Fizikai Intézetében	38
Beszámoló a 2023. évi Eötvös-versenyről	41
Mérési feladatok megoldása (425.)	50
Fizika feladatok megoldása (5499., 5504., 5506.)	54
Fizikából kitűzött feladatok (428., 837–840., 5535–5543.)	58
Problems in Mathematics	61
Problems in Physics	63

Főszerkesztő: KORÁNDI JÓZSEF

Fizikus szerkesztő: VANKÓ PÉTER

Műszaki szerkesztő: FRIED KATALIN

Borító: BURGHARDT ZSUZSA

Kiadja: MATFUND ALAPÍTVÁNY

Alapítványi képviselő: KÓS RITA

Felelős kiadó: KATONA GYULA

Nyomda: OOK-PRESS Kft.

Felelős vezető: SZATHMÁRY ATTILA

INDEX: 25 450 ISSN 1215-9247

A matematika bizottság vezetője:

HERMANN PÉTER

Tagjai: BÍRÓ BÁLINT, GYENES ZOLTÁN,

HUJTER BÁLINT, IMOLAY ANDRÁS,

KISS GÉZA, KÓS GÉZA, KOZMA KATALIN

ABIGÉL, MATOLCSI DÁVID,

ÖKÖRDI PÉTERNÉ, PACH PÉTER PÁL,

RATKÓ ÉVA, SZMERKA GERGELY,

SZTRANYÁK ATTILA, VÍGH VIKTOR

A fizika bizottság tiszteletbeli elnöke:

HOLICS LÁSZLÓ

Vezetője:

SZÉCHENYI GÁBOR

Tagjai: BARANYAI KLÁRA, GNÄDIG PÉTER,

HONYEK GYULA, OLOSZ BALÁZS, SZÁSZ

KRISZTIÁN, VÍGH MÁTÉ, VLADÁR KÁROLY,

WOYNAROVICH FERENC

Az informatika bizottság vezetője:

SCHMIEDER LÁSZLÓ

Tagjai: FODOR ZSOLT, LÓCZI LAJOS,

SIEGLER GÁBOR, TÓTH TAMÁS

Fordítók: GYENES ZOLTÁN, TASNÁDI ANIKÓ

Szerkesztőségi titkár: VÁRDAI KITTI

A szerkesztőség címe: 1117 Budapest,

Pázmány Péter sétány 1/C III. emelet 3.405.

Telefon: + 36 20 320-1143

A lap megrendelhető az Interneten:

www.komal.hu/megrendelolap/reszletek.h.shtml.

Előfizetési díj egy évre: 10 400 Ft

Kéziratokat nem őrzünk meg és nem küldünk vissza.

Minden jog a KöMaL tulajdonosaié.

E-mail: szerk@komal.hu

Internet: <http://www.komal.hu>

This journal can be ordered from the Editorial office:

Pázmány Péter sétány 1/C III. emelet 3.405.

1117-Budapest, Hungary

telephone: +36 20 320-1143

or on the Postal address

H-1518 Budapest 112, P.O.B. 32, Hungary,

or on the Internet:

www.komal.hu/megrendelolap/reszletek.h.shtml.

A Lapban megjelenő hirdetések tartalmáért felelősséget nem vállalunk.



Matematika és informatika – kéz a kézben, avagy egy 2023-as Schweitzer példa előzménye¹

Ebben a cikkben megpróbáljuk bemutatni, hogy hogyan segíthet matematikai problémák megoldásában a számítógép, és hogyan segíthet a matematika memória vagy processzor használat szempontjából optimálisabb és/vagy több esetet is kezelni képes programok megírásában. A programjainkat algoritmusleíró nyelven adjuk meg (eredetileg C/C++ nyelven írtuk meg), reméljük, hogy az olvasó kedvet kap arra, hogy kipróbálja azokat saját kedvenc programozási nyelvén implementálva. A cikkben nem mindig közlünk teljes kódokat, inkább a programok lényegi/érdekes részeit emeljük ki, a „körítés” megírását a felhasználóra bizzuk. Kezdjünk is neki.

A Bolyai János Matematikai Társulat Érintő című elektronikus kiadványának 28. számában (2023. június) [1] jelent meg a Héttusa rovatban a következő

Feladat. *Van-e olyan szám, amelynek pontosan tíz pozitív osztója van a 20-nál nem nagyobb számok között?*

Megoldás. Mivel oszthatóságról, osztókról van szó, az embernek óhatatlanul eszébe jut a prímfelbontás. A keresett számunk legyen

$$p_1^{\alpha_1} p_2^{\alpha_2} \dots p_l^{\alpha_l}$$

alakú, ahol a p_i -k különböző prímek, α_i -k pedig pozitív egészek. Feltehetjük, hogy $p_i^{\alpha_i} \leq 20$, hiszen ha egy prímhatalvány 20-nál nagyobb, akkor minden osztó, amelyben az tényezőként szerepel, szintén 20-nál nagyobb lesz. Kezdjük felépíteni egy egyszerű „mohó algoritmussal”² a számunkat prímhatalványtényezőnként (a legkisebb prímétől kiindulva) úgy, hogy a 20-nál nem nagyobb osztók száma ne lépje túl a 10-et:

$$\begin{aligned} 2^4 &= 16; & 20\text{-nál nem nagyobb osztóinak száma } 5, \\ 2^4 \cdot 3^2 &= 144; & 20\text{-nál nem nagyobb osztóinak száma } 10. \end{aligned}$$

És a feladatot megoldottuk.

A feladat elég könnyen megoldható egyszerű számolással. A Héttusa feladatokra bárki küldhet választ, megoldást. A kitűző (Róka Sándor) szándéka és megfogalmazása szerint a rovatban olyan feladatokat tűznek ki, amelyekre szinte bárki találhat megoldást, de ugyanakkor újabb kérdéseket inspirálhatnak. Ez a feladat is ilyen, kezdjük hát el más módon is megközelíteni és vele kapcsolatos újabb kérdéseket feltenni.

1. kérdés. *Mely számok jöhetnek szóba megoldásként?*

¹Ezt a kutatást a TKP2021-NVA-09 projekt támogatta. A TKP2021-NVA-09 számú projekt a Magyar Innovációs és Technológiai Minisztérium támogatásával valósult meg a Nemzeti Kutatási, Fejlesztési és Innovációs Alapból, a TKP2021-NVA támogatási konstrukció keretében.

²A mohó algoritmus olyan döntéshozó algoritmus, amely minden egyes lépésnél a legjobbnak tűnő döntést hozza a rendelkezésre álló lehetőségek közül, anélkül hogy előretekintene.

Megoldás. Már pedzegettük a kérdést a feladat megoldásában: olyan megoldásokat keresünk, ahol a számnak nincs 20-nál nagyobb p^α osztója, tehát ezek a prímhatalványok jöhetnek szóba:

$$2^4, 3^2, 5, 7, 11, 13, 17, 19.$$

Más szóval a lényegesen különböző megoldások a $2^4 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 = 232\,792\,560$ osztói közül kerülhetnek ki. Az osztókból véges sok van, ellenőrizhető, hogy közülük melyek megoldások (lásd a következő kérdést). Egy megoldásból újabb megoldást kapunk, ha egy olyan prímhatalvánnyal szorozzuk meg, amellyel megszorozva nem keletkezik újabb 20-nál kisebb osztó. Ilyen lehet egy 20-nál nagyobb prímszám, vagy lehet például a 2 valamilyen pozitív egész hatványa is, hiszen ez utóbbi esetben a 2 legalább az 5.-en szerepelne, ami már nagyobb, mint 20. Például a $144 \cdot 2 = 288$ és a $144 \cdot 3 = 432$ megoldások, nem megoldás a $144 \cdot 5 = 720$, de megoldás a $144 \cdot 23 = 3312$. Ezzel a módszerrel megkaphatjuk bármelyik megoldást.

A megoldáshalmaz egy érdekes tulajdonsága, hogy a megoldások $232\,792\,560$ -periodikusan helyezkednek el a számegyenesen. Ez azért teljesül, mert egy megoldás azért megoldás, mert a 2, 3, 5, 7, 11, 13, 17, 19 prímek olyan hatványon szerepelnek benne, amilyeneken, és ez nem változik a $232\,792\,560$ -nal való eltolással. Ebben a kijelentésben azért van egy kis csúsztatás: ha a megoldásban az egyik prímszám maximális (vagy annál nagyobb) hatványon szerepel, akkor az eltolással neki változhat a kitevője, de nem csökkenhet a maximális kitevő alá, és ez a 20-nál kisebb osztók számát nem befolyásolja.

2. kérdés. *232 792 560-nak hány olyan pozitív osztója van, amelynek pontosan tíz pozitív osztója van a 20-nál nem nagyobb számok között?*

Megoldás. $232\,792\,560$ osztóiból az ismert állítás alapján $5 \cdot 3 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 960$ darab van, így ezt már nem szívesen ellenőrizné az ember „kézzel”, érdemes programot írni rá.

De ha már emlegettük az „ismert állítást”, ne essünk neki teljesen naiv módon az osztók keresésének, végignézzük a számokat 1-től $232\,792\,560$ -ig: a trükk az állítás bizonyításában, hogy az osztókban minden prímtényező kitevője lehet bármi 0-tól a számban található kitevőig, hát listázzuk ez alapján az osztókat a programban.

Nyilván tovább is fogunk általánosítani, tehát ne direktben erre a 20-as számra írjuk meg a programunkat, hanem általánosabban: k darab osztót fogunk keresni, amelyek nem nagyobbak n -nél.

Amikor az ember elkezd gondolkodni egy program megírásán, akkor végig kell gondolnia, hogy mit hogyan kíván „modellezni” a programban, mit hogyan kell tárolni, melyik adatot hogyan kell kezelni és milyen algoritmus alapján fog dolgozni a program. Ebben a konkrét esetben:

1. Egyelőre nem akarunk prímszámok keresésére programot írni, „égezzük be” a programunkba az első néhány prímszámot például a <https://oeis.org/A000040> weboldal alapján.
2. A prímszámok hatványainak szorzatával kell számolnunk, ezt úgy tárolhatjuk, hogy egy tömbben megadjuk a megfelelő kitevőt a prímszámokhoz.
3. Az összes lehetséges kitevő-kombinációt ki kell próbálnunk, erre legegyszerűbb egy rekurzív függvényt írni, ami kipróbálja az éppen aktuális prímszám kitevőit és meghívja önmagát a következő prímszámra.

4. És ezt kétszer kell megtennünk: a konkrét esetben a 232 792 560 osztóira kell ellenőriznünk, hogy azok osztóiból 10 darab van-e, amelyek nem nagyobbak, mint 20.

Ezek alapján elkezdhetjük megírni a programot:

```
p : Egészek tömbje := [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53]
NP : Egész := Elemszám(p)
ph, oh, koh : Egészek tömbje
n, k : Egész
```

A p tömbben tároljuk a „beégetett” prímszámainkat, az NP megadja ezek számát. A ph (prím hatvány) tömbben lesz, hogy n -ig melyik prím maximum melyik hatványon szerepelhet, a konkrét $n = 20$ esetben (a fentiek alapján) ebben a tömbben a következő számok lesznek majd:

4, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0.

Az oh (osztó hatvány) tömbben lesznek az osztók kitevői, és a koh (kis osztó hatvány) az osztók osztóinak kitevői.

Ízlések és pofonok különbözőek, ahogy szokták mondani: egy programot lehet alulról felfelé, és felülről lefelé felépíteni. A szerző ez utóbbit preferálja: hiába írjuk meg az egyszerűbb, egy későbbi, összefoglaló algoritmusban felhasználandó függvényeket, ha végül azokból mégsem sikerül összerakni a teljes programot. Úgyhogy kezdjük a fő (main) függvénnyel. Miután bekértük n és k értékét és persze ellenőriztük, hogy azok megfelelőek-e ($1 \leq n \leq 53, 1 \leq k \leq n$) meg kell határoznunk, hogy melyik prím maximum mekkora hatványon szerepelhet:

```
j : Egész := 1
i : Egész := 0
ph[i] := Egész(log(n+0.5)/log(p[i]))
Ciklus amíg i < NP és ph[i] > 0
  Ki: p[i] ^ ph[i]
  j := j * Egész(Kerekít(Hatvány(p[i], ph[i])))
  i := i+1
  Ha i < NP akkor ph[i] := Egész(log(n+0.5)/log(p[i]))
Ciklus vége
Ki: "=" j "osztóit vizsgáljuk"
```

A $ph[i]$ kiszámolásakor lényegében az n adott prím alapú logaritmusának egészrészét vesszük, a +0.5-re azért van szükség, hogy az esetleges kerekítési hibákat kiküszöböljük. (Mottó: „Tévedni emberi dolog, de igazán összekutyulni valamit csak számítógéppel lehet.”) Ellenőrzésképpen kiírjuk, hogy melyik prím maximum hányadik hatványon szerepelhet, és hogy ezeknek mi a szorzata (itt is vigyázva az esetleges kerekítési hibákra). A konkrét $n = 20$ esetben meg is kapjuk a 232 792 560 számot. Ezek után meghívhatjuk a „külső” rekurzív függvényünket, ami meghatározza ezen szám összes osztóját a prímkitevők végigpróbálgatásával:

osz : Egész

Eljárás osztok(s : Egész)

Ha $s < NP$ és $ph[s] > 0$ akkor

oh[s] := 0

Ciklus amíg $oh[s] \leq ph[s]$

osztok(s+1)

oh[s] := oh[s] + 1

Ciklus vége

Különben

osz := 0

Ha osztokszama(0) = k akkor

szam := 1

i := 0

Ciklus amíg $i < NP$ és $ph[i] > 0$

Ha $oh[i] > 0$ akkor

Ki: p[i] "^" oh[i]

szam := szam * Egész(Kerekít(Hatvány(p[i], oh[i])))

Elágazás vége

i := i + 1

Ciklus vége

Ki: "=" szam

Elágazás vége

Elágazás vége

Eljárás vége

A függvény alapértelmezett paramétere, hogy az első (0. indexű) prímszámtól kezdjük a kitevők próbálgatását. A függvény először ellenőrzi, hogy kell-e még kitevőket próbálgatni, ha igen, akkor a megadott prímmel ezt megteszi, és meghívja saját magát a következő prím p tömbbeli indexével. A függvény második részében ellenőrizzük egy még meg nem írt osztokszama függvénnyel, hogy a kapott osztónak hány osztója nem nagyobb, mint n , és ha pontosan k darab, akkor kiírjuk az adott osztó prímtenyezős felbontását és magát az osztót is. És végül az osztokszama függvény, ami az előző alapján már könnyen érthető:

Függvény osztokszama(s : Egész) : Egész

Ha $s < NP$ és $ph[s] > 0$ akkor

koh[s] := 0

Ciklus amíg $koh[s] \leq oh[s]$

osztokszama(s+1)

koh[s] := koh[s] + 1

Ciklus vége

Különben

szam := 1

i := 0

Ciklus amíg $i < NP$ és $ph[i] > 0$

szam := szam * Egész(Kerekít(Hatvány(p[i], koh[i])))

i := i + 1

Ciklus vége

Ha $szam \leq n$ akkor osz := osz + 1

Elágazás vége

osztokszama := osz

Függvény vége

Futtatva ezt a programot $n = 20$ és $k = 10$ bemenő paraméterekkel összesen 84 darab osztót kapunk:

$2^4 \cdot 3^2 \cdot 5^1 \cdot 7^1 \cdot 11^1 \cdot 13^1 \cdot 17^1 \cdot 19^1 = 232792560$ osztóit vizsgáljuk.

$3^2 \cdot 5^1 \cdot 7^1 \cdot 11^1 \cdot 13^1 \cdot 17^1 \cdot 19^1 = 14549535$

$2^1 \cdot 5^1 \cdot 7^1 \cdot 11^1 \cdot 13^1 \cdot 17^1 \cdot 19^1 = 3233230$

$2^1 \cdot 3^1 \cdot 7^1 \cdot 11^1 \cdot 13^1 \cdot 17^1 \cdot 19^1 = 1939938$

...

$2^4 \cdot 3^1 \cdot 11^1 \cdot 13^1 = 6864$

$2^4 \cdot 3^1 \cdot 7^1 = 336$

$2^4 \cdot 3^2 = 144$

Néhány osztót „kézzel” ellenőrizve úgy tűnik, hogy a programban nincs hiba. Hát akkor kísérletezzünk, játsszunk egy kicsit a programmal. Gyorsan kiderül, hogy mégis van valami probléma, ugyanis $n = 23$ esetén a futtatás első sorában ezt látjuk (ez operációs rendszer függő lehet, ha az Egész típus 4 byte-os egészt jelent, akkor nem lesz jó az eredmény):

$2^4 \cdot 3^2 \cdot 5^1 \cdot 7^1 \cdot 11^1 \cdot 13^1 \cdot 17^1 \cdot 19^1 \cdot 23^1 = -2147483648$ osztóit vizsgáljuk.

Pozitív számok szorzataként negatív számot kaptunk: ez matematikailag nyilván lehetetlen. Ez egy tipikus túlsordulási hiba: a tényleges $5\,354\,228\,880$ -os eredmény nem fér bele a 4 byte-os egész számaábrázolásba. Kicserélhetnénk a megfelelő helyeken az Egész típust ún. Hosszú Egész típusra (8 byte-os egész szám), akkor 43-nál fog túlsordulni a dolog. De ha nem ragaszkodunk a kiinduló szám és az osztók kírásához, meglegszünk azok prímtényezőös felbontásával is, akkor egy minimális módosítással a programunk tökéletesen fog működni. Az olvasóra bízunk annak a programrésznek a megtalálását, ahol ezt a módosítást végre kell hajtani ahhoz, hogy a program jó eredményt produkáljon. Ugyancsak érdekes feladat lehet ennek a programnak a felgyorsítása, hiszen már $n = 42$ és $k = 21$ esetén is kb. 15 másodpercig fut a program, amíg megtalálja a 3218 darab megfelelő osztót.

De menjünk tovább.

3. kérdés. Van-e olyan szám, amelynek pontosan k db pozitív osztója van az n -nél nem nagyobb számok között ($k \leq n$)?

Megoldás. A eredeti feladat megoldásában használt mohó algoritmus alapján írhatunk egy programot, ami adott n és k , vagy akár elég sok (n, k) pár esetén ellenőrzi az állítást. Mit kell megváltoztatni az eddigi programunkban ehhez? A fő függvényben végigpróbáljuk a lehetséges (n, k) párokat:

i : Egész

Ciklus n := 1-től p[NP-1]-ig

i := 0

ph[i] := Egész(log(n+0.5)/log(p[i]))

Ciklus amíg i < NP-1 és ph[i] > 0

i := i + 1

ph[i] := Egész(log(n+0.5)/log(p[i]))

Ciklus vége

Ciklus k := 2-től n-ig

Ki: n, k

Ha osztok(0) akkor Ki: "van" Különbén Ki: "nincs"

Ciklus vége

Ciklus vége

Vegyük észre, hogy nem ellenőrizzük a $k = 1$ esetet egyetlen n esetén sem, hiszen pontosan 1 osztót az 1-es szám ad minden n -re (az n -nél nagyobb príme

hatványainak szorzataiként képzett számok ilyenek még, más lehetséges megoldás erre nincs). Az osztok függvény annyiban változik, hogy visszaadja, hogy sikerült-e megfelelő számot találni az adott (n, k) párhoz, és persze átírjuk „mohóra”, vagyis nagy kitevőktől megyünk a kisebbek felé, és az első találatnál kiszállunk:

osz: Egész

Függvény osztok(s) : Logikai

```

Ha s = 0 akkor
  i := 0
  Ciklus amíg i < NP és ph[i] > 0
    oh[i] := 0
    i := i + 1
  Ciklus vége
Elágazás vége
Ha s < NP és ph[s] > 0 akkor
  oh[s] := ph[s]
  Ciklus amíg oh[s] >= 0
    osz := 0
    aktosz := osztokszama(0)
    Ha aktosz = k akkor
      osztok := IGAZ
    Különben
      Ha aktosz < k és osztok(s+1) akkor osztok := IGAZ
  Elágazás vége
  oh[s] := 1
  Ciklus vége
Elágazás vége
osztok := HAMIS
Függvény vége

```

Érdemes végiggondolni, hogy kezdetben itt miért kell 0-val feltölteni az oh tömböt, és miért nem kellett ezt megtenni korábban.

Az osztokszama függvényen viszont el kell végezni azt a bizonyos minimális átalakítást, ami azon kívül, hogy garantálja a helyes működést, gyorsít is a programon. Ez ugyanis az egyetlen hely, ahol ténylegesen ki kell számoljunk egy prímszámokkal megadott számot (hiszen össze kell hasonlítanunk n -el), és emiatt ez túlsordulhat. Ennek megoldása, hogy már a számolási ciklusban ellenőrizzük, hogy túl nagy számot kaptunk-e, és kiszállunk, ha igen:

```

...
szam := 1
i := 0
Ciklus amíg i < NP és ph[i] > 0 és szam <= n
  szam := szam * Egész(Kerekít(Hatvány(p[i], koh[i])))
  i := i + 1
Ciklus vége
Ha szam <= n akkor osz := osz + 1
...

```

A program így módosítva kevesebb, mint másfél másodperc alatt lefut, és azt adja eredményül, hogy $n \leq 53$ -ra minden (n, k) párhoz van megfelelő szám. Itt is lehet egy kis matematikai okoskodással, és az annak megfelelő program-átalakítással gyorsítani: az olvasó gondolja meg, hogy a $k = n$ esetet sem kell ellenőrizni, és azt továbbgondolva egy akár tízszeres gyorsítás is elérhető. De jobb algoritmus is van: olyan, aminél a túlsordulás nem jön elő (legalábbis addig nem, amíg egyáltalán van

türelme az embernek kivárni a programfutást), és el tudunk menni pár másodperc alatt akár $n = 600$ -ig is (házi feladat ☺). Csak összehasonlításképpen: a fent leírt algoritmus szerint $n = 600$ -ra a kiinduló számnak kb. $3 \cdot 10^{35}$ darab osztója van, elvileg azokat kellene ellenőrizni, ami már kivárhatatlan idő lenne. Persze azért nem ennyire rossz a helyzet, hiszen a mohó algoritmus miatt nem ellenőrizzük minden osztót, de a fenti algoritmussal akkor is órákba kerül már csak az $n = 600$ -as eset is, nemhogy addig az összes szám. Mindenesetre az „eredmény”, a sejtés megvan: minden (n, k) párhoz van megfelelő szám, már „csak” be kellene bizonyítani.

Mielőtt nagyon nekiesnénk a bizonyításnak, nézzük meg, mennyire hatékony a mohó algoritmus ennél a problémánál. Módosítsuk egy kicsit az osztók függvényünket, ellenőrizzük le, hogy ha egy adott prímhez felvettük a maximális kitevőt, ami még belefér az elvárt osztók számába, akkor utána kell-e visszább lépünk:

```
...
Különben
  Ha aktosz < k akkor
    Ha osztok(s+1) akkor osztokszama := IGAZ
    Különben Ki: "Visszalépés"
  Elágazás vége
Elágazás vége
...
```

És az derül ki, hogy nem! Vagyis a mohó algoritmus nagyon hatékony ebben az esetben (nem mindig az). Másrészt ez azt is jelenti, hogy a bizonyításunkhoz van egy sejtésünk a megfelelő szám megkonstruálására: a mohó algoritmus szerint osszuk ki a kitevőket. Mondjuk hát ki az állításunkat.

4. állítás. *A fenti mohó algoritmussal lehet találni olyan természetes számot, amelynek pontosan k db pozitív osztója van az n -nél nem nagyobb számok között ($k \leq n$).*

Megoldás. Ennek az állításnak egy egyszerűbb (?) változata volt a 2023. évi Schweitzer Miklós Matematikai Emlékverseny hatodik feladata, amelyben a mohó algoritmusra való utalás nélkül és csak elegendően nagy n -ekre kellett belátni az állítást. A kitűzés és a bizonyítás megtalálható a Bolyai Társulat honlapján ([2]). A 4. Állítás annyival több annál a feladatnál, hogy ezt minden n -re be kell látnunk. A megjelent bizonyítás nem foglalkozik azzal a kérdéssel, hogy konkrétan milyen nagy n -ektől kezdve teljesül az állítás, de óvatosabb becsléseket alkalmazva ellenőrizhető, hogy $n \geq 599$ esetére is adható hasonló bizonyítás. Az állítás bizonyítását a fent említett házi feladat teszi teljessé: ha megírunk egy olyan programot, ami $n \leq 598$ esetén ellenőrzi azt (nem volt véletlen a 600-as szám ☺). Amennyiben az olvasót érdekli a probléma, és talál egy megfelelően gyors algoritmust rá, a program forráskódját elküldheti a hf@server.math.u-szeged.hu e-mail címre 2024. február 15-ig. A leginnovatívabb megoldásokból egy cikket tervezünk írni egy későbbi KöMaL számban.

[1] <https://ematlap.hu/fooldal-2023-2>

[2] <https://www.bolyai.hu/versenyek-schweitzer-miklos-emlekverseny/>

Makay Géza

(SZTE, TTIK, Bolyai Intézet)

e-mail: makayg@math.u-szeged.hu