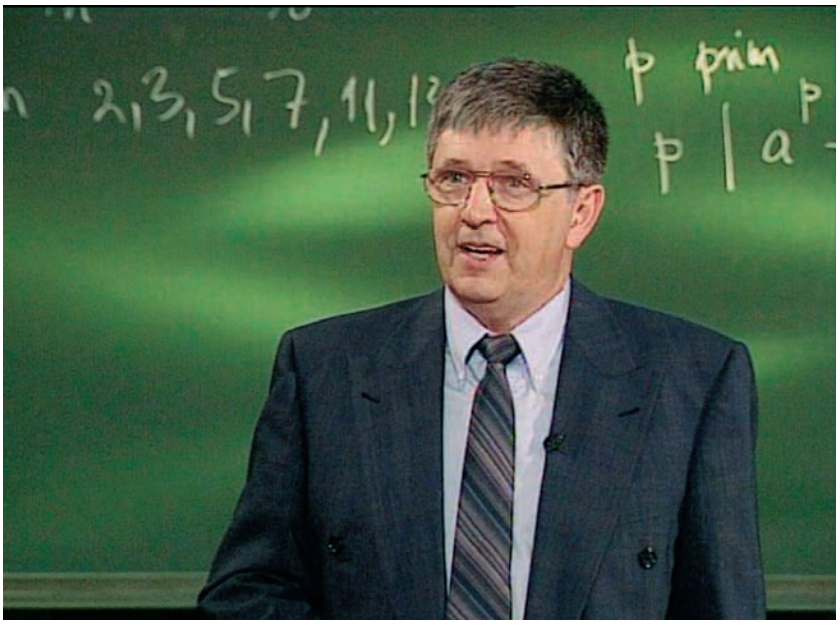
LOVÁSZ LÁSZLÓ

Mit kívánnak a számítógépek a matematikától, és mit adnak neki?



Lovász László
matematikus
az MTA rendes tagja

1948-ban született. A Fazekas Mihály Gyakorló Gimnázium első matematika tagozatos osztályába járt. Szinte minden matematika versenyt megnyert (Ki miben tudós?, KöMaL, OKTV, Nemzetközi Matematikai Diákolimpia, Schweitzer-verseny stb.). 1971-ben végzett az ELTE Természettudományi Karának matematikus szakán. 1970-ben, ötödéves egyetemistaként megvédte a gráfok faktorairól írt kandidátusi disszertációját. Világhírű eredménye a perfekt gráfsejtés bizonyítása volt. 1979-ben megoldotta az információelmélet egyik legnevezetesebb problémáját, a Shannon-problémát. Cikke az év publikációja lett az *IEEE Transaction Information Theory* című folyóiratban. 1979-ben, harmincegy éves korában lett az MTA levelező, 1985-ben pedig rendes tagja. 1999-ben megkapta a matematikusok Nobel-díjának tartott Wolf-díjat.

Pályáját 1971-ben az ELTE tudományos főmunkatársaként kezdte, 1982-től tanszékvezető egyetemi tanár, 1987-től a Princeton Egyetem, majd a Yale Egyetem egyetemi tanára. Számos nemzetközi tudományos folyóirat – az *Acta Mathematica*, az *Advances in Mathematics* szerkesztőbizottságának tagja, a *Combinatorica* főszerkesztője.

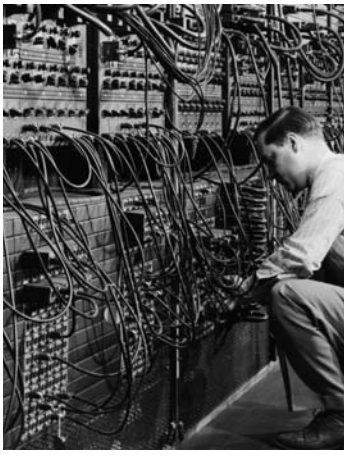
Főbb kutatási területei: kombinatorika, algoritmuselmélet.

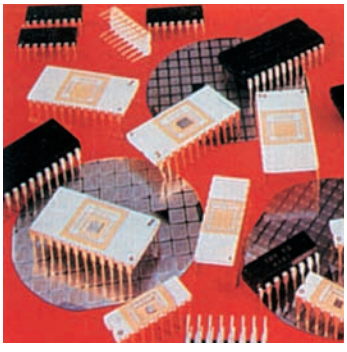
Napjainkban teljesen magától értetődőnek tűnik, hogy munkánk során számítógépet használunk, internetezünk, bankkártyával fizetünk vagy meghallgatunk egy CD-lemezt. Ilyenkor talán végig sem gondoljuk, hogy ezek egyikét sem tehetnénk meg, ha nem jöttek volna létre a matematika új ágai: az algoritmuselmélet, a kriptográfia vagy a bonyolultságelmélet. Gondolnánk-e, hogy adatvédelmünk, a katonai létesítmények, a bankrendszer, s így a gazdaság biztonsága azon is múlik, hogy egy elég nagy szám felbontható-e belátható időn belül prímszámok szorzatára? Miért fontos tudni egy számról, hogy azt véletlen módon adták-e meg nekünk vagy valamilyen algoritmus eredményeként? Milyen régi problémák megoldásában segíthet nekünk a számítógép s melyekben nem? Ezekre az izgalmas, mindannyiunkat érintő kérdésekre keresi a választ az előadás.

Bevezetés

A 20. században az igazi forradalom a számítógépek megjelenésével tört ki. Az elmúlt évszázad második felét gyakran „atomkorszaknak” nevezik, a „számítógépek korszaka” elnevezés azonban pontosabb lenne. Akár egy

Eukleidész görög matematikus, Raffaello freskójának részlete





Neumann János alkotta számítógépben minden egyes rádiócső egy logikai egység; ma az ábrán látható integrált áramkörökben (amelyek közel mérethűek) akár 50 millió ilyen egység is van

évig tartó előadás-sorozatot is össze lehetne állítani arról, hogy a számítógépek, az információs technológia milyen hatást gyakoroltak a kultúra és a tudomány különböző területeire. Ez az előadás csak a matematika szempontjából veszi szemügyre a számítógépeket.

Legjobb, ha mindjárt az elején felhívom a figyelmet arra, hogy a matematika az egzakt gondolkodásról szól, és nem lehet a matematikáról anélkül beszélni, hogy legalább egy kicsit ne kóstoljunk bele ebbe az egzakt gondolkodásba. Elkerülhetetlen tehát, hogy a hallgatókat, olvasókat legalábbis időnként arra kérjem, hogy együtt gondolkozzunk, hogy ne csak a tényeket, hanem azok logikáját is kövessék.

Mindjárt fel is teszek egy kérdést, mely gyakran felvetődik, és melyre az első, átgondolatlan válaszunk könnyen az igazság ellenkezője lehet: *Feleslegessé teszik-e a számítógépek a matematikát?*

Minden nagy fejlődés sok olyan dolgot tesz feleslegessé, mely korábban fontos, sőt alapvető volt. Odakerül-e a matematika is a logarléc és a gőzmozdony mellé? Látni fogjuk, hogy éppen ellenkezőleg, a számítógépek igénye a matematika új fejezeteit hívja létre, és a régi fejezeteket egészen új megvilágításba helyezi. A számítógép új kérdéseket tesz fel a matematikának, és ezek a kérdések új fogalmak, új paradigmák kialakulásához vezetnek.

A probléma gyakran konkrétabb formában is megfogalmazódik: *Nem teszi-e feleslegessé a hardver fejlődése a matematikai módszereket a programozásban?* A hardver hihetetlen fejlődésen ment és megy keresztül. Ezt a fejlődést **Moore törvény**ével lehet leírni. Gordon Moore – az Intel egyik alapítója – 1965-ben azt a megfigyelést tette, hogy az egy-egy integrált áramkörben használt tranzisztorok száma mintegy kétévenként megduplázódik. Ez a gyors fejlődés azóta is tart.

Sok olyan feladat van, amelyet csak nehezen, bonyolult trükköket bevetve tudunk megoldani, de alig telik el egy év, és a sokkal gyorsabb, nagyobb kapacitású gépek játszva megoldják őket. Úgy tűnhet tehát, hogy kár a bonyolultabb matematikai módszereket erőltetni. Ám ha jobban megnézzük, kiderül, hogy éppen ellenkezőleg: a technikai fejlődés olyan feladatokat hoz létre, amelyeket „trükkös” gondolkodással már nem lehet megoldani, csak komoly matematikai módszerekkel.

Meg lehet-e érteni például egzakt matematikai gondolkodás nélkül az internetet, ami több százmillió számítógépet köt össze egymással? Meg lehet-e tervezni egzakt matematikai módszerek nélkül egy modern integrált áramkört, ahol ötvenmillió számítási alapegység van egy négyzetcentiméteren összezsúfolva? Lehet-e pontos matematikai modellezés nélkül gondolkodni egy olyan rendszer biztonságáról, amelyet milliárdnyi ember használ, akiket nem ismerünk – de tudjuk, hogy vannak köztük bűnözők és terroristák is?

A matematika tehát nagyon is sokat tud nyújtani a számítástechnikának. De mit kap cserébe? Nagyon sok mindent: izgalmas új fogalmakat, problémákat és módszereket, új kísérletezési lehetőségeket. Ezek közül nézünk meg néhányat a következőkben.

A bonyolultság új fogalma

A *bonyolultság* fogalma hasonló fejlődésen ment át az utóbbi ötven évben, mint sok más alapvető fogalom. Először egy megértést akadályozó körülményt jelentett (vagy talán csak kényelmes kifogást?). Ha egy jelenség vagy struktúra túl bonyolult, akkor a kutatás során megkerüljük, leegyszerűsítjük vagy részeire bontjuk. A fogalom történetében az a következő fázis, amikor a tudós elkezdi a bonyolultságot mint önálló jelenséget szemlélni: kidolgozza, hogyan lehet mérni, szabályokat és törvényeket állapít meg, ezeket kapcsolatba hozza más, korábban megismert dolgokkal. Végül jelentkezik a mérnök, hogy a bonyolultságot eszközként használja fontos tervezési problémák megoldására: az adatvédelem, az elektronikus posta, a kereskedelem, a pénzforgalom biztonsága ma nagyrészt a bonyolultság fogalmán és annak tulajdonságain alapul.

A 19. századi matematika egyik nagy sikere a *végtelen* fogalmának megragadása volt. Ennek bűvölete olyan erős, hogy hajlamosak vagyunk mindenre, ami véges, rálegyinteni: „Véges sok eset van, amit végig lehet nézni!” A számítógépek megjelenése következtében rájöttünk: a véges nagyon nagy lehet, bekövetkezhet az „exponenciális robbanás”.

Moore-törvény:
Gordon Moore 1965-ben tett megfigyelése, mely szerint egy-egy integrált áramkörben használt tranzisztorok száma mintegy kétévenként megduplázódik.

Exponenciális robbanás:
azon jelenség, hogy ha egy mennyiséget ismételten egy 1-nél nagyobb számmal szorzunk, akkor az igen gyorsan (robbanásszerűen) növekszik.

Bináris fa:
olyan fa (gráf), melynek minden ága (éle) tovább ágazik kétfelé.

Algoritmus:
egy elvégzendő cselekvéssorozat megtervezése lépésről lépésre.

Exponenciális robbanás

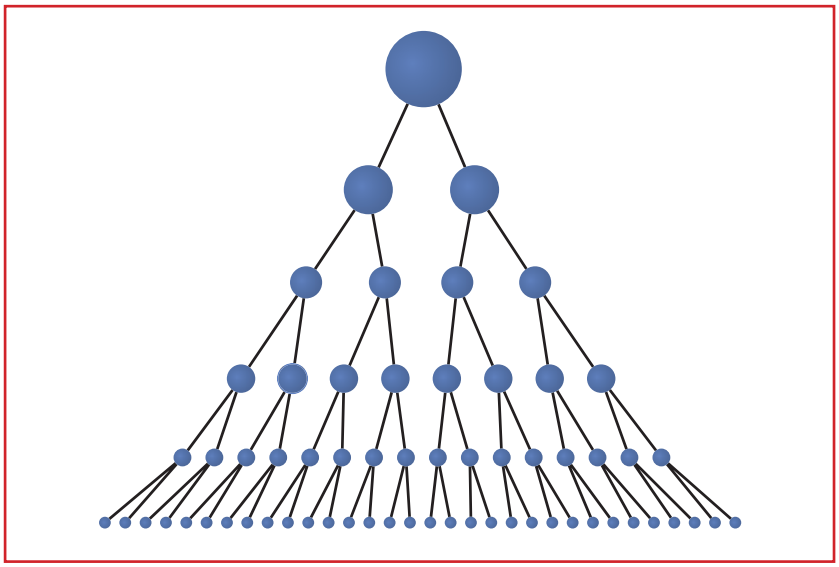
A sakkjáték feltalálójáról szóló klasszikus történet szerint az unalma elűzéséért hálás király felajánlotta neki, hogy azt kívánhat jutalmul, amit akar. A feltaláló azt kérte, hogy a sakktábla első mezejére tegyenek egy búzaszemet, a másodikra kettőt, a harmadikra négyet, a negyedikre nyolcat és így tovább, minden mezőre kétszer annyit, mint az előzőre. A király nagyon megörült, hogy ilyen olcsón megúszta, de aztán rá kellett jönnie: nemcsak hogy a 10–12-ik mezőtől már nem férnek el a búzaszemek, de a tábla közepe táján már birodalmának teljes búzatermése sem lett volna elegendő, hogy ígéretét betartsa.

Ezt a jelenséget, hogy ha egy mennyiséget ismételten duplázunk (vagy bármilyen egynél nagyobb számmal szorzunk), akkor igen gyorsan növekszik, **exponenciális robbanás**nak hívjuk. Moore említett törvényében az a meglepő, hogy az informatikában bekövetkezett exponenciális robbanás ilyen hosszan tart. Az 1960-as évek környezetvédelmi forradalma azért jött létre, mert egyre többen értették meg: az exponenciális növekedés nem tarthat örökké sem a népesség, sem a termelés, sem a fogyasztás területén.

A számítástudományban az exponenciális robbanás leggyakrabban akkor lép fel, ha azt a kijelentést, hogy „Véges sok eset van, amit végig lehet nézni!”, megpróbáljuk közelebbről is megvizsgálni. Mondjuk, az esetek megkülönböztetéséhez először is két alapesetet kell megkülönböztetni; aztán ezek mindegyikén belül két újabb eset van stb. Ezt a logikai helyzetet egy hálózattal, az ún. **bináris fá**val ábrázolhatjuk (1. ábra). Látható, hogy a végignézendő esetek száma már néhány elágazás után is ugrásszerűen nő, és bekövetkezik az exponenciális robbanás. Ha **algoritmus**unknak egy ilyen

Sakkozók. Miniatúra a Manesse-féle kéziratból, 14. sz.

1. ábra. *Bináris fa*



Polinomiális algoritmus: olyan algoritmus, melynek lépésszáma a feladat méretével úgy nő, mint a méret egy hatványa.

Prímszám: olyan egész szám, melynek csak az 1 és önmaga az osztója.

Prímfaktorizáció: a nemprím egész számok (összetett számok) felbontása prímszámok szorzatára. Például $90 = 2\times3\times3\times5$.

fát kell végigvizsgálnia, akkor a modern számítógépek – mondjuk – harminc szint mélységig még bírják, csakhogy minden egyes újabb szint megduplázza az esetek számát. Itt még a Moore-törvény sem segít: ha tíz évig várunk, akkor – mondjuk – öt szinttel tudunk továbbmenni.

Ezért aztán a számítástudományban elég hamar megfogalmazódott, hogy olyan algoritmusok tervezésére kell törekedni, melyek lépésszáma a feladat méretével csak mérsékelten nő úgy, mint a méret egy hatványa, nem pedig exponenciálisan. Tehát ha a feladat mérete n, akkor a lépésszám lehet n^2 vagy n^3, de nem lehet 2^n. Az ilyen algoritmust **polinomiális**nak nevezik. A kétfajta növekedés közötti különbséget úgy lehet a legjobban szemléltetni, ha elgondoljuk, mekkora teljesítménynövekedést lehet várni a technológia fejlődésétől. Tegyük fel, hogy mind az A algoritmus, mind a B algoritmus ma 50 bemenő adattal tud egy adott problémát megoldani. Az A algoritmus lépésszáma úgy nő a bemenő adatok számával, mint n^3, a B algoritmusé, mint 2^n. Ha várunk tíz évet, és a számítógép teljesítménye harminckétszer nagyobb lesz, mint ma (a Moore-törvény szerint), akkor az A algoritmus már 150 bemenő adatot tud kezelni, míg a B algoritmus teljesítőképessége csak 55 adatra emelkedik.

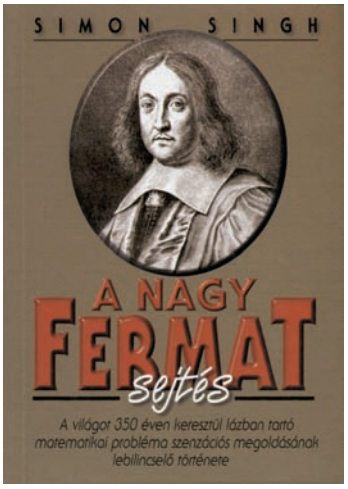
Prímtesztelés és -faktorizáció

Egy pozitív egész számot **prímszám**nak nevezünk, ha 1-en és önmagán kívül más egész számmal nem osztható. Az 1-et nem tekintjük prímnek, de utána aztán sok példát látunk: 2, 3, 5, 7, 11, 13 stb. Azokat a természetes számokat, melyek nem prímek, fel lehet bontani prímszámok szorzatára, például $6 = 2\times3$, $40 = 2\times2\times2\times5$ stb. – ezt a felbontást nevezzük **prímfaktorizáció**nak.

A *prímszámokkal* kapcsolatban két fontos algoritmikus problémát fogalmazhatunk meg: ha adott egy k jegyű szám, hogyan tudjuk eldönteni róla, hogy prímszám-e; és ha nem prímszám, hogyan tudjuk megtalálni a prímtényezőit?

Mindkét kérdésre könnyű „véges” választ adni: csak ki kell próbálni, hogy a szám osztható-e a következő számokkal: 2, 3, 4, 5 stb. Ha találunk olyan számot, amelyik osztója, akkor tudjuk, hogy nem prímszám, és a felbontást is könnyű elvégezni. A baj ezzel csak az, hogy iszonyú sok számot kell kipróbálni: ahhoz, hogy egy 100 jegyű számról eldöntsük, prím-e: 10^{100} számot kell kipróbálni, ami nagyobb, mint a világegyetem bármilyen paramétere. Kis odafigyeléssel észrevehetjük, hogy elegendő a szám négyzetgyökéig kipróbálni a számokat. De ez sem segít eleget: egy 100 jegyű szám négyzetgyöke 50 jegyű, és 10^{50} is messze túl van a lehetőségek határán.

Kiderül, hogy az első kérdés sokkal könnyebb, mint a második: olyan hatékony (polinomiális) algoritmusok vannak, amelyek akár 1000 jegyű számról is könnyedén eldöntik, hogy prím-e. A tényezőkre bontásra azonban csak olyan algoritmust ismerünk, amely lényegében exponenciális: 100-nál több jegy esetén már csak nagy üggyel-bajjal működik, 150 jegy fölött pedig egyáltalában nem. Látni fogjuk, hogy ennek az egyszerű ténynek hallatlan gyakorlati következményei vannak.



Hogyan lehetséges, hogy a prímfaktorizáció ennyivel nehezebb, mint a prímtesztelés? Azt gondolnánk, hogy mindegyikhez végig kell próbálni az adott számnál kisebb számokat, meg kell győződni: nem osztható-e velük az adott szám. Nyilvánvaló, hogy ez nagyon hosszadalmas lenne, ezért valahogy másképp kell eljárnunk. A hatékony prímtesztelő algoritmusok az ún. „kis” Fermat-tételen alapulnak (a „kis” jelző nem a tétel fontosságára utal, azért használjuk, hogy megkülönböztessük a „nagy” Fermat-tételtől. Ennek bizonyítása sokat váratott magára, amíg Andrew Wiles brit matematikus végre megoldotta a problémát. A bizonyítást 1995-ben tette közzé az *Annals of Mathematics* hasábjain. Pierre de Fermat, a nagy 17. századi francia matematikus csak a „kis tétel” bizonyítását adta meg.)

Fermat tétele szerint, ha p prímszám, és a tetszőleges egész szám, akkor

$$p \mid a^p - a$$

(p osztója az a^p-a számnak). Például, $2^5-2=30$ osztható 5-tel. Ha egy p számról el akarjuk dönteni, hogy prímszám-e, akkor megnézzük, hogy 2^p-2, 3^p-3 stb. oszthatók-e p-vel. Ellentétben a fenti egyszerű teszttel, ezt nem kell nagyon sok számra kipróbálni, elegendő csak néhányra.

(Sok problémát söpörtem itt a szőnyeg alá: néhány p számra ez a módszer hamis pozitív eredményt ad, ezért kicsit módosítani kell; nagy számok nagyon nagy hatványaival kell számolni, ami megoldható, de nem nyilvánvaló, hogyan stb. De mindezeket megoldva az a teszt, melyet Miller–Rabin-tesztnek neveznek, nagyon jól működik.)

A véletlen új fogalma(i)

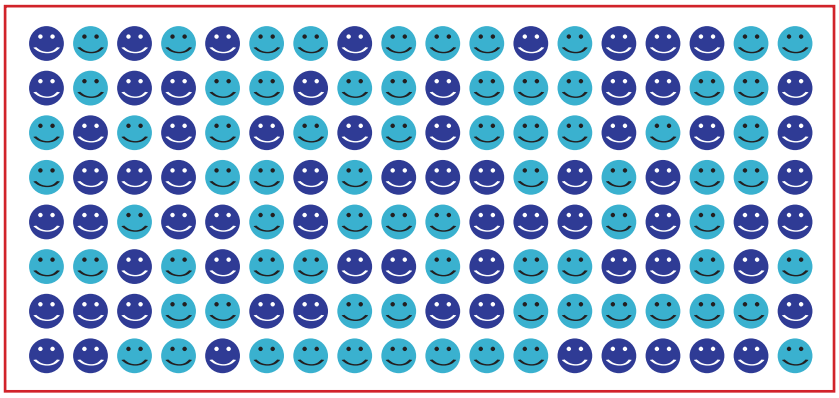
A *véletlen* a modern tudomány egyik sarkalatos fogalma. Szinte minden tudomány lépten-nyomon használ olyan modelleket, melyekben a jelenségek véletlen, statisztikus jellege dominál.

Ugyanakkor a véletlen igen nehéz fogalom is. Véletlen-e az, hogy egy földdobott pénz fejre vagy írásra esik? Ha jól meggondoljuk, miért is ne tudná egy igazán éles szemű és gyors eszű ember az alatt, amíg a pénz fölfelé száll, megfigyelni a pályáját, perdületét és amit csak még kell, és ebből kiszámítani, hogy melyik oldalára fog esni? A valóságban (talán a kvantumfizikát leszámítva) csupa olyan „véletlen" jelenséggel találkozunk, melyek igazából nem véletlenek, csak prognosztizálásukhoz nem áll rendelkezésre elegendő adat és idő.

Mitől nem véletlen egy sorozat?

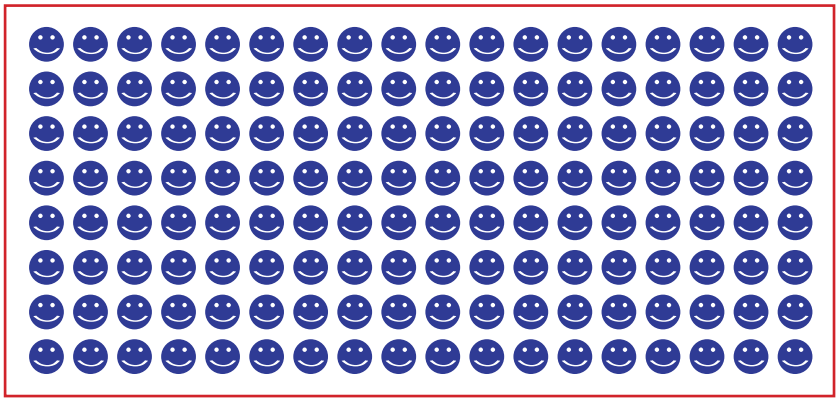
Nézzük a véletlen legegyszerűbb matematikai modelljét. Dobjunk fel egy pénzdarabot – mondjuk – százszor, és írjuk le, hogy fejet vagy írást kapunk. Valami olyasmit fogunk látni, amit a 2. ábra mutat.

2. ábra. *Az érmefeldobások eredménye*

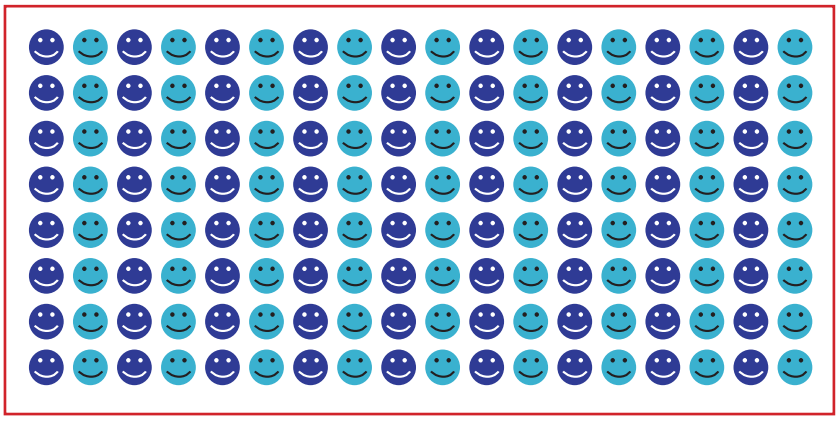


Tényleg véletlen pénzdobálással kaptam ezt a sorozatot? (Nem árulom el.) Ezt a kérdést nem is könnyű eldönteni. Persze ha valami könnyebbet kérdeznének, például a 3. ábrán látható sorozatot, akkor mindenki azonnal látná, hogy ez nem lehet véletlen. Tudjuk, hogy a pénzfeldobásnál ugyanannyi a fej, mint az írás valószínűsége, ezért egy hosszú pénzfeldobás-sorozatban körülbelül ugyanannyi fej kell legyen, mint írás.

3. ábra. *Hol az érem másik oldala?*



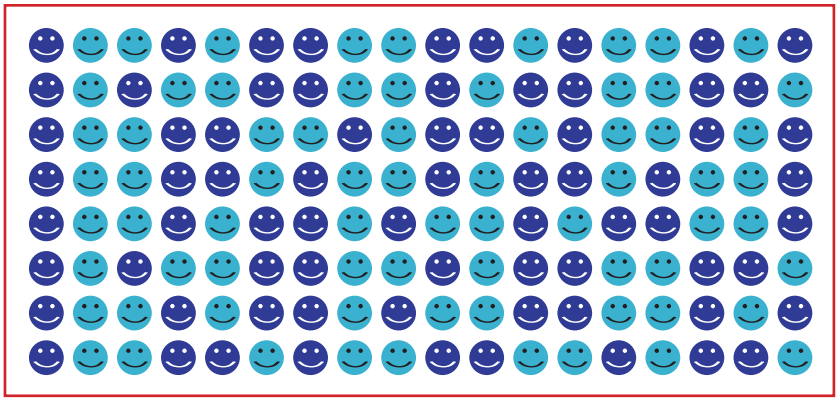
Nézzünk akkor egy újabb sorozatot! (4. ábra.) Ebben ugyanannyi fej van, mint írás, mégis nyilvánvaló, hogy ez se véletlen. (Érvelhetünk azzal, hogy a páros helyeken is fele-fele arányban kellene fejet és írást látni.)



4. ábra. *Ha figyelembe vesszük, hogy az érmének két oldala van…*

Nézzük a következőt! (5. ábra.)

Ez már sokkal kevésbé szabályos, sokkal véletlenszerűbb, de azért gyanús. Aki jártas a valószínűség-számításban, rögtön látja, hogy túl gyakran váltakozik, nincsen benne például három egyforma érme egymás után. De a legmeggyőzőbb, ha elmondom: a sorozat k-ik eleme  , ha a k–1 szám kettes számrendszerbeli alakjában az 1-esek száma páratlan, és  , ha páros. Noha a sorozat meglehetősen szabálytalan, igen egyszerű szabállyal leírható, tehát egyáltalában nem véletlenszerű.



5. ábra. *Vajon véletlen sorozat ez? I.*

Nézzünk most egy számsorozatot is! (6. ábra.) Véletlen ez? Gondolom, többen észrevették, hogy ezek egyszerűen a π („pi”) jegyei.

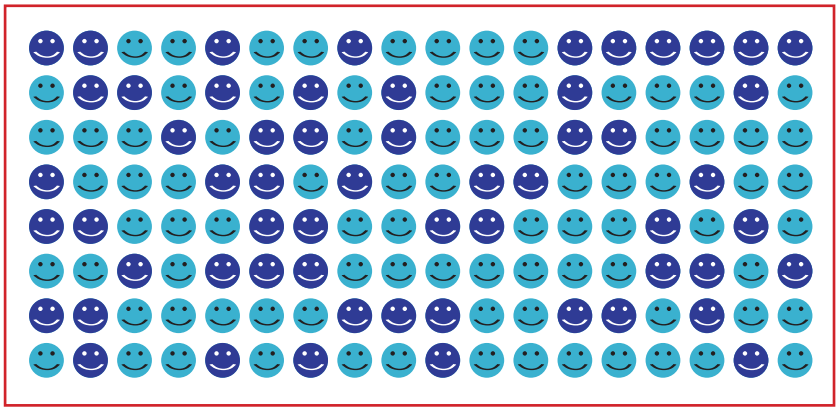
3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3
8 4 6 2 6 4 3 3 8 3 2 7 9 5 0 2 8 8
4 1 9 7 1 6 9 3 9 9 3 7 5 1 0 5 8 2
0 9 7 4 9 4 4 5 9 2 3 0 7 8 1 6 4 0
6 2 8 6 2 0 8 9 9 8 6 2 8 0 3 4 8 2
5 3 4 2 1 1 7 0 6 7 9 8 2 1 4 8 0 8
6 5 1 3 2 8 2 3 0 6 6 4 7 0 9 3 8 4
4 6 0 9 5 5 0 5 8 2 2 3 1 7 2 5 3 6

6. ábra. *Vajon véletlen sorozat ez? II.*

Még egy sorozatot nézzünk meg! (7. ábra.)

Ez már igazán véletlenszerűnek tűnik! Még a valószínűség-számításban jártas olvasó is nehezen találna kivetnivalót benne. Pedig ez is π, csak most 2-es számrendszerben van felírva.

7. ábra. *Vajon véletlen sorozat ez? III.*



Információtartalom és véletlenség

Információtartalom:
egy sorozat vagy egyéb struktúra információtartalma az, hogy milyen hosszú a lehető legrövidebb leírása a lehető leghatékonyabb kódolást használva.

Az elsőnek megadott sorozatot (2. ábra) azonban nem tudnám ilyen egyszerűen leírni, definiálni: igazában semmiféle más értelmes módon nem írható le, csak azzal, hogy felsoroljuk az elemeit. Pontosan ezzel definiálhatjuk a véletlen sorozatot: egy sorozat akkor véletlen, ha nem írható le rövidebben, mint a saját hossza.

Általánosabban megfogalmazva azt nevezzük egy sorozat vagy bármilyen más struktúra, kép stb. **információtartalmá**nak (vagy más néven *információs bonyolultságának*), hogy milyen hosszú a lehető legrövidebb leírása, mennyire lehet tömöríteni a sorozatot a lehető leghatékonyabb kódolást használva. (Ez matematikailag pontosan definiálható.)

Ezek után véletlen az a sorozat, melynek az információtartalma a hosszánál nem lényegesen kisebb. Megmutatható, hogy az ilyen módon definiált véletlen sorozatokra a valószínűség-számítás alapvető eredményei, például a nagy számok törvényei érvényesek lesznek.

Hilbert, David (1862–1943)

A 19–20. század fordulóján David Hilbert, a nagy német matematikus megfogalmazta az akkori matematika legfontosabb megoldatlan problémáit. Ezek egyike a *valószínűség* fogalmának megalapozása volt. Először Richard von Mises tett kísérletet arra, hogy ezt megoldja, olyasformán, ahogyan mi próbáltuk az előbb: egy fej–írás-sorozatot véletlennek nevezett, ha a sorozatban közel azonos a fejek és írások gyakorisága, és ez érvényes akkor is, ha csak minden második vagy minden harmadik elemet nézzük stb. Ez azonban nem vezetett sikerre.

Az 1930-as években A. Ny. Kolmogorov – egészen más úton elindulva – az analízis eszközeit (a mértékelméletet) felhasználva alapozta meg a valószínűség fogalmát. Ez matematikailag nagyon jól használható volt, azonnal elterjedt, és mindenki úgy tekintette, hogy ezzel a probléma meg van oldva. Kivéve magát Kolmogorovot, aki az 1960-as években visszanyúlt von Mises kísérletéhez, és azt úgy fejlesztette tovább, hogy matematikai-

lag használható lett. Ezzel nagy lépést tett a véletlen nehéz fogalmának megértése felé.

Egy sorozat információtartalmának a definíciója a valószínűség fogalmától függetlenül is sok izgalmas megoldatlan kérdéshez vezet: Mekkora a genetikus kód, például egy emberi kromoszóma információtartalma? Mekkora az agy bonyolultsága? Milyen nagyságrendű az egész világegyetem bonyolultsága?

Véletlen és álvéletlen

Van azonban a bonyolultság fogalmának egy hátránya is: egy sorozat bonyolultságát nem lehet semmilyen algoritmussal sem kiszámítani. Továbbá, ilyen értelemben véletlen számokat számítógéppel generálni fából vaskarika, hiszen egy számítógéppel generált sorozat információtartalma csak annyi, mint az őt generáló program információtartalma, ami a sorozat hosszától függetlenül korlátos.

Mivel azonban számítógép által generált véletlen számokra szükség van, egy másik fogalommal kell dolgozni. A bonyolultságelmélet segítségével két új kritériumot kaphatunk arra a kérdésre, hogy mikor tekinthetünk egy sorozatot véletlennek.

1. Képzeljük el, hogy két gépünk van, melyek mindegyike 0-kat és 1-eseket nyomtat egy papírra. Az egyik gépben egy igazi véletlent előállító fizikai eszköz van, és a kiadott sorozat egymástól független bitekből áll, melyek 1 valószínűséggel lehetnek 0 vagy 1. A másik gépben egy program gyártja a számokat egy rövid „magból", azaz egy olyan számból, amely a generáló algoritmus kiinduló (inicializáló) értéke. A programot ismerjük, a magról azonban csak annyit tudunk, hogy hány jegyű. A legfontosabb: nem tudjuk, melyik gép melyik. A feladatunk, hogy ezt megtippeljük. (Előfordulhat, hogy a valódi véletlent adó gép *véletlenül* olyan sorozatot állít elő, melyet az **álvéletlen**t adó gép is előállíthat. Ebben az esetben nem tudunk biztos választ adni. Ennek azonban igen kicsi a valószínűsége.)

Álvéletlen számok: algoritmussal előállított kellően bonyolult számsorozat, amely a véletlen számsorozat bizonyos jegyeit viseli magán.

Ha korlátlan idő állna rendelkezésre, akkor könnyű volna igen jó eséllyel tippelni: egyszerűen kipróbálnánk minden egyes magot, hogy abból indulva melyik generátor produkálja a megfigyelt sorozatot. Ehhez azonban 2^n sorozatot kell kipróbálnunk. Így ezt a túl könnyű megoldást kizárhatjuk azzal, hogy csak polinomiális algoritmust engedünk meg.

Egy másik triviális, érdektelen megoldás az volna, ha úgy tippelnénk meg, melyik sorozat az igazi véletlen, hogy feldobunk egy forintot. Az esetek felében ez jó eredményt ad. Ahhoz, hogy ezt kizárjuk, megköveteljük, hogy a felismerő algoritmus az esetek több mint felében adjon jó eredményt. Pontosabban azt követeljük meg, hogy annak a valószínűsége, hogy az algoritmus helyesen tippeli meg, hogy melyik sorozat a véletlen, legalább 51 százalék legyen.

Akkor mondjuk tehát, hogy a generátor *az igazitól megkülönböztethetetlen,* ha nincs olyan algoritmus, mely polinomiális időben az esetek 51 százalékában helyesen tippeli meg, hogy melyik gép az, amelyik a valódi véletlen sorozatot adja.

Véletlenszám-generátor: álvéletlen-számsorozatot előállító algoritmus.

2. Képzeljük el, hogy csak egy gépünk van, melyről tudjuk, hogy álvéletlen-sorozatot állít elő ismert programmal, de a magról ismét csak azt tudjuk, hogy milyen hosszú. Megfigyeljük a kijövő sorozatot, de mielőtt egy-egy újabb jegyét megnéznénk, megpróbáljuk megtippelni, hogy mi lesz a következő. Az előzőekhez hasonlóan a tippeléshez csak polinomiális időt használhatunk, és annyit kell elérnünk, hogy az esetek 51 százalékában sikeresen tippeljünk. Ha nincs ilyen algoritmus, akkor azt mondjuk, hogy a generátor *megjósolhatatlan.*

Ez a két definíció ekvivalens: *ha egy álvéletlen-sorozat az igazitól megkülönböztethetetlen, akkor megjósolhatatlan – és viszont.* Ez egyáltalában nem nyilvánvaló.

Még kevésbé nyilvánvaló az, hogy létezik elméletileg is jó generátor – ennek a kérdésnek a tárgyalása azonban nem fér bele ebbe az előadásba. Annyit mégis meg szeretnék jegyezni: ez is azon alapszik, hogy míg két számot könnyű összeszorozni, nem könnyű egy számot tényezőire bontani.

Ezeknek az eredményeknek filozófiai tartalma is van. Van-e értelme valamit „kiszámíthatónak", „determináltnak" nevezni, ha csak „elvileg" számítható ki, ami azt jelenti, hogy a számítás a világ végezetéig tartana?

A bizonyítás új fogalma

Püthagorasz görög matematikus, Raffaello freskójának részlete

A bizonyítás a matematika lelke, a görög tudomány talán legfontosabb alkotása. Mégis, az iskolai matematikatanításból sokszor kimarad. Még egyetemi hallgatók is gyakran nem értik, miért van rá szükség: „Elhisszük mi a tanár úrnak, minek azt bizonygatni."' (Remélem, ez csak amerikai tapasztalatom, Magyarországon nem így van.)

A Pitagorasz-tételre már nagyon-nagyon sok bizonyítást talált az emberiség, és világos, hogy a bizonyításon nem azért kell végigmenni, hogy a tétel helyességéről még jobban meg legyünk győzve, hanem azért, hogy a bizonyításban szereplő matematikai módszereket elsajátítsuk. A programok helyessége, a kommunikációs protokollok biztonsága azonban nap mint nap bizonyítást igényel(ne).

Interaktív bizonyítás

Még érdekesebb azonban, hogy az informatika a bizonyítás újszerű fogalmához is elvezet: *az interaktív bizonyításhoz.*

Alan M. Turing angol matematikus, a számítógép-tudomány egyik megalkotója azon gondolkodott, hogy vajon hogyan lehet megkülönböztetni egy nagyon fejlett számítógépet egy embertől. Azt a kísérletet javasolta, hogy egy szobába ültessünk be egy embert egy képernyővel és billentyűzettel, míg a másik szobába vagy egy másik embert ültessünk hasonló képernyővel és billentyűzettel, vagy egy számítógépet tegyünk. Az első ember kérdéseket tehet fel, vagy bármi egyéb módon társaloghat a másik szobában

levő lénnyel, és el kell döntenie ennek alapján, hogy emberrel vagy géppel áll-e szemben. Ezt a kísérletet nevezik **Turing-teszt**nek.

Hinnénk-e, hogy ezt a nyilvánvalóan csak gondolatkísérletnek szánt tesztet naponta sok ezerszer végzik el? Ha valaki például új elektromos postafiókot akar nyitni, ilyesféle kérdéssel találkozhat: „Kérjük, gépelje be az alábbi betűket”, és szürkés, kicsit piszkosnak tűnő alapon néhány kuszán odaírt betűt lát. (8. ábra.)

8. ábra. *Kérjük, gépelje be az alábbi karaktereket*

Ha valaki nem érti, mire való ez, rákattinthat a „Miért?” feliratra, és megtudja, hogy azért van erre szükség, mert sok program van, ami automatizálja a címek létrehozását, hogy aztán hirdetéseket küldjön szét róluk, vagy ami rosszabb, levelek tömeges küldésével megbénítson valakit. A képen az emberi szem könnyedén felismeri a betűket, de egy számítógépet (ma még legalábbis) a betűk torzulásai és a kusza vonalak megtévesztenek, így a programozott címgenerálást ki lehet szűrni.

Látszik tehát, hogy a mai számítógépek még nagyon gyorsan megbuknak a Turing-teszten, bár érdemes azt is megemlíteni, hogy itt a tesztet magát nem ember, hanem egy számítógép hajtja végre.

A fenti eljárás az interaktív bizonyítás szép példája: két résztvevő van, és ebben az esetben a Bizonyító azt a „tételt” akarja bebizonyítani, hogy ő ember. A hagyományos matematikában a tételhez hozzá lehet csatolni a bizonyítást – itt azonban van egy Ellenőr, aki egy vagy több kérdést tesz föl, és a „bizonyítás” a kérdéstől függ. Ez a séma sokkal erősebb a hagyományos bizonyítási formáknál – gondoljunk csak bele, hogyan tudnánk kérdés nélkül bebizonyítani, hogy emberek vagyunk.

Turing-teszt: gondolatkísérlet, mely azt hivatott eldönteni, hogyan lehet egy nagyon fejlett számítógépet megkülönböztetni egy embertől.

Érdemes azt is megjegyezni, hogy a séma ereje a kérdés (a kusza betűsorozat) véletlen voltán múlik. Ha mindenkitől ugyanazt kérdezné vagy akár csak előre kiszámítható kérdést tenne fel, könnyű volna egy számítógépet úgy programozni, hogy az a jó választ adja. A jó véletlen-generátor tehát ennek a protokollnak is elengedhetetlen része!

Elektronikus boríték

Hasonló interaktív bizonyítások lépten-nyomon előfordulnak – gondoljunk a jelszavakra, az „okos kártyákra” stb. Számítógépes hálózataink biztonsága nagyrészt az ilyen bizonyításokon múlik. Hadd mutassak be egy nagyon leegyszerűsített példát. Tegyük fel, hogy Aliz és Béla interneten keresztül sakkozik. Beesteledett, és meg kell szakítani a mérkőzést. Hagyományos sakkversenyen ilyenkor az történik, hogy – mondjuk – Aliz eldönti az utolsó lépést, de nem teszi meg a táblán, hanem borítékolja és odaadja a bírónak. A borítékot Béla csak másnap nyithatja ki. Így egyikük sem ismeri a

Euler, Leonhard svájci matematikus (1707–1783)

másik utolsó lépését, ezért nincs az a nagy előnye, hogy egész éjjel gondolkodhat a lépésén.

De most nincs bíró és nincs boríték. Aliz üzenhet valamit Bélának este, és megmondhatja a lépését másnap reggel. De mit üzenjen este? Ha már esti üzenetével elkötelezi magát, akkor Béla előnyben lesz; ha nem, akkor Aliz lesz előnyben, mert megváltoztathatja a lépést az éjszaka folyamán. A hagyományos információelmélet szerint a „borítékolás" lehetetlen.

A bonyolultságelmélet azonban megoldást kínál. Előre megállapodnak abban, hogy a lépéseket egy-egy négyjegyű számmal írják le: például Kf3 helyett azt mondják, hogy 9083. Ezek után Aliz választ magának két 100 jegyű prímszámot, amelyek közül a kisebbik úgy kezdődik, hogy 9083… (Láttuk, hogy számítógéppel nem nehéz ellenőrizni, hogy egy szám prímszám-e; be lehet bizonyítani a klasszikus matematika mély eszközeit felhasználva, hogy ha néhány száz 100 jegyű számot találomra kipróbál, ezek között nagy valószínűséggel lesz prím.) Legyen ez a két szám p és q, ahol $p<q$. Este Aliz elküldi a pq szorzatot Bélának, reggel pedig elküldi a két tényezőt (ami a boríték felbontásának felel meg).

Láthatjuk, hogy Béla már előző este megkapta a teljes információt: egy körülbelül 200 jegyű természetes számot, melynek a kisebbik prímtényezőjéből az első 4 jegy megadja Aliz lépését. De mivel a számot nem tudja hatékonyan tényezőire bontani, a lépést másnap reggelig (vagy akár évekig) nem tudja kiolvasni. Ezt a „titkot" hétpecsétként őrzi a számítási bonyolultság.

(Egyébként Béla jól teszi, ha ellenőrzi, hogy p és q tényleg prímek. Aki szeret logikai feladatokon eltöprengeni, elgondolkodhat, miért van erre szükség.)

Ennél sokkal bonyolultabb módon – de azért hasonló gondolatokat használva – lehet digitálisan létrehozni a társadalmi érintkezés olyan fontos elemeit, mint a mások által felbonthatatlan boríték (titkosírás), elismervény, számla, aláírás és annak hitelesítése, vízjel stb.

Gauss, K. F. német matematikus (1777–1855)

Klasszikus kérdések új megvilágításban

A klasszikus matematikát sokan elefántcsonttoronynak látják. Godfrey H. Hardy, aki a prímszámok elméletének egyik legkiemelkedőbb kutatója volt a 20. század első felében, ezt írja *Egy matematikus védekezése (A mathematician's apology)* című könyvében:

Soha nem tettem semmi „hasznosat". Semelyik felfedezésem sem volt közvetlenül vagy közvetve jó vagy rossz hatással a világ folyására, és nem valószínű, hogy valaha is hatással lesz…

Az „igazi" matematikusok „igazi" matematikája – Fermat és Euler és Gauss és Abel és Riemann matematikája – csaknem teljesen „haszontalan".

Amikor az interneten vásárolunk vagy bankügyeket intézünk, számítógépünknek több száz jegyű számokról kell eldöntenie, hogy prímek-e – tizedmásodpercek alatt. Ehhez Fermat tételét használja. A különböző számítógépes protokollok, biztonsági módszerek a Hardy által felsorolt nagyságok szinte mindegyikének a munkájára építenek.

Ha azt kérdezzük, melyik az a megoldatlan matematikai probléma, amelynek a legnagyobb a gyakorlati jelentősége, szerintem egyértelmű a válasz: *Fel lehet-e egy – mondjuk – 1000 jegyű számot hatékonyan (nem csillagászati idő alatt) prímtényezőire bontani?*

Azt hiszem azonban, hogy ezek a tények Hardy kutatási elveit legalább annyira alátámasztják, mint amennyire az állításait cáfolják. Ha ezeket a nagyságokat csak kutatásuk közvetlen haszna motiválta volna – és nem a matematikai kérdések szépsége, a megismerés vágya –, akkor ma nem lennének eszközeink a számítógép-rendszerek biztonságának védelmére.

Abel, N. H. norvég matematikus (1802–1829)

Ajánlott irodalom

Aho, Alfred V. – Hopcroft, John E. – Ullman, Jeffrey D.: Számítógép-algoritmusok tervezése és analízise. Bp.: Műszaki Kvk., 1982.

Gács Péter – Lovász László: Algoritmusok. Bp.: Műszaki Kvk., 1978.; Tankönyvkiadó, 1987.

Goldreich, Oded: Modern Cryptography, Probabilistic Proofs and Pseudorandomness. In: Algorithms and Combinatorics, Vol 17, Springer-Verlag, 1998.

Lovász László: Algoritmusok bonyolultsága. ELTE egyetemi jegyzet. Bp.: Tankönyvkiadó, 1989.

Lovász László: Egységes tudomány-e a matematika? *Természet Világa,* Matematika különszám, 1998.

Lovász László: Véletlen és álvéletlen. *Természet Világa,* Informatika különszám, 2000.

Lovász, László: Information and complexity (how to measure them?) In: The Emergence of Complexity in Mathematics, Physics, Chemistry and Biology (ed. B. Pullman), Pontifical Academy of Sciences, Vatican City. Princeton University Press, 1996. 65–80. p.

Luby, Michael: Pseudorandomness and Cryptographic Applications. Princeton, NJ: Princeton University Press, 1996.

Rónyai Lajos – Ivanyos Gábor – Szabó Réka: Algoritmusok. Bp.: Typotex, 1998.