Számítógépek, algoritmus, matematikai[1]

A fiataloknak ma több lehetőségük van arra, hogy a számítógépekkel megismerkedjenek, mint nekünk volt annak idején. Ezért elképzelhető, hogy sok olyan dolgot mondok el, mely számukra már nem ismeretlen, vagy amelyet nyilvánvalónak tartanak. Így belekapaszkodtam a cím „matematika" szavába, és arra szeretnék példákat mutatni, hogy a számítógépek fejlődése a matematikában milyen új kérdéseket vet fel, illetve hogy bizonyos nagyon is régi kérdéseket mennyire új megvilágításba helyez. Sok elintézettnek tekintett problémáról derül ki napjainkban, éppen a számítógépek kapcsán, hogy mennyire nem ismerjük a lényegét.

A tudomány és technika fejlődése, a társadalmi és gazdasági élet bonyolultabbá válása egy nagyon régi matematikai problémát hozott előtérbe : a véges és végtelen problémáját, ill. a folytonos és a „nem folytonos" vagy – matematikailag szólva – a diszkrét kérdését. A görögök ösztönösen idegenkedtek a végtelentől, s ez részben kerékkötője is volt a fejlődésnek, pl. a valós szám fogalmát nem tudták kellő egyszerűséggel bevezetni. Természetesnek érezték az egész és a racionális szám fogalmát, s ezt kezelni is tudták, de amikor kiderült az a tény, hogy a négyzet átlója, $\sqrt{2}$, nem írható fel két egész szám hányadosaként, ez már lényeges filozófiai nehézséget okozott. Mint tudjuk, ezt a lényegében pszichikai jellegű gátlást a 15–16. században átlépte a tudomány, bevezette a valós szám fogalmát, s kiderült, hogy ez nem is okoz problémát, azon a szinten jól lehetett vele számolni.

A valós szám fogalmát rögzítettnek tekintette a matematika a legújabb időkig. Amikor kiderült, hogy azt pl. nagyon könnyű felírni, hogy π, de ha most ezzel számolni kell, egy számítógépnek kell megadni, hogy tudjuk ezt megtenni? A szokásos zsebszámítógépen a π 8 vagy 12 jegyig be van írva a memóriába. A gép tehát π-n ezt a racionális számot érti. Elképzelhető persze olyan számítás, melynél nagyobb pontosságra van szükség.

Természetesen meg lehet adni a π-t másféleképpen is, be lehet a gépbe programozni valamit, ami a π jegyeit megadja. Végül is ez a π valós szám nem más, mint egy program, mely π-t tetszőleges pontossággal ki tudja számítani. Ez csak egy példa arra, hogy a valós szám fogalma nem is annyira nyilvánvaló, ha arra gondolunk, hogy ezekkel a számokkal mi nagy pontossággal számolni akarunk.

A matematika egész nyelvezetével kapcsolatban felvetődik hasonló probléma. Mindenki találkozott olyan megfogalmazással, hogy adott egy pont, adott egy szám, adott egy függvény stb. Amint ezeket a számítógépnek meg akarjuk fogalmazni, akkor át kell gondolni, hogy hogyan is vannak adva ezek a dolgok. Pl. egy függvény adva lehet képlettel, táblázattal, grafikonnal. Ezek nem mindegyikét tudja a számítógép értelmezni. Pl. a grafikont nem tudja. Előtérbe kerülnek tehát a matematikában azok a problémák, melyek a számítógépeknek megtanítható eljárásokon, algoritmuson alapulnak. Egy *algoritmus* olyan eljárás, amely formalizálva van, olyan precízen meg van határozva, hogy számítógépnek is megtanítható. Az algoritmus tehát olyan eljárás, amely számítógépre programozható. Ugyanígy előtérbe kerülnek az olyan definíciók, amelyek számításokkal követhetők. Így elválik egymástól a táblázattal vagy kiszámítási utasítással megadott függvény és az egyéb „nem konstruktív" módon megadott függvény fogalma.

Azért fontosak a véges módszerek, mert végtelen halmazokkal a számítógép nem sokat tud kezdeni, s mert a memóriája is véges. Mi persze úgy képzeljük, hogy egy függvény végtelen sok pontban van értelmezve, de amikor számítógéppel dolgozunk, akkor csak 10 tizedes jegyre van megadva a függvény értéke is, meg a helye is. Igazából tehát egy véges halmazon értelmezett és véges számú értéket felvevő függvénnyel dolgozunk.

Egy másik dolog, amit a számítógépek használata hangsúlyozott ki, a következő. Hajlamosak vagyunk arra, hogy ami véges, arra azt mondjuk, hogy nyilvánvaló, hiszen azt „végig lehet nézni". Egy geometriai feladatnál például, ha sikerült véges sok esetre visszavezetni a diszkussziót, akkor azt mondjuk, hogy meg van oldva, már csak véges sok esetet kell megvizsgálni. Ugyanígy általában az egyenletrendszert is megoldottnak tekintjük, ha megoldását visszavezettük véges sok eset vizsgálatára. A matematika, de főleg a számelmélet, tele van olyan tételekkel, amelyeknek az érvényessége valami iszonyú nagy számmal kezdődik (mint pl. $10^{10^{10}}$). „Véges sok szám kivételével" igaz, amit „végig lehet nézni". Mégis felmerülhet a kérdés, mennyi értelme van egy ilyen tételnek, hiszen soha nem vonatkozik olyan számokra, amikre alkalmazni szeretnénk. Kezd kialakulni ezért egy olyan szemlélet, amely különbséget tesz az ilyen reménytelenül nagy, véges sok lépést igénylő és a ténylegesen véges sok lépést felhasználó eljárások között. Sok vita van akörül, hogy mi az a lépésszám, ami még elfogadható. Teljesen kialakult konvenció nincs, de elég általánosan elfogadott, hogy az m^{const} lépésszám még „kicsi", de annál nagyobb már „nagy". Mit jelent ez? Az m jelenti a bemenő adatok hosszát, ha azokat 2-es számrendszerben írtuk fel. Azt vizsgáljuk tehát, hogy az eljárás a bemenő adatok függvényében legfeljebb milyen sokáig tart, és azt mondjuk, hogy *hatékony,* ha megadhatók olyan K_1 és K_2 pozitív valós számok úgy, hogy m hosszúságú bemenő adat esetén $K_1 m^{K_2}$ lépésben mindig véget ér.

Így most már matematikailag is pontos fogalmát kapjuk egy algoritmus hatékonyságának (persze, még az algoritmus fogalmát is előbb pontosan meg kellene alkotni, de erre ebben a cikkben nem vállalkozhatunk).

Ezt a fogalmat felhasználva, új logikai fogalmakhoz juthatunk el. Hogy kicsit világosabbá tegyük a problémát, a következő két mesét hallgassuk meg.

Párosítási probléma

Artúr király udvarában élt 150 lovag és 150 udvarhölgy, továbbá néha az udvarban tartózkodott Merlin, a gonosz varázsló. Artúr király elhatározta, hogy összeházasítja a 150 lovagot a 150 udvarhölggyel. Ez sehogyan sem sikerült,

[1] A szerzőnek az Ifjusági Matemetikai Kör téli ankétján elhangzott előadása alapján.

mert a lovagok és udvarhölgyek között többen nagyon nem szívlelték egymást, s nem akartak összeházasodni. Végül is Artúr király megunta a próbálkozást és ráparancsolt Merlinre, a varázslóra, hogy találjon egy házasítást az udvarhölgyek és a lovagok között.

Merlin – mivel természetfölötti képességekkel rendelkezett – rögtön látta, hogy ez nem lehetséges. (Persze, ezt nem olyan könnyű átlátni, hiszen az összes lehetséges párosítások száma 150!, ami egy rettenetesen nagy szám.) Ezt jelentette is Artúr királynak. De mivel Merlin nagyon sötét lelkű volt, a király nem tudhatta, hogy igazat mond-e vagy sem; ezért nem hitt neki, bezáratta egy toronyba, amíg csak ki nem talál egy megoldást a problémára. Mit csináljon most? – töprengett Merlin, mivel nem tudta bebizonyítani igazát Artúr királynak. Végül is felfedezett egy tételt, amelyet később *Frobenius* német és *Kőnig* magyar matematikusok is felfedeztek.

A tétel megfogalmazásához gondoljuk el a következő ábrát. A 150 udvarhölgy mindegyikét ábrázoljuk egy-egy ponttal, legyen ez az alsó sor, hasonlóan a lovagokat is, ez lesz a felső sor. Azokat, akik hajlandók összeházasodni, kössük össze egy-egy vonallal.

A vonalakat *él*eknek nevezzük. Az egész, pontokból és élekből álló alakzatot *gráf*nak.

Olyan éleket kellene kiválasztanunk, hogy minden pontból pontosan egy él induljon ki, és ha az összes pontot összekötöttük ezt *teljes párosításnak* nevezzük.

Frobenius–Kőnig tétele azt mondja ki, hogy: *egy gráfban akkor és csak akkor van teljes párosítás, ha bármely k alsó pontnak legalább k felső szomszédja van.*

Ez azt jelenti, ha lent kiválasztok pl. 50 udvarhölgyet, akkor ezek szomszédjainak száma, vagyis azoknak száma, amelyekkel valamelyikük össze van kötve, legalább 50 (lehet több is). Ez a feltétel nyilván szükséges ahhoz, hogy legyen párosítás.

Nehezebb annak belátása, hogy elégséges is, vagyis ha ez a feltétel teljesül, akkor van teljes párosítás.

Mikor erre Merlin rájött, homlokára csapott, és felhívatta az összes udvarhölgyet és lovagot, felállította őket két sorba. Majd kijelölt kb. mondjuk 50 lovagot és felszólította az udvarhölgyeket, hogy tartsa fel a kezét az, aki hajlandó ezen 50 lovag valamelyikéhez feleségül menni. S mivel csak 48 kéz emelkedett a levegőbe, ezzel már be is bizonyította a királynak, hogy a házasítás nem oldható meg.

Ültetési probléma

Artúr király udvarában a vacsorát közösen, egy nagy kerek asztalnál költötte el mind a 150 lovag. A problémát csak az jelentette, hogy a lovagok között voltak olyanok, akiket nem lehetett egymás mellé ültetni, mert egy idő után összeverekedtek. Bárhogyan próbálkozott is a király olyan ültetéssel, hogy ilyen párok ne kerüljenek egymás mellé, az sohasem sikerült. Egy idő után megunta a próbálkozást, és ismét magához hívatta Merlint, hogy készítsen ilyen ülésrendet. Tudjuk, hogy az összes ültetések száma 149! Merlin ismét azonnal látta, hogy nincs megfelelő ülésrend, de a király nem hitte el, és bezáratta a toronyba. De sajnos ezúttal Merlinnek nem jutott eszébe semmilyen egyszerű fogás, amellyel bizonyíthatta volna az igazát Artúr királynak. Ezért szegény azóta is ott ül.

*

Két hasonló feladattal találkoztunk : a párosítással és a körbeültetéssel. Ezek úgynevezett *kereső feladatok*: ha sikerül a problémára egy megoldást megtalálnunk, nincs tovább mit bizonyítani. Ez hasonló ahhoz, mint amikor valaki a kesztyűjét keresi. Ha megtalálta, akkor már nincs mit tennie.

A kereső feladat pontosan megfogalmazva azt jelenti, hogy adva van a 0, 1 számok egy sorozata (amely jelentheti pl. egy gráfnak a pontjait). Az eredmény, amit keresünk, ugyancsak egy, a 0 és 1 számokból álló sorozat (pl. a párosítás). Feltesszük még, hogy azt, hogy a második sorozat valóban megoldása az elsőnek, „könnyen” ellenőrizni tudjuk (ami azt jelenti, hogy az ellenőrzéshez szükséges lépések száma a bemenő adatok hosszának valamilyen hatványánál nem nagyobb).

A párosítási probléma esetén adott egy gráf, és ebben párosítást keresünk. Az ültetési probléma esetén ugyancsak egy gráffal írhatók le a bemenő adatok (a pontok a lovagokat ábrázolják, és két lovagot egy éllel kötünk össze, ha képesek verekedés nélkül végigenni a vacsorát egymás mellett). Vegyük észre, hogy amikor Merlin rájött, hogy a párosítási problémának nincs megoldása, és hogy a Frobenius–Kőnig tételt tudja alkalmazni, akkor egy másik keresési feladattal találta magát szemközt : keresnie kellett valamilyen k-ra k lovagot úgy, hogy k-nál kevesebb udvarhölgy legyen hajlandó ezek valamelyikéhez feleségül menni. Nevezzük a lovagok ilyen halmazát *majomszigetnek*.

A párosítási és az ültetési feladat sok szempontból hasonló, de van amiben különbözik: az első feladathoz (a párosításhoz) találtunk egy másik kereső feladatot (a majomsziget feladatot) úgy, hogy ha az elsőnek van megoldása, akkor a másiknak nincs, és fordítva. Vagyis az egyiknek akkor és csak akkor van megoldása, ha a másiknak nincs. Ezeket *komplementáris kereső feladatok*nak nevezzük. Az ültetési feladathoz azonban ilyet eddig még nem találtak, s nagyon valószínű, hogy nincs is. Nem csak azért, mert eddig senki sem talált, hanem az alábbiak miatt is.

A keresési feladatok nagyon gyakran egymásra visszavezethetők, pl. a teljes párosítás visszavezethető a körbeültetésre. (Ez nem nyilvánvaló, érdemes rajta gondolkozni!)

Vannak olyan kereső feladatok, amelyekre minden más kereső feladat visszavezethető. Ezt a meglepő tényt 12 évvel ezelőtt *Cook* amerikai matematikus bizonyította.

Később mások (*Karp* amerikai és *Levin* szovjet matematikus) azt is kimutatták, hogy a gyakorlatban fellépő kereső feladatok többsége ilyen. Így pl. minden más kereső feladat visszavezethető a körbeültetési feladatra. Ez azt jelenti, hogy minden kereső feladathoz meg lehet konstruálni egy gráfot, méghozzá hatékonyan úgy, hogy ebben a gráfban

akkor és csak akkor lehet egy minden pontot felfűző zárt poligont találni, ha az eredeti feladatnak van megoldása. S ennek a poligonnak az ismeretében az eredeti feladat megoldását is meg lehet határozni. Ez azt támasztja alá, hogy azokat a kereső feladatokat, melyekre minden más kereső feladat visszavezethető, nem lehet hatékonyan megoldani. Ez ugyanis azt jelentené, hogy ha ezek egyikét megoldanánk, akkor ezzel az összeset megoldanánk, s ez olyan széles, mind elméleti, mind gyakorlati szempontból fontos körét jelentené a feladatoknak, hogy ez már túl szép volna. Más jelenségek is vannak, melyek ezt alátámasztják. (Általánosan elfogadott hipotézis, hogy ezeket nem lehet megoldani.)

Ha valaki nem is tudja a 150! lehetőséget átlátni, de tudja, hogy van jó párosítás, kérdés, hogy hogyan tudja ezt megtalálni. Olyan algoritmust kellene találni, ami hatékony és minden gráfhoz vagy talál egy teljes párosítást – és ezt megadja, vagy a komplementer feladatot oldja meg, s ezzel bebizonyítja, hogy nincs megoldás. Ilyen algoritmus is létezik. Ezt *Kőnig Dénes* és *Egerváry Jenő* magyar matematikusok alkották meg, és magyar módszernek nevezik. Ez az algoritmus n^3 számú lépést igényel, ahol n a gráf pontjainak száma.

A pontos matematikai, fogalmak birtokában könnyen bizonyítható, hogy ha egy kereső feladat megoldható hatékony algoritmussal, akkor van komplementáris feladata. Nem ismeretes azonban, hogy ez az állítás megfordítható-e. A konkrét esetekben az a tapasztalat, hogy egy-egy kereső feladathoz könnyebb komplementáris feladatot találni, mint hatékony megoldási algoritmust; ez a komplementáris feladat azonban előbb-utóbb (néha 10-20 év múlva) a hatékony megoldó algoritmus kulcsa lesz.

A fenti fogalmak nemcsak a kombinatorikában alkalmazhatók sikerrel, hanem a matematika más ágaiban is, még a legidősebb, legrangosabb ágakban is, mint pl. a számelméletben. Egészen elemi kérdések, melyeket hosszú idő óta megoldottnak tekintettek, kerülnek új megvilágításba.

Tekintsük az m természetes számot, melyről azt akarjuk eldönteni, hogy prímszám-e vagy sem. Másképpen azt is mondhatjuk, hogy keressük meg ennek a számnak a prímtényezős felbontását, azaz írjuk fel

$$m = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k},$$

alakban, ahol p_1, p_2, ..., p_k különböző prímszámok, és α_1, α_2, ..., α_k természetes számok.

Euklidész bebizonyította, hogy minden egész számnak egyértelmű a prímfelbontása. De hogyan lehet ezt megtalálni? Ha valaki ezt a problémát hallja, azt úgy oldja meg, hogy veszi az egész számokat 2-től $\sqrt{m}$-ig és megnézi, hogy osztója-e valamelyik a számnak. Ha egyikkel sem volt osztható, akkor a szám prím. Ez elég sok számolást jelent, pl. egy 100 jegyű számnál 10^{50}-ig kell kipróbálni a számokat. Sokáig az volt a vélemény, hogy a számelmélet a legtisztább matematika, olyan tiszta, hogy már semmire sem használható. Mégis a prímtényezős felbontás kérdése egy teljesen gyakorlati probléma, egy titkosírás kapcsán merült fel, amely szerencsére nemcsak katonai célokra használható. (A Természet Világa 1981. júniusi számában *Babai László*tól jelent meg egy cikk erről az alkalmazásról.)

Ennek ötlete a következő. Rögzítsünk egy p prímet (mondjuk 100 jegyűt). Legyen x egy természetes szám, melyre $1 \leqq x \leqq p$. Ez az x az üzenet. Tekintsünk egy e természetes számot, amelyre $1 \leqq e \leqq p-1$ és $(e, p-1) = 1$, azaz e és $p-1$ relatív prímek. Ez az e szám az üzenet küldőjének *kulcsa*.

Tekintsük az x^e üzenet p-vel való osztási maradékát, jelöljük ezt $\overline{x^e}$-sal.

Valahogy ezt kiszámítjuk (hogy hogyan, arra később térünk ki), és küldjük el ezt a számot mint sifrírozott üzenetet. Akkor ebből a címzett az x-et vissza tudja kapni. A címzett kulcsa ugyanis egy olyan d szám lesz, amelyre $p-1 \mid de-1$.

Legyen $de = (p-1)k + 1$, valamilyen k-val, akkor $\overline{(\overline{x^e})^d} = x$.

Ez miért igaz? Nem nehéz belátni, hogy egy osztási maradék r-edik hatványa ugyanaz, mint az egész szám r-edik hatványának osztási maradéka. Ezért

$$\overline{(x^e)^d} = \overline{x^{(p-1)k+1}} = \overline{x\big(x^{(p-1)k} - 1\big) + x}.$$

Bármely x számra $p \mid \big(x^{p-1} - 1\big)$ (ez a „kis Fermat-tétel", ezt felhasználva $p \mid x^{(p-1)k} - 1$. Ezt tehát, mivel osztható p-vel, el is hagyhatjuk, így marad $\overline{x}$. Ezzel tehát valóban meg lehet fejteni az üzenetet. Ahhoz azonban, hogy az eljárás működjön, azt kell tudni, hogy hogyan számíthatjuk ki x^e-t.

Ezt nem lehet úgy kiszámolni, hogy x-et e-szer összeszorzom! Ha p kb. 100 jegyű, akkor x is kb. 100 jegyű, az e maga is kb. 100 jegyű, akkor ez azt jelentené egyrészt, hogy nagyon sokszor kell szorozni, másrészt hogy iszonyúan hosszú számot kapnánk. A számok hosszúságán könnyű segíteni. Mivel a végén úgyis csak a p-vel való osztási maradékot tekintjük, azt megtehetjük, hogy minden egyes szorzás után osztunk p-vel, és csak az osztási maradékkal dolgozunk tovább. A szorzások számát a következő ötlettel lehet csökkenteni. Ha a 32-edik hatványt kell venni, akkor először négyzetre emelek, majd újra négyzetre emelek és így tovább 5-ször. (Közben minden négyzetreemelés után veszem a p-vel való osztási maradékot, és azzal számolok tovább.) Ugyanazt az eljárást használva megtehetem, hogy az e-t 2-es alapú számrendszerben írom fel:

$$e = 2^{a_1} + 2^{a_2} + \dots + 2^{a_r}, \quad \text{ahol} \quad a_1 > a_2 > \dots > a_r,$$

ekkor legfeljebb 100 tagot kapok, s mindegyik tag legfeljebb 2^{100}. Másrészt

$$x^{2^{a_1}} \cdot x^{2^{a_2}} \dots = x^e,$$

a tényezőket könnyen ki tudom számolni, az elsőt a_1-szeres, a másodikat a_2-szörös négyzetre emeléssel, s azokat összeszorzom. Ez összesen legfeljebb 100×100 művelet, melyet egy számítógép kényelmesen el tud végezni.

Ahhoz, hogy az eljárás működjön, szükségünk van még arra is, hogy p prímszám legyen. Ezért vetődött fel az a kérdés, hogyan lehet eldönteni egy p számról, hogy prím-e. Ezt máig sem tudjuk elég hatékonyan eldönteni, de vannak olyan algoritmusok, amelyek gyakorlati szempontból hatékonyak.

Ez a titkosírás jó, mert valaki csak akkor tudja megfejteni, ha ismeri hozzá a kulcsot, a d-t. Egy másik kulcs is van hozzá, ami a küldőé, ez az e. Nem lehet-e kihasználni azt, hogy két kulcs van? Például úgy, hogy ne lehessen megfejteni csak a küldő kulcs ismeretében a titkosírást?

Vannak ennek a titkosírásnak olyan változatai, hogy nem lehet az e birtokában az üzenetet megfejteni, a d birtokában pedig nem lehet „hamisítani". Ezek a módosítások azon múlnak, hogy az m szám prímtényezős felbontására jelenleg nem ismerünk hatékony módszert. (Pontosabban: nincs nyilvánosságra hozva ilyen módszer.)

Bár eddig a titkosírásoknak elsősorban katonai alkalmazásai voltak, újabban hasonló kérdések a polgári életben is felvetődnek. Az utóbbi időben foglalkoznak az elektromos posta kérdésével, hogyan lehetne a levelezést papír nélkül, „dróton" továbbítani. Ez egy alapvető problémát vet fel: az aláírást és pecsétet nem lehet elektromos úton hitelesíteni, s ez visszaélésekre adhat alkalmat. Ezt úgy próbálják kiküszöbölni, hogy ravasz kódolási eljárásokat használnak. Abból, hogy egy üzenet egy adott kulccsal megfejthető, már következik, hogy csak az küldhette, akinek a neve alá van írva.

Ennek a módszernek az alapgondolatát szemléltetjük egy sokkal egyszerűbb eljáráson keresztül. Ennek kulcsa, mondjuk, egy százjegyű szám. Ha ezt beadom a számítógépbe, megnyílik a széf. A probléma az, hogy a számítógép személyzete hozzáférhet ehhez az információhoz, s ez visszaélésekre ad lehetőséget. Hogyan lehet ezt megakadályozni?

Ahelyett, hogy a számítógépbe beprogramoznám a kulcsot, a következőt csinálom: felrajzolok 150 pontot és felfűzöm valamilyen sorrendben egy zárt poligonra, azután hozzáveszek még éleket, így egy gráfot kapok. Ezután megjegyzem az eredeti poligont, a gráfot pedig berakom a számítógép memóriájába. A gép csak erre fog emlékezni, a poligonra nem. A széf kinyitását a következő program szabályozza: akkor nyisd ki az ajtót, ha valaki ennek a gráfnak minden pontját felfűző poligonját adja meg.

Ha a gép kezelője ki akarná nyitni a széfet, akkor a gép memóriájából könnyen ki tudja olvasni a gráfot. Ahhoz azonban, hogy a széf kinyíljon, még egy minden pontot felfűző poligont (ún. Hamilton-kört) is kell találnia, és – mint ezt Merlin már szerencsétlenségére tapasztalta – ez emberi idő alatt nem oldható meg. Hiába áll minden információ rendelkezésére, a számítási bonyolultság jobban védi a széfet, mint akármilyen hosszú, bonyolult jelszó vagy kulcs.

Ajánlott irodalom: *Gács-Lovász:* Algoritmusok (Műszaki Kiadó, Bp., 1978.).
Babai László: Prímszámok és titkosírás (Természet Világa 1981. 6. szám).