Ez a beszámoló arról szól, hogyan oldottunk meg a *Számítástechnikai Koordinációs Intézetben* egy egész számokkal kapcsolatos feladatot. Az eredményt a gépkocsikra, hajókra és vonatokra felszerelt ultrarövid hullámú rádió adó-vevők frekvenciáinak megválasztásában hasznosítják. Hogy elmondhassuk, milyen leckét adtak fel számítóközpontunknak a Magyar Posta mérnökei, egy kis matematikai és műszaki bevezetőre van szükség.

**A feladat**

*Tekintsük a számegyenes egész értékű pontjait. Jelöljünk ki köztük $n$ darabot $(a_1,\ a_2,\ \ldots,\ a_n)$ úgy, hogy az általuk alkotott szakaszok* $\left(\text{összesen}\dfrac{n(n-1)}{2}\text{darab}\right)$ *különböző hosszúságúak legyenek.*

A számelméletben 40 éve foglalkoznak azzal a kérdéssel, hogyan adható meg olyan módszer, amellyel elég „sűrű", ilyen tulajdonságú véges sorozatok készíthetők, tehát ahol az $a_n - a_1$ különbség lehetőleg kicsi. A matematikusok, akik főleg a prímszámok tulajdonságaira támaszkodtak konstrukcióikban (lásd később az 1. feladatot) valószínűleg nem is sejtették, hogy az ilyen sorozatoknak nagy szerepük van a rádiózásban.

Legyenek ugyanis valamilyen $c$ valós számmal a $c \cdot a_1,\ c \cdot a_2,\ \ldots,\ c \cdot a_n$ értékek egyazon helyről, pl. adótoronyból egyidejűleg sugárzott URH-adások frekvenciái. Ha az $a_i$ számoknak megvan a mondott tulajdonságuk, az adás hallgatói megmenekülnek egy kellemetlen zavarástól, az ún. „harmadrendű intermodulációtól". Tehát a jó vétel szükséges feltétele, hogy

(a) $$a_j - a_i \neq a_i - a_k$$

teljesüljön az olyan $i$, $j$, $k$, $l$ számnégyesekre, amelyekre $1 \leq i < j \leq n$; $1 \leq k < l \leq n$, kivéve ha $i = k$ és $j = l$ egyszerre teljesül.

Ha ezek valamelyik négy frekvenciára nem állnának fenn, akkor a négy adás mindegyikét zavarná a másik három. [A műszakiak akkor is harmadrendű intermodulációról beszélnek, ha az (a) képletben pl. $j = k$, azaz a zavaró frekvenciák száma csak kettő.]

Számítógéppel ilyen frekvenciasorokat készíteni nem nehéz. A mi feladatunk azonban az volt, hogy egyszerre 7 adó frekvenciáit adjuk meg harmadrendű intermodulációtól mentesen, minden adóhoz 8 frekvenciát, méghozzá úgy, hogy az így kapott 56 számot nagyság szerint sorba rendezve a szomszédosok közötti különbség ugyanaz a $c$ állandó legyen. (Az $55 \cdot c$ szélességű frekvenciatartományban így fér el a lehető legtöbb frekvencia.)

A számegyeneshez visszatérve, a probléma így fogalmazható át: 1-től 56-ig 7 színnel kell kiszíneznünk az egész értékű pontokat. Minden színnel 8 pontot, mégpedig úgy, hogy az ezek által alkotott 28 távolság különbözzék egymástól. Mindehhez jön még két követelmény: a fenti távolságoknak 1-nél határozottan nagyobbnak és 41-nél határozottan kisebbnek kell lenniük. Tehát ha $a_i$ és $a_j$ egyszínűek, akkor

(b) $$1 < |a_i - a_j|;$$

(c) $$|a_i - a_j| < 41.$$

Ezek a korlátozások az adók műszaki jellemzőiből fakadtak. A (b) feltétel pl. azért kellett, mert szomszédos frekvenciák között esetleg „áthallás" léphet fel.

**Kezdeti próbálkozások. Elméleti úton nem megy**

Nyomozni kezdtünk a szakirodalomban, hátha tisztán matematikai módszerekkel célhoz érünk. Matematikus találkozókon, hasonló témájú szemináriumokon feladtuk a problémát. Kiderült, hogy nagyon sok hasonló struktúrát lehet algebrai módszerekkel létrehozni, de teljes megoldás sem a könyvekből, sem matematikus társainktól nem jött. Az ilyen remények meghiúsulásával helyzetünk egyre jobban hasonlított a hagyományos mérnöki kiindulóponthoz: a feladatot a rendelkezésre álló eszközökkel kell megoldani, és ha ez nem megy, akkor a megrendelővel folytatott „alku" során addig kell enyhíteni a feltételeken, amíg azok teljesíthetőkké nem válnak. Munkánk konkrét feltételei közé tartozott az idő is, kétféle értelemben. Egyrészt a frekvenciakiosztást határidőre meg kellett csinálni, másrészt ügyelnünk kellett arra, hogy ha számítógépes programot írunk – és ez elkerülhetetlennek látszott – , akkor a futási idővel arányosan fogy a rendelkezésre álló pénz.

**A számítógép keres, de semmit sem talál: túl szigorúak a feltételek**

A továbbiakban végig táblázatként lesz szó a kívánt frekvenciakiosztásról, mert számítógépünk az ugyanahhoz az adóhoz tartozó („egyszínű") számokat egy sorba írta, és a sorokat egymás alá helyezve kereste a megoldást. A keresést itt majdnem szó szerint kell érteni, mert az eljárás, amit végül is választottunk, olyan volt, mint amikor valaki egy szénakazalban akar megtalálni egy tűt és ehhez finom műszere, pl. mágneses fémkeresője van.

Először is elkészítettük az összes lehetséges „jó" sort, vagyis azokat a 8 elemű sorokat, amelyekre teljesülnek az (a), (b), (c) feltételek, és elemeik 1 és 56 közé esnek. Összesen 904 ilyen sor van. Néhány példa:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 4 | 9 | 15 | 19 | 31 | 38 | 40 |
| 7 | 11 | 17 | 30 | 33 | 38 | 45 | 47 |

Ezután közös elem nélküli jó sorokból elkezdtünk felépíteni egy táblázatot. Amikor már nem volt készletünkben olyan sor, amit hozzátehettünk volna résztáblázatunkhoz, akkor ebből elhagytuk az utolsó sort, hátha helyette kettőt

vagy többet illeszthetünk oda. Néhány tízezerszer megismételtettük a géppel ezt a műveletet, mely a táblázatot hol növelte, hol fogyasztotta. A kiírt résztáblázatok nagyságának megoszlása kiábrándító volt: legtöbbször már két sor után sem lehetett folytatni a növelést, és örültünk, hogy sok drága gépidő felhasználása után a kevés háromsoros mellé szerencsésen kijött egy négysoros. Azt sejtettük – de ezt bizonyítani ma sem tudjuk –, hogy ez a statisztikusan megnyilvánuló sikertelenség szükségszerű: nincs olyan táblázat, amelynek sorai az (a), (b), (c) feltételeket kielégítik. A baj gyökerét abban láttuk, hogy a (c) feltételben levő korlát túl kicsi volt, így a számegyenesen az „egyszínű" pontoktól akkora sűrűséget követeltünk, amit már nem lehetett teljesíteni.

Amikor ezt megrendelőinknek megmondtuk, kiderült, hogy a legnagyobb és legkisebb sorelem különbségét felülről korlátozó számot a műszaki irodalom túl óvatos adataiból vették át. Éppen ekkor folytak azok a mérések, amikből pontosan meg akarták tudni, legfeljebb mennyi lehet az adóknak ez a műszaki jellemzője, amit átkapcsolási sávszélességnek hívnak. Szerencsénk volt. A méréssorozat a 41-es felső korlátot 50-re engedte emelni. Tehát az egy sorban levő $a_i$ és $a_j$-re:

(d) $$|a_i - a_j| < 50.$$

### A keresésnek is van matematikája

Mielőtt rátérnénk arra, mit kezdtünk ezzel a váratlan ajándékkal, térjünk ki néhány szóval a keresési stratégiákra. Sok könyv és cikk foglalkozik a leggyorsabb, legolcsóbb keresés módszereivel. Más-más eljárásokkal kell felkutatni játékokra (pl. sakk) beprogramozott számítógépek segítségével a jó lépést; a különböző szempontok szerint tárolt adatokat; tételbizonyító programokban a keresett levezetést. Megkönnyíti az áttekintést, ha elkülönítjük azokat a feladatokat, ahol egy alaphalmaz valamilyen szempont szerinti *legjobb* elemét keressük, azoktól a problémáktól, ahol az alaphalmaz jó elemei közül *bármelyikre* „vadászunk". A mi esetünk az utóbbi csoportba tartozik, tehát – újabb hasonlattal élve nem egy tűkből álló halmaz leghegyesebb tűjét, hanem *csak egy* eléggé hegyes tűt keresünk. Míg az előbbi esetben erőfeszítésünk a tűhalmaz méreteivel lenne arányos, addig az utóbbi feladatnál az alaphalmaz mérete nem sorsdöntő.

Jó, hogy így van ez, mert ha alaphalmaznak pl. az első 56 természetes szám összes lehetséges felbontását tekintenénk 7 darab 8 elemű részhalmazra, akkor e halmaz $\frac{56!}{(8!)^7} \approx 10^{42}$ elemének előállítása még a leggyorsabb számító gépeken is legalább $10^{25}$ évig tartana. (A komputerek számolási sebességének elvi felső korlátai vannak, pl. a fénysebesség túlszárnyalhatatlansága miatt.)

A keresés helyes módszere a második esetben: viszonylag olcsón felkutatni a tűhalmaz olyan részeit, ahol a kielégítően hegyes tűk nagyobb valószínűséggel fordulnak elő. Ezeken az „ígéretes" részhalmazokon belül aztán finomabb eljárással kellett folytatni a keresést.

### Módszeres bolyongás a résztáblázatok erdejében

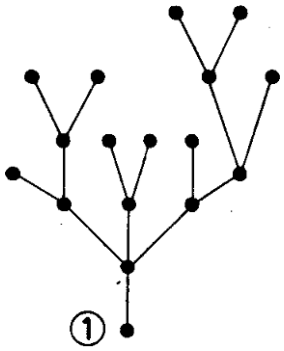
Térjünk vissza a $7 \times 8$-as táblázathoz. Létrehoztunk néhány ezer, az (a), (b) és (d) feltételnek eleget tevő sort. Ezekkel is lefuttattuk a különböző elemekből álló sorokat egymás alá válogató és a keletkező résztáblázatot hullámokban növelő-csökkentő programot. Most már nem volt ritka a négysoros résztáblázat, és nagysokára összejött egy hatsoros is. Ezzel azonban a módszer ereje kimerült: hatalmas szerencse vagy sok számítógépidő (azaz több pénz) kellett volna a teljes táblázat ilyen módon való megkereséséhez. Ezért a következő fokozatos finomításhoz folyamodtunk.

A jó 6 sorból álló résztáblázat elemeit egyenként kicseréltük a maradék számokból álló „rossz" sor egy-egy elemével az összes lehetséges módon. Ha mindkét sor megjavult volna a csere hatására, akkor a teljes táblázatot megkaptuk volna. De a keresésnek ezen a még „sokat markoló" szintjén nem volt ekkora szerencsénk. Ha az egyik sorból „egy híján jó sort" sikerült csinálni, a másikból pedig jó sort, akkor az egy szám híján teljes táblázatot átadtuk egy részprogramnak, a „finoman keresőnek", amiről később lesz szó. [Egy híján jónak neveztünk egy 8 elemű sort, ha volt olyan 7 elemű részhalmaza, ami eleget tett az (a), (b), (d) feltételeknek.] Minden esetben az elemeket visszatettük eredeti helyükre, és vettük a következő kicserélendő elempárt. A menet közben keletkező új hatsorosakat kimentettük, hogy később ezekből kiindulva számolhassunk. (Nehéz volt megszervezni, hogy ugyanaz a táblázat a cserélgetés folyamán még egyszer vissza ne térjen, azaz hogy elkerüljük a végtelen ciklust.)

Közben figyeltük a keresés hatékonyságát, amit a percenként megtalált „egy szám híján teljes táblázatok" számával mértünk. Azt tapasztaltuk, hogy a hatékonyság többszörösére nőtt, ha a jó és a rossz sor között nem egy-egy, hanem két-két elem cserélt helyet. A továbbiakban ezzel az átalakítással állítottuk elő a „finoman kereső" részprogram bemenő adatát, az egyre gyakrabban felbukkanó „egy híján teljeseket".

A „finoman kereső" feladata szintén a rossz és jó sorok közti egyenkénti elemcsere volt. Ezt úgy szerveztük, hogy ha közben kialakult egy új „egy híján teljes táblázat", akkor a részprogram ezt is végigfésülte, és csak akkor lépett vissza „anyatáblázatához", ha eközben nem bukkant újabb „egy híján teljesre". A szervezésben az ismert keresési elvet, a „fa először mélységben való bejárását" alkalmaztuk. Olyan ez, mint amikor egy lombozatától megfosztott fa ábráját a következő módon rajzoljuk le:

A fa bejárását a gyökérnél, 1-nél kezdjük, azaz ceruzánk az 1 ponttól indul. Menet közben nem emelkedhet fel a papírról, és az elágazási pontok kivételével minden vonalon legfeljebb kétszer futhat végig. Az ágak találkozási pontjainak és végeinek az „egy híján teljes táblázatok" felelnek meg, maguk az ágak pedig az egyik ilyen táblázatot a másikba alakító elemcserék szimbólumai. Az 1 pont a kiinduló táblázat megfelelője.




**Célhoz érünk. Néhány szó a program futási sebességéről**

Egy izgalmas éjszakai programfuttatás során végre ránk mosolygott egy teljes táblázat, és pár perc múlva még egy. A „finoman kereső" találta meg őket. Íme az egyik:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 14 | 20 | 34 | 36 | 41 | 44 |
| 2 | 9 | 13 | 18 | 21 | 31 | 46 | 48 |
| 3 | 8 | 19 | 25 | 33 | 40 | 43 | 52 |
| 4 | 12 | 16 | 22 | 37 | 42 | 51 | 53 |
| 6 | 10 | 27 | 30 | 32 | 39 | 45 | 55 |
| 7 | 11 | 24 | 26 | 29 | 35 | 49 | 56 |
| 15 | 17 | 23 | 28 | 38 | 47 | 50 | 54 |

Az állandóan módosított program végső formájában kb. 500 FORTRAN nyelven írt utasításból állott. Futási sebességét úgy növeltük, hogy a legtöbbször hívott programrészeket – pl. a sorok jóságát ellenőrző VIZSGA szubrutint – a géphez közelebb álló ASSEMBLER nyelven írtuk. (A VIZSGA-t olyan módszerekkel is gyorsítottuk, amelyek a 2. feladatban szerepelnek.)

A táblázat keresése több hónapon át tartó izgalmas játék, szinte vadászat volt. Azoknak a kedves olvasóinknak, akik egészen idáig követték beszámolónkat, sok hasonlóan érdekes feladatot kívánunk!

A következőkben a problémához kapcsolódva közlünk néhány feladatot.

1. *Erdős Pál* és *Turán Pál* 1941-ben a következőképpen adtak meg az (a) követelményeket kielégítő, egész számokból álló sorozatot.

*Ha $n = p-1$ és $p$ prímszám, legyen $a_i = 2 \cdot p \cdot i + r_i$, ahol $r_i$ az a maradék, amit $i^2$-nek $p$-vel való osztásakor kapunk $(i = 1,\ 2,\ \ldots,\ p-1)$. Bizonyítsuk be, hogy a kapott sorozat rendelkezik a kívánt tulajdonsággal!*

2. *Edwards, Durkin* és *Green* a következőképpen konstruáltak olyan egész számokból álló $a_1 < a_2 < \ldots < a_n$ sorozatot, amire az (a) követelmény teljesül, és amire $D_n = a_n - a_1$ a lehető legkisebb. Előzetes eredményekből (pl. az 1. feladat tanulságából) tudták, hogy $D_n < 2n^2$. Ezért

(1) *Legyen* $D_n = 2n^2 - 1$.

(2) *A* $D_n$ *számot felbontották* $n-1$ *különböző pozitív egész összeadandóra:*
$D_n = d_1 + d_2 + \ldots + d_{n-1}$.

(3) *Legyen*
$a_1 = 0$ *és*
$a_i = d_1 + d_2 + \ldots + d_{i-1} (i = 2,\ 3,\ \ldots,\ n)$.

(4) *Az* $a_1 < a_2 < \ldots < a_n$ *sorozatra megvizsgálták, teljesül-e az* (a) *kikötés.*

(5) *Ha igen, akkor* $D_n$*-et eggyel csökkentették, és a* (2) *ponttól folytatták.*

(6) *Ha* (4)*-ben* (a) *nem teljesült, akkor* $D_n$*-nek megkeresték egy új, eddig még nem vizsgált felbontását* $n-1$ *különböző pozitív egész szám összegére. (Egy* $D_n = d_1 + d_2 + \ldots + d_{n-1}$ *felbontás akkor is újnak számít, ha az előzőktől csak elemeinek sorrendjében különbözik.) Ezekkel a* $d_i$*-kkel folytatták a számítást a* (3) *ponttól. Ha nincs több új felbontás, akkor* $D_n$ *a minimálisnál eggyel kisebb, és vége.*

*Az eljárás* (4) *pontjában meg kell vizsgálni, teljesülnek-e az* (a) *feltételek. Bizonyítsuk be, hogy ez bármely* $a_1 < a_2 < \ldots < a_n$ *sorozat esetében elvégezhető úgy, hogy legfeljebb* $\binom{n+1}{4}$ *feltétel teljesülését kelljen ellenőrizni! Mutassuk ki, hogy ez a felső korlát akkor is érvényes, ha eljárásunk során csak azt vizsgáljuk, hogy az egyes* (a)*-beli nem-egyenlőségek két oldalán levő mennyiségek egyenlők-e, de azt nem használjuk ki, melyikük a nagyobb.*

*Igazoljuk; hogy az* (a) *kikötés teljesülésének vizsgálatához már* $\frac{n(n-1)}{2} \cdot \log \frac{n(n-1)}{2}$ *összehasonlítás is elegendő. (Itt már nem kerülhető el a nagyság szerinti sorbarendezés.)*