Stabil házasságok
avagy
hogyan működik a felvételi pontszámhatárok meghatározása

Az érettségi vizsgák eredménye jelentős mértékben meghatározza az egyetemi felvételiken figyelembe vett pontszámokat. Ugyanez igaz a középiskolai felvételi vizsgákra is. De vajon hogyan határozza meg a jelentkező pontszáma és az egyetemek, illetve középiskolák általa megadott sorrendje azt, hogy végül hova veszik fel? Erről szól ez a cikk, pontosabban arról az algoritmusról, aminek segítségével az egyetemeket, illetve középiskolákat és a felvételizőket egymáshoz rendelik.

A különböző országok egyetemei különböző felvételi eljárásokat alkalmaznak, sokszor ez még egy országon belül sem egységes. Ezeket az eljárásokat nehéz összemérni, mert sok szempont szerint hasonlíthatók össze, egyikben az egyik jobb, másikban a másik. Ráadásul egy adott tulajdonság esetén is szubjektív lehet annak megítélése, hogy melyik a jobb. Ennek ellenére megkockáztatom azt a kijelentést, hogy Magyarországon az egyetemi és a középiskolai felvételi eljárás is nagyon igazságos, és közel van az optimálishoz.

Ebben a cikkben csak a felvételi eljárás legvégére koncentrálunk, amikor már minden diáknak megvannak a pontszámai azokra a szakokra, amelyekre jelentkezett, illetve minden diák rangsorolta a szakokat, amelyekre jelentkezett, vagyis megadta, hogy melyik szakra menne a legszívesebben, melyikre a második legszívesebben stb.

Ezek után mindössze a következő elvárást fogalmazzuk meg:

Ne forduljon elő az, hogy a D_1 diákot az E_1 szakra vették fel, a D_2-t pedig az E_2-re, miközben D_1 szívesebben ment volna az E_2-re, mint E_1-re és az E_2 szak is előbbre rangsorolta D_1-et, mint D_2-t.

A *„D_1 szívesebben ment volna az E_2-re”* azt jelenti, hogy D_1 rangsorában az E_2 szak előrébb van, mint az E_1, az *„E_2 szak előbbre rangsorolta D_1-et, mint D_2-t”* pedig azt, hogy a D_1 diák E_2 szakra vonatkozó pontszáma magasabb, mint a D_2 diáké.

Ez a probléma bonyolult ahhoz, hogy itt részletesen tárgyaljuk, ezért ennek egy erősen egyszerűsített változatának megoldására mutatunk egy egyszerű, de hatékony algoritmust. Ezt az egyszerűsített helyzetet a matematikában **stabil házasság problémának** hívjuk, melynek szokásos megfogalmazása a következő:

Adott k nő ($N = \{n_1, n_2, \ldots, n_k\}$) és k férfi ($F = \{f_1, f_2, \ldots, f_k\}$). Mindenkiről tudjuk, hogy a másik nem tagjait hogyan rangsorolja abból a szempontból, hogy mennyire szívesen kötné össze vele az életét. Minden nő rangsorolja a férfiakat, vagyis minden n_i megad egy teljes $>_{n_i}$ rendezést a férfiakon, és minden férfi rangsorolja a nőket, vagyis minden f_i megad egy teljes $>_{f_i}$ rendezést a nőkön, holtverseny most nem megengedett. Ez felfogható úgy is, hogy minden nő esetében adott F egy permutációja, és minden férfi esetében pedig N egy permutációja. (Az eredeti problémához képest itt a nők az egyetemi szakok és a férfiak a diákok, vagy fordítva. Nyilvánvaló, hogy ez erős egyszerűsítése az eredeti problémának, hiszen ebben az esetben ugyanannyi szak van, mint ahány diák.)

Ezt a kiindulási helyzetet két $k \times k$-s táblázattal kiválóan tudjuk szemléltetni. (Vagyis a bemeneti adatok mérete $2k^2$).

Hogy jobban megértsük az algoritmust, ezért egy konkrét példán végig is nézzük a lépéseit. 5 nő és 5 férfi szerepel a mi történetünkben, vagyis két 5×5-ös táblázat írja le az algoritmus bemeneti adatait:

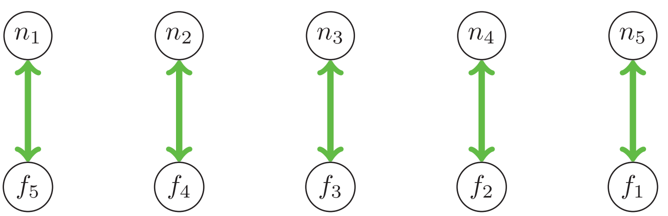
n_1	n_2	n_3	n_4	n_5
f_4	f_5	f_4	f_3	f_2
f_3	f_4	f_1	f_2	f_5
f_1	f_1	f_2	f_1	f_4
f_2	f_2	f_5	f_5	f_1
f_5	f_3	f_3	f_4	f_3

f_1	f_2	f_3	f_4	f_5
n_1	n_1	n_5	n_4	n_5
n_2	n_4	n_1	n_2	n_2
n_5	n_5	n_3	n_3	n_3
n_4	n_2	n_2	n_5	n_1
n_3	n_3	n_4	n_1	n_4

Ez tehát azt jelenti, hogy n_1 legszívesebben f_4-hez menne feleségül, és legkevésbé f_5-höz, míg f_2-nek n_1 tetszik a legjobban és n_3 a legkevésbé.

Vizsgáljuk meg, hogy egy párbaállítást mikor tekinthetünk stabilnak, vagyis mit jelent az az egyetlen elvárásunk, amit megfogalmaztunk a párosítással kapcsolatban. Azt jelenti, hogy nincs olyan nő és férfi, akik kölcsönösen jobban vonzódnak egymáshoz, mint a párosításbeli párjukhoz. Vagyis nincs olyan n_i és f_j, akiknek a párosításbeli párjaik f_r és n_s, de $f_j >_{n_i} f_r$ és $n_i >_{f_j} n_s$. Ők ugyanis szívesebben elszöknének együtt, minthogy elfogadják az aktuális párjaikat.

Nézzük meg, hogy stabil-e az a párosítás, amikor a párban lévő tagok indexének összege 6:

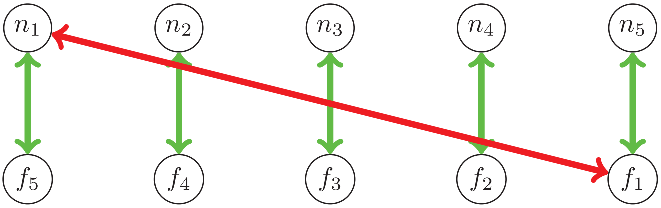


Ez nem stabil, hiszen n_1 és f_1 szívesen elszöknének. Miért? Mert f_1 rangsorában n_1 az első, így megelőzi n_5-öt, és n_1 rangsorában is előrébb van f_1 (a 3. helyen), mint f_5 (aki az utolsó).

(A következő ábrán zöld nyíllal jelöltük a párosítást, pirossal pedig a párt, akik szívesen elszöknének.)

Érdemes megnézni, hogy ha a párokat az azonos indexű nőkből és férfiakból képezzük, akkor az a párosítás stabil lesz-e.

Egy párosítást tehát akkor mondunk stabilnak, ha nincs olyan férfi és nő, akik nem egymás párjai, de együtt szívesen elszöknének.



A feladat a következő: találni egy $h : N \to F$ kölcsönösen egyértelmű megfeleltetést (bijekciót), amelyre igaz az, hogy nincs olyan n_i és n_s, hogy

$$h(n_s) >_{n_i} h(n_i) \quad \text{és} \quad n_i >_{h(n_s)} n_s.$$

(Az eredeti jelölések és a h kölcsönösen egyértelmű megfeleltetés közötti kapcsolat alapján: $f_r = h(n_i)$ és $f_j = h(n_s)$.)

A probléma az 1950-es évek elején már felvetődött és azt a National Resident Matching Program meg is oldotta, orvosrezidenseket és kórházakat párosítva megfelelően.

1962-ben *David Gale* és *Lloyd Shapley* belátta, hogy mindig létezik stabil párosítás, amelyet könnyen meg is lehet találni az azóta **Gale–Shapley algoritmusnak** nevezett eljárással. Később *Alvin E. Roth* Nobel-díjas közgazdász észrevette, hogy ez az algortimus pontosan ugyanaz, amit a National Resident Matching Program 10 évvel Gale és Shapley előtt már alkalmazott.

Az algoritmus a következő:

1. kör

1. lépés: Minden férfi megkéri a listáján első helyen szereplő nő kezét.

2. lépés: Minden nő „feltételesen" elfogadja a számára legkedvezőbb ajánlatot (eljegyzések).

Elképzelhető, hogy egy nő nem kap ajánlatot. Ekkor vár tovább.

Ha egy férfit elutasít egy nő, mert volt kedvezőbb ajánlata, akkor azt a nőt a férfi végleg kihúzza a listájáról.

...

***k*. kör**

1. lépés: Minden nem eljegyzett férfi megkéri a listáján legelöl szereplő nő kezét, akit még nem kért meg.

2. lépés: Minden nő megnézi, hogy az új ajánlatok, illetve az esetleg már meglévő eljegyzés közül melyik a legkedvezőbb a számára, és azt választja. Ha ehhez fel kell bontania a meglévő eljegyzését, akkor felbontja azt.

Addig lesznek újabb és újabb körök, amíg mindenki tagja nem lesz egy párnak. Az is elképzelhető, hogy már az első körben véget ér az algoritmus.

Nézzük meg az algoritmust a mi példánkon:

Az első körben f_1 és f_2 megkéri n_1 kezét, f_3 és f_5 pedig n_5 kezét, míg f_4 ajánlatot tesz n_4-nek.

Jelöljük az ajánlattétel tényét **A**-val egy táblázatban. Ezt követően tegyük meg a második lépést, és jöjjenek létre az eljegyzések, amiket **E**-vel jelölünk. n_1 két ajánlatot is kap, f_1-től és f_2-től. A listáján f_1 előrébb szerepel, mint f_2, így f_1 ajánlatát átmenetileg elfogadja, f_2-t viszont elutasítja. Hasonló történik n_5-tel is, aki f_5 ajánlatát fogadja el és f_3-at utasítja vissza.

	n_1	n_2	n_3	n_4	n_5
f_1	**A**				
f_2	**A**				
f_3					**A**
f_4				**A**	
f_5					**A**

$\longrightarrow$

	n_1	n_2	n_3	n_4	n_5
f_1	**E**				
f_2					
f_3					
f_4				**E**	
f_5					**E**

Mivel n_1 elutasította f_2-t, n_5 pedig f_3-at, ezért a férfiak ezeket a nőket kihúzzák a listájukról.

Mivel ennek a két férfinak nincs eljegyzése, így ők a listájukon legelöl álló nőnek tesznek ajánlatot. Vagyis f_2 n_4-nek, f_3 pedig n_1-nek.

Mivel n_4 jobban szimpatizál f_2-vel, mint f_4-gyel, ezért felbontja a meglévő eljegyzését. Hasonlóan jár el n_1 is.

	n_1	n_2	n_3	n_4	n_5
f_1	**E**				
f_2				**A**	
f_3	**A**				
f_4				**E**	
f_5					**E**

$\longrightarrow$

	n_1	n_2	n_3	n_4	n_5
f_1					
f_2				**E**	
f_3	**E**				
f_4					
f_5					**E**

Mivel n_1 elutasította f_1-et, n_4 pedig f_4-et, ezért a férfiak ezeket a nőket kihúzzák a listájukról, és a következő körben ők kérik meg az aktuális listájukon legelőkelőbb helyen álló nő kezét, vagyis mindketten n_2-ét. n_2 a saját listája alapján f_1-et elutasítja, f_4-gyel eljegyzi magát:

	n_1	n_2	n_3	n_4	n_5
f_1		**A**			
f_2				**E**	
f_3	**E**				
f_4		**A**			
f_5					**E**

$\longrightarrow$

	n_1	n_2	n_3	n_4	n_5
f_1					
f_2				**E**	
f_3	**E**				
f_4		**E**			
f_5					**E**

Mivel f_1-nek nincs eljegyzése, ezért ő ajánlatot tesz n_5-nek, mivel ő a legelöl álló hölgy a listáján. (n_1-et és n_2-t már kihúzta a listáról.)

Mivel n_5 listáján előbb szerepel f_5, mint f_1, így nem bontja fel a meglévő eljegyzését, hanem elutasítja f_1-et.

	n_1	n_2	n_3	n_4	n_5
f_1					**A**
f_2				**E**	
f_3	**E**				
f_4		**E**			
f_5					**E**

$\longrightarrow$

	n_1	n_2	n_3	n_4	n_5
f_1					
f_2				**E**	
f_3	**E**				
f_4		**E**			
f_5					**E**

Így megint csak f_1 tesz ajánlatot, mégpedig n_4-nek, de most sem jár sikerrel, mert n_4 előbbre rangsorolta f_2-t, így ő sem bontja fel az eljegyzését.

	n_1	n_2	n_3	n_4	n_5
f_1				**A**	
f_2				**E**	
f_3	**E**				
f_4		**E**			
f_5					**E**

$\longrightarrow$

	n_1	n_2	n_3	n_4	n_5
f_1					
f_2				**E**	
f_3	**E**				
f_4		**E**			
f_5					**E**

Az utolsó lépéshez érkeztünk, f_1 végső elkeseredésében ajánlatot tesz n_3-nak, akinek nincs eljegyzése és más ajánlata sem, így jobb híján elfogadja f_1 ajánlatát.

	n_1	n_2	n_3	n_4	n_5
f_1		**A**			
f_2				**E**	
f_3	**E**				
f_4		**E**			
f_5					**E**

$\longrightarrow$

	n_1	n_2	n_3	n_4	n_5
f_1			**E**		
f_2				**E**	
f_3	**E**				
f_4		**E**			
f_5					**E**

Így tehát azt kaptuk, hogy az n_1–f_3, n_2–f_4, n_3–f_1, n_4–f_2, n_5–f_5 egy stabil párosítás, senki nem akar elszökni senkivel.

Ez a „hagyományos” modell volt, hiszen a férfiak kérték meg a nők kezét. Ha a nők kérnék meg a férfiak kezét – ki lehet próbálni –, akkor is a fenti párosítást kapnánk.

Ez alapján felmerülhet az a sejtés, hogy mindig pontosan egy stabil párosítás létezik. Ez sejtés azonban gyorsan hamisnak bizonyul.

Vegyük a következő nagyon egyszerű példát: összesen két férfit és két nőt akarunk összeházasítani. A bemeneti adatokat a következő két egyszerű táblázat tartalmazza:

n_1	n_2
f_2	f_1
f_1	f_2

f_1	f_2
n_1	n_2
n_2	n_1

Mi történik, ha a férfiak tesznek ajánlatot? Akkor az első kör első lépésében mindkét nő egy-egy ajánlatot kap, így az algoritmus egy lépésben véget is ér, a kapott párosítás: f_1–n_1, f_2–n_2.

Mi történik, ha a nők tesznek ajánlatot a férfiaknak? Akkor az első kör első lépésében mindkét férfi egy-egy ajánlatot kap, így az algoritmus ekkor is egy lépésben véget ér, de a kapott párosítás: f_1–n_2, f_2–n_1.

A stabil párosítás nem mindig egyértelmű, de azt Gale és Shapley bebizonyította, hogy ilyen feltételek mellett mindig létezik megfelelő párosítás, és egy párosítást a fenti algoritmus meg is talál (méghozzá a bemeneti adatok számától lineárisan függő számú lépésben).

A fenti nagyon egyszerű példában az látszik, hogy ha a férfiak tesznek ajánlatot, akkor nekik kedvez az algoritmus (mindkét férfi a számára szimpatikusabb nővel került párba), ha pedig a nők teszik az ajánlatot, akkor nekik kedvez (mindkét nő a számára szimpatikusabb férfival került párba).

Ez általában is igaz. Mit jelent az, hogy a nőknek kedvez az algoritmus? Azt, hogy ha P_n az a stabil párosítás, amelyet úgy kaptunk, hogy a nők tettek ajánlatot, akkor nincs olyan P stabil párosítás, amelyben van olyan nő, akinek ott számára kedvezőbb párja van, mint P_n-ben. Vagyis ha vesszük az összes stabil párosítást, és megnézzük, hogy melyik nőnek kik a párjai valamelyik stabil párosításban, akkor P_n-ben ezek közül a lehetséges párok közül a saját rangsorában a legjobbat kapja meg párként.

Nézzük még meg azt, hogy miben különbözik az egyetemi felvételi rendszere a stabil párosítási problémától:

- Az egyetemi szakok és a felvételiző diákok száma nem egyezik meg.
- Az egyetemi szakokra nem egy diákot vesznek fel, hanem többet, és meg van határozva egy maximális létszám, ahány diákot fel lehet venni az adott szakra.
- A felvételizők szigorú rendezést adnak meg a számukra szimpatikus egyetemi szakok között, ahogy ez a stabil házasság problémában is szerepel. Azonban az egyetemek úgy rangsorolják a diákokat, hogy lehet közöttük holtverseny, hiszen lehet két diáknak azonos felvételi pontszáma egy adott szakra.

Tekintsünk el most az utolsó különbségtől. Ekkor az algoritmust tudjuk módosítani úgy, hogy továbbra is működjön.

Most tehát vannak az egyetemi szakok ($S = \{s_1, s_2, \ldots, s_k\}$), ezekhez adottak pozitív egészek ($M = \{m_1, m_2, \ldots, m_k\}$) úgy, hogy az s_i szakra legfeljebb m_i diákot lehet felvenni. Illetve adottak a diákok ($D = \{d_1, d_2, \ldots, d_n\}$). Tegyük fel továbbá, hogy minden diák azokat a szakokat rangsorolta, ahová szívesen menne (nincs tehát diákonként az összes szak rendezve), illetve minden szak szigorúan rangsorolja a diákokat, akik oda jelentkeztek.

Gondoljunk most minden s_i szakra úgy, hogy hozzá van rendelve egy terem, amelyben 1-től m_i-ig számozott székek vannak.

Az első körben minden diák odamegy annak a szaknak a terméhez, amelyet elsőnek jelölt meg. Minden s_i szak terménél sorbarendezik a diákokat az s_i szakra szóló pontszámuk alapján (itt most nem lehet holtverseny). Az első m_i ember leül a székekre a sorrendnek megfelelően. A többieket az s_i szak elutasítja, és ők kihúzzák a listájukról ezt a szakot, oda már biztosan nem fogják felvenni őket.

A második körben azok, akik nem ülnek valamelyik teremben, megnézik, hogy melyik a következő szak a rangsorukban, és odamennek annak a szaknak a terméhez. Most minden s_i szak terménél az összes ott lévő diákot (tehát azokat, akik eddig a teremben voltak és azokat, akik most érkeztek) újra rendezzük, és megint az első m_i diák ül le a székekre. Vagyis a most érkezők kiszoríthatják azokat, akik már eddig ott ültek, ha magasabb a pontszámuk (előrébb vannak a szak listáján). Akik a rendezés után nem jutnak székhez, azokat az adott szak elutasítja, és ezek a diákok ezt a szakot kihúzzák a listájukról.

A további körök pontosan ugyanígy zajlanak egészen addig, amíg van még olyan diák, aki nem ül egyik teremben sem és van még a listáján szak. Amikor már nincs ilyen diák, akkor az algoritmus véget ér.

Elképzelhető, hogy az algoritmus legvégén valamelyik terem nem telik meg, maradnak üres székek. Ekkor az adott szakra nem sikerül felvenni a maximálisan lehetséges számú diákot. Sajnos az is elképzelhető, hogy egy diákot az összes listáján lévő szaknál elutasítanak.

A fenti algoritmus a diákoknak kedvez. Ez azt jelenti, hogy ha veszünk egy tetszőleges diákot, és megnézzük az összes lehetséges „stabil párosításban", hogy hová vették volna fel, akkor ez az algoritmus olyan párosítást ad, amely ezek közül a szakok közül a diák által legelőrébb rangsorolt szakot rendeli a diákhoz.

Belátható, hogy ez az algoritmus véges sok lépésben véget ér, és megfelel annak a feltételnek, amit a legelején megfogalmaztunk. Ennek átgondolását az olvasóra bízzuk.

Fontos felhívni a figyelmet arra a momentumra, hogy ha valaki egy adott terembe csak egy későbbi körben érkezik, de magas pontszáma van, akkor kiszorítja azokat, akik esetleg már körök óta ott vannak (vagyis a listájukon előrébb szerepelt az a szak), de alacsonyabb a pontszámuk, mint a később érkezőnek. Ez az oka annak, hogy nem érdemes azt a taktikát követni ebben a rendszerben, hogy első helyre olyan szakot ír a felvételiző, ahová jó eséllyel felveszik, és a rizikósabb szakokat hátrább rangsorolja. Azt kell az első helyre írni, ahová a felvételiző a leginkább menni szeretne, függetlenül attól, hogy mekkora az esélye az adott szakra bekerülni. Semmilyen mértékben nem rontja a felvételi esélyeit a diák egy hátrébb rangsorolt szakon azzal, hogy voltak számára előrébb rangsorolt szakok is.

Végezetül ejtsünk pár szót a holtversenyekről, mert Magyarországon elképzelhető, hogy több diáknak is ugyanannyi pontja van egy adott szakra. Az aktuális szabály az, hogy az azonos ponttal rendelkező diákokkal szemben egyformán kell viselkednie a szaknak, vagyis ha két diáknak ugyanaz a pontszáma, akkor vagy mindkettőt fel kell venni, vagy mindkettőt el kell utasítani. Ez nem elhanyagolható nehézség, ha ezt a problémát is kezelni akarjuk, akkor egy bonyolultabb algoritmust kell választanunk.

A holtversenyek kezelésére a különböző országok különböző stratégiákat alkalmaznak. Van, ahol az a szabály, hogy ha a határon holtverseny van, akkor mindenkit fel kell venni az adott pontszámmal, azaz növelni kell a megfelelő keretszámot. Van, ahol épp az ellenkezője történik, vagyis akkor az adott pontszámmal senkit nem vesznek fel. Előfordul olyan is, ahol ilyen esetekben sorsolással döntenek.

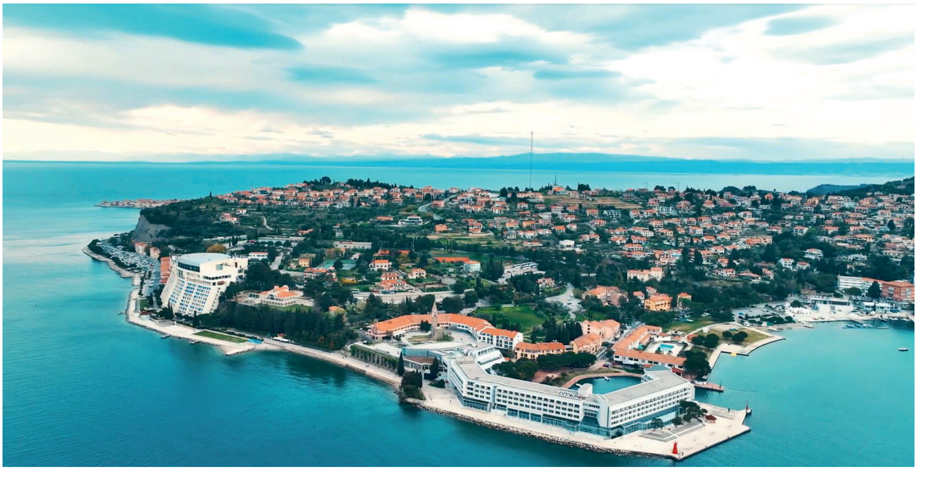
A holtversenyek problémáját el lehet úgy kerülni, hogy a pontszámítás lényegében kizárja, hogy azonos pontszáma legyen két diáknak. Illetve az is egy létező módszer, hogy a pontszám mellett egy második tényező (pl. születési dátum) is szerepet játszik a szigorú rangsor kialakításában, mert elég valószínűtlen az, hogy két azonos pontszámú diák napra pontosan egykorú is legyen.

Magyarországon még legalább két tényező bonyolítja a helyzetet. Az egyik, hogy vannak olyan szakok, ahol alsó korlátja is van a létszámnak, vagyis a szak nem indul el, ha nem tudnak megfelelő számú hallgatót felvenni. A másik, hogy az államilag finanszírozott helyek teljes számára is létezik egy közös felső korlát.

Juhász Péter

Beszámoló a 2023-as EGMO versenyről

Az idei Európai Leány Matematikai Diákolimpia (EGMO) 2023. április 13. és 19. között került megrendezésre Szlovéniában, Portorožban. Idén 55 ország 213 versenyzője kezdett neki a feladatok megoldásának.



Portorož

A magyar csapat nagyon szép teljesítménnyel, két arany-, egy ezüst- és egy bronzéremmel tért haza, ezzel az összes (55) résztvevő ország között 9., míg az európai országok (38) között 6. helyezést szerezve. Az eredmények:
Fülöp Csilla (Szegedi Radnóti Miklós Kís. Gimn.) 40 ponttal aranyérmet,
Sztranyák Gabriella (Budapest, Berzsenyi Dániel Gimn.) 38 ponttal aranyérmet,
Wiener Anna (Budapesti Fazekas Mihály Gyak. Ált. Isk. és Gimn.) 35 ponttal ezüstérmet,