



# Rundele #26 - #40

În cadrul acestui articol vom publica detalii despre desfășurarea rundelor #26-#40 ale ediției 2003/2004 a concursului de programare Bursele Agora, organizat de revista noastră.

## P030425: Perechi

Există mai multe metode de rezolvare pentru această problemă. O variantă care duce la obținerea punctajului maxim este strategia *greedy* care va fi descrisă în continuare.

Pornind de la perechea  $(a, b)$  vom încerca să determinăm toate perechile  $(a', b')$  care pot fi obținute prin operații de scădere (prima operație permisă) și inversare (a treia operație permisă) numai dacă scăderea ar duce la obținerea unei valori negative.

La fiecare pas vom transforma perechea curentă  $(a', b')$  în perechea  $(b', a')$  dacă  $(a' < b')$  sau în perechea  $(a' - b', b')$  în caz contrar. Ne vom opri în momentul în care vom obține valoarea 0.

În continuare vom aplica aceiași pași pentru perechea  $(c, d)$ . Vom păstra toate perechile  $(a', b')$  și  $(c', d')$  obținute.

Evident, pentru fiecare astfel de pereche este cunoscut numărul de pași necesari obținerii ei.

În cazul în care identificăm două perechi  $(a'', b'')$  și  $(c'', d'')$  pentru care  $a'' = c''$  și  $b'' = d''$  sau pentru care  $a'' = d''$  și  $b'' = c''$  vom ști că perechea  $(c, d)$  poate fi obținută prin transformări succesive pornind de la perechea  $(a, b)$ . În cea de-a doua situație ( $a'' = d''$  și  $b'' = c''$ ) va trebui să efectuăm o inversare suplimentară.

Este ușor de observat că vom efectua scăderi și inversări până în momentul în care obținem perechea  $(a'', b'')$ , eventual efectuăm încă o inversare și apoi, vom efectua adunări și inversări până în momentul în care obținem perechea  $(c, d)$ .

Evident, dintre toate perechile  $(a'', b'')$  care satisfac proprietatea amintită, o vom alege pe cea pentru care numărul operațiilor efectuate este minim.

### Au rezolvat corect...

Problema s-a dovedit a fi mult mai dificilă decât ar părea la prima vedere; doar trei dintre cei 32 de concurenți care au participat la această rundă au obținut punctajul maxim:

**Mugurel-Ionuț Andreica, România**  
**Marius Dumitran, România**  
**Sorin Stancu-Mara, România**

## P030426: Triunghi

Folosind teorema lui *Pitagora* observăm imediat că va trebui să determinăm numărul perechilor  $(b, c)$  care satisfac următoarele condiții:

- $a^2 + b^2 = c^2$ ;
- $a < b < c$ ;
- $b$  și  $c$  sunt numere întregi.

Pentru început vom transforma prima relație pentru a descoperi o modalitate mai simplă de determinare a perechilor  $(b, c)$ .

Avem:

$$\begin{aligned} a^2 + b^2 &= c^2 \\ a^2 &= c^2 - b^2 \\ a^2 &= (c + b) \cdot (c - b) \end{aligned}$$

Observăm acum că pentru a obține soluții întregi este necesar (dar nu și suficient) ca numărul  $c - b$  să fie divizor al numărului  $a^2$ .

Ca urmare, vom lua în considerare toți divizorii  $d$  ai numărului  $a^2$  și apoi vom rezolva sistemul de ecuații:

$$\begin{aligned} c - b &= d \\ c + b &= a^2 / d \end{aligned}$$

Rezolvând acest sistem de ecuații obținem valorile:

$$\begin{aligned} c &= (a^2 / d + d) / 2 \\ b &= (a^2 / d - d) / 2 \end{aligned}$$

Acum vom verifica dacă aceste valori satisfac cele trei condiții amintite.

Principala dificultate o constituie determinarea valorilor  $d$  deoarece nu poate fi utilizat un algoritm simplu (valoarea  $a^2$  poate fi foarte mare).

Vom determina o descompunere în factori primi a numărului  $a$ . Aceasta va avea forma:

$$a = d_1^{p_1} \cdot d_2^{p_2} \cdot \dots \cdot d_k^{p_k}.$$



Evident, descompunerea în factori primi a numărului  $a^2$  va avea forma:

$$a^2 = d_1^{2p_1} \cdot d_2^{2p_2} \cdot \dots \cdot d_k^{2p_k}.$$

Pe baza acestei descompuneri deducem că orice divizor al numărului  $a^2$  va avea forma:

$$d = d_1^{q_1} \cdot d_2^{q_2} \cdot \dots \cdot d_k^{q_k}.$$

unde:

- $0 \leq q_1 \leq 2 \cdot p_1$ ;
- $0 \leq q_2 \leq 2 \cdot p_2$ ;
- ...
- $0 \leq q_k \leq 2 \cdot p_k$ .

Așadar, problema determinării divizorilor numărului  $a^2$  se reduce la problema determinării produsului cartezian al mulțimilor  $\{0, 1, \dots, 2 \cdot p_1\}$ ,  $\{0, 1, \dots, 2 \cdot p_2\}$ , ...,  $\{0, 1, \dots, 2 \cdot p_k\}$ . Fiecărui element al produsului cartezian îi va corespunde un divizor. De exemplu, divizorul  $d = d_1^{q_1} \cdot d_2^{q_2} \cdot \dots \cdot d_k^{q_k}$  corespunde elementului  $(q_1, q_2, \dots, q_k)$ .

Se observă că numărul mulțimilor pentru care este construit produsul cartezian este de cel mult nouă deoarece produsul primelor zece numere prime este mai mare decât 2.000.000.000 (valoarea maximă a numărului  $a$ ):

$$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 = 6.469.693.230.$$

De asemenea, se observă că suma maximă a puterilor este  $2 \cdot 30 = 60$  deoarece avem  $2^{31} = 2.147.483.648$ .

Deducem că algoritmul de determinare a acestui produs cartezian (și a divizorilor numărului  $a^2$ ) va rula suficient de repede pentru a se respecta limita de timp admisă.

O altă dificultate importantă este dată de precizia calculului. Pentru a nu apărea erori din această cauză era necesară simularea operațiilor cu numere mari.

#### Au rezolvat aproape corect...

Dintre cei 34 de concurenți care au participat la această etapă a concursului, nici unul nu a reușit să obțină punctajul maxim. Cel mai mare punctaj a fost de 70 de puncte și a fost obținut de trei concurenți. Un singur alt concurent a reușit să obțină un punctaj de peste 50 de puncte.

<i>Liviu Ciortea, România</i>	70
<i>Marius Dumitran, România</i>	70
<i>Alexandru Popa, România</i>	70
<i>Sorin Stancu-Mara, România</i>	51

#### P030427: Fotbal

Vom lua în considerare toate submulțimile care se pot forma folosind primele  $[n/2]$  oi și le vom sorta în funcție de valoarea fotbalistică totală a oilor care fac parte dintr-o submulțime.

În continuare vom determina toate submulțimile formate din ultimele  $n - [n/2]$  oi. Pentru fiecare astfel de

submulțime  $A$ , vom căuta în lista primelor submulțimi (folosind căutarea binară) acea submulțime  $B$  care are valoarea cea mai apropiată de  $S/2 - x$ , unde  $S$  este valoarea fotbalistică însumată a tuturor oilor, iar  $x$  este valoarea fotbalistică totală a oilor din mulțimea  $A$ .

Echipa "candidată" va fi obținută prin reuniunea mulțimilor  $A$  și  $B$ . Dacă valoarea fotbalistică a echipei este mai apropiată de  $S/2$  decât cea mai bună soluție obținută anterior, atunci avem acum o nouă diferență minimă și o singură posibilitate (deocamdată) de a o obține. Dacă diferența este egală cu cea curentă, atunci vom crește numărul posibilităților.

În final vom scrie valoarea diferenței minime, precum și numărul posibilităților prin care aceasta poate fi obținută.

#### A rezolvat corect...

Un singur participant (dintre cei 43 care au trimis soluții la această problemă) a reușit să obțină punctajul maxim:

*Marius Nicolae, România*

#### P030428: Dispozitive de comunicare

Dispozitivele de comunicare pot fi considerate a fi nodurile unui graf orientate ale cărui arce descriu legăturile de comunicare (și direcțiile acestora).

În aceste condiții problema se reduce la determinarea numărului componentelor tare-conexe într-un graf orientat.

#### Au rezolvat corect...

Dintre cei 34 de participanți la această rundă a concursului de programare *Bursele Agora*, 14 au reușit să obțină punctajul maxim:

*Chattopadhyay Amit, India*  
*Mugurel-Ionuț Andreica, România*  
*Andrei Blaj, România*  
*Liviu Ciortea, România*  
*Claudiu-Dan Gheorghe, România*  
*Marius Nicolae, România*  
*Mircea Pașoi, România*  
*Daniel Păsăilă, România*  
*Csaba Pătaș, România*  
*Alexandru Peptan, România*  
*Vlad Petcu, România*  
*Alexandru Popa, România*  
*Claudiu Rad, România*  
*Adrian Vladu, România*

#### P030429: Media aritmetică

Se observă imediat că valoarea căutată este chiar media aritmetică a celor  $n$  numere date. Dacă notăm valoarea căutată prin  $m$ , atunci suma celorlalte  $n - 1$  numere este  $(n - 1) \cdot m$ . Suma tuturor celor  $n$  numere va fi  $(n - 1) \cdot m + m = n \cdot m$ , iar media lor este  $n \cdot m / n = m$ .



După determinarea mediei, vor trebui căutate toate aparițiile sale în fișierul de intrare.

Principale dificultate o constituie faptul că numerele din fișierul de intrare nu pot fi memorate. Ca urmare, fișierul va trebui citit de două ori (o dată pentru determinarea mediei și o dată pentru determinarea aparițiilor). Din aceste motive, pentru citirea și scrierea datelor trebuie utilizate *buffer*-e de dimensiuni mari. În caz contrar, programele nu vor respecta limita de timp admisă.

#### Au rezolvat corect...

La această a treizecea etapă a concursului am primit 38 de rezolvări. Patru dintre cei care au participat la această rundă au obținut punctajul maxim:

*Mugurel-Ionuț Andreica, România*  
*Adrian Cârțu, România*  
*Mihail-Cosmin Piț-Rada, România*  
*Sorin Stancu-Mara, România*

#### P030430: Keops

Problema se reduce la determinarea unui simplu drum într-un labirint, deci poate fi rezolvată folosind, de exemplu, algoritmul lui Lee.

Numărul minim de minute va fi dat de partea întreagă superioară a jumătății distanței minime dintre pozițiile inițiale ale celor doi hoți.

#### Au rezolvat corect...

Dintre cei 27 de concurenți care au trimis soluții la această problemă, 12 au obținut punctajul maxim:

*Mugurel-Ionuț Andreica, România*  
*Adrian Cârțu, România*  
*Liviu Ciortea, România*  
*Adrian Diaconu, România*  
*Kazmenko Ivan, Federația Rusă*  
*Kiril Minkov, Bulgaria*  
*Mihai Moalfă, România*  
*Marius Nicolae, România*  
*Csaba Pățaș, România*  
*Mihail-Cosmin Piț-Rada, România*  
*Alexandru Popa, România*  
*Claudiu Rad, România*

#### P030431: Colectare

Pozițiile obiectelor din labirint pot fi considerate a fi vârful unui graf neorientat. Cu ajutorul algoritmului lui Lee pot fi determinate costurile muchiilor dintre aceste noduri (costul unei muchii este dat de distanța minimă dintre obiectele corespunzătoare nodurilor care reprezintă extremitățile muchiei respective).

Vom adăuga un nod corespunzător poziției inițiale care va fi legat, în același mod, prin muchii (cu aceeași semnificație) de nodurile corespunzătoare obiectelor. În sfârșit, vom adăuga un ultim nod care va corespunde ieșirii

din labirint. Acest nod va fi legat de toate celelalte noduri prin muchii ale căror lungime va fi dată de numărul minim de pași necesari pentru a ieși din labirint pornind din pozițiile corespunzătoare celorlalte noduri ale grafului.

În aceste condiții problema se reduce la determinarea unui drum hamiltonian de cost minim care pornește din nodul corespunzător poziției inițiale a lui *Merlin* și ajunge în nodul corespunzător ieșirii din labirint.

Se observă că graful construit are cel mult șapte noduri, deci nu apar probleme dacă folosim un algoritm exponențial pentru determinarea drumului hamiltonian.

#### Au rezolvat corect...

La această rundă au participat 38 de concurenți dintre care 18 au obținut punctajul maxim:

*Chattopadhyay Amit, India*  
*Mugurel-Ionuț Andreica, România*  
*Andrei Blaj, România*  
*Adrian Cârțu, România*  
*Liviu Ciortea, România*  
*Marius Dumitran, România*  
*Kazmenko Ivan, Federația Rusă*  
*Haryanto Lego, Statele Unite ale Americii*  
*Kiril Minkov, Bulgaria*  
*Mihai Moalfă, România*  
*Mircea Pașoi, România*  
*Daniel Păsăilă, România*  
*Csaba Pățaș, România*  
*Mihail-Cosmin Piț-Rada, România*  
*Tudor-Alexandru Răbaea, România*  
*Claudiu Rad, România*  
*Andrei-Ionuț Tudor, România*  
*Adrian Vladu, România*

#### P030432: Lego

Aceasta este o problemă clasică pentru a cărei rezolvare trebuie aplicată metoda *backtracking*. Evident, un *backtracking* simplu (chiar folosind o mulțime de optimizări) nu ar fi dus la obținerea unui punctaj foarte mare fiindcă limita de timp ar fi fost depășită.

Puteau fi utilizate diferite tehnici euristice sau a unor algoritmi probabilistici cu ajutorul cărora s-ar fi obținut punctaje mai mari.

#### Au rezolvat aproape corect...

Dintre cei 30 de concurenți care au participat la această etapă a concursului, nici unul nu a reușit să obțină punctajul maxim. Cel mai mare punctaj a fost de 58 de puncte și a fost obținut de trei concurenți. Un singur alt concurent a reușit să obțină un punctaj de peste 50 de puncte.

<i>Adina Pampu, România</i>	58
<i>Alexandru Peptan, România</i>	58
<i>Claudiu Rad, România</i>	58
<i>Mugurel-Ionuț Andreica, România</i>	57

### P030433: Cele două tipuri de turnuri

Pentru început vom determina teritoriile celor două specii. Acestea sunt date de înfășurătorile convexe ale punctelor care reprezintă turnurile. După determinarea acestor înfășurători putem verifica foarte simplu, pentru fiecare turn în parte, dacă se află în interiorul teritoriului celeilalte specii (vom utiliza un algoritm de verificare a faptului dacă un punct se află în interiorul unui poligon convex).

#### Au rezolvat corect...

Dintre cei 25 de participanți la această etapă, opt au reușit să obțină numărul maxim de puncte.

*Chattopadhyay Amit, India*  
*Mugurel-Ionuț Andreica, România*  
*Andrei Cârțu, România*  
*Vasanth Jai, India*  
*Haryanto Lego, Statele Unite ale Americii*  
*Kiril Minkov, Bulgaria*  
*Csaba Pătcăș, România*  
*Sorin Stancu-Mara, România*

### P030434: 008 - James Bomb

De această dată avem de a face cu un labirint tridimensional. Cu toate acestea algoritmul aplicat este același (Lee).

Vom determina distanța până la bombă și, în cazul în care *James Bomb* poate ajunge lângă bombă înainte ca aceasta să explodeze, vom scrie rezultatul în fișierul de ieșire.

În caz contrar, vom determina distanța minimă din poziția inițială până la marginea clădirii.

#### Au rezolvat corect...

Problema s-a dovedit a fi relativ simplă; 22 dintre cei 38 de concurenți care au participat la această rundă au obținut punctajul maxim:

*Adrian Diaconu, România*  
*Silviu-Ionuț Ganceanu, România*  
*Sergiu Hlihor, România*  
*Vasanth Jai, India*  
*Andrei Janca, România*  
*Stelian Kelemen, România*  
*Haryanto Lego, Statele Unite ale Americii*  
*Kiril Minkov, Bulgaria*  
*Mihai Moalfă, România*  
*Mircea Pașoi, România*  
*Daniel Păsăilă, România*  
*Mihail-Cosmin Piț-Rada, România*  
*Claudiu Rad, România*  
*Paul-Horațiu Stan, România*  
*Valentin Stanciu, România*  
*Sorin Stancu-Mara, România*  
*Andrei-Ionuț Tudor, România*  
*Adrian Vladu, România*

### P030435: Raze LASER

Pentru fiecare particulă radioactivă vom calcula tangenta unghiului format de dreapta care o unește de origine cu axa orizontală.

Vom alege apoi particulele pentru care obținem cea mai mică și cea mai mare valoare a tangentei și vom determina unghiul format de dreptele care unesc cele două particule de origine.

Trebuie observat faptul că în locul tangentei se poate utiliza orice altă funcție  $f$  care păstrează ordinea (în cazul în care  $a < b$ , avem întotdeauna fie  $f(a) < f(b)$ , fie  $f(b) < f(a)$ ). Așadar, se pot utiliza funcțiile sinus sau cosinus, dar și altele care satisfac proprietatea amintită.

#### Au rezolvat corect...

Problema a fost una dintre cele mai simple probleme propuse spre rezolvare la această ediție a concursului nostru; aproape trei sferturi (23) dintre cei 31 de concurenți care au participat la această rundă au reușit să obțină punctajul maxim:

*Chattopadhyay Amit, India*  
*Mugurel-Ionuț Andreica, România*  
*Adrian Cârțu, România*  
*Cristian Cernat, România*  
*Liviu Ciortea, România*  
*Adrian Diaconu, România*  
*Marius Dumitran, România*  
*Silviu-Ionuț Ganceanu, România*  
*Claudiu-Dan Gheorghe, România*  
*Sergiu Hlihor, România*  
*Andrei Janca, România*  
*Haryanto Lego, Statele Unite ale Americii*  
*Kiril Minkov, Bulgaria*  
*Adina Pampu, România*  
*Ciprian Păduraru, România*  
*Csaba Pătcăș, România*  
*Alexandru Peptan, România*  
*Mihail-Cosmin Piț-Rada, România*  
*Bogdan-Ștefan Popescu, România*  
*Tudor-Alexandru Rabaea, România*  
*Claudiu Rad, România*  
*Sorin Stancu-Mara, România*  
*Adrian Vladu, România*

### P030436: Sertare

Se observă imediat că putem considera o bilă neagră ca fiind o paranteză deschisă și o bilă albă ca fiind o paranteză închisă. În aceste condiții problema se reduce la determinarea numărului de parantezări corecte pentru  $n$  perechi de paranteze. Această valoare este dată de cunoscutul număr al lui Catalan.

#### Au rezolvat corect...

Problema era foarte cunoscută, motiv pentru care a fost rezolvată corect de mulți participanți. Dintre cei 28 de con-





curenți care au trimis soluții, 16 au reușit să obțină punctajul maxim.

*Chattopadhyay Amit, India*  
*Mugurel-Ionuț Andreica, România*  
*Andrei Blaj, România*  
*Adrian Cârțu, România*  
*Liviu Ciortea, România*  
*Adrian Diaconu, România*  
*Marius Dumitran, România*  
*Silviu-Ionuț Ganceanu, România*  
*Sergiu Hlihor, România*  
*Vasanth Jai, India*  
*Haryanto Lego, Statele Unite ale Americii*  
*Mihai Moalfă, România*  
*Mircea Pașoi, România*  
*Daniel Păsăilă, România*  
*Mihail-Cosmin Piț-Rada, România*  
*Tudor-Alexandru Rabaea, România*

### P030437: Rețea de calculatoare

Problema se reduce la determinarea valorii unei tăieturi minime într-un graf neorientat. Deoarece potrivit teoremei *flux maxim - tăietură minimă* această valoare este egală cu cea a fluxului maxim, putem aplica orice algoritm de determinare a acestuia.

#### Au rezolvat corect...

Și această problemă este foarte cunoscută și a fost rezolvată corect de o mare parte dintre cei care au trimis soluții. Dintre cei 28 de concurenți, 19 au reușit să obțină punctajul maxim:

*Mugurel-Ionuț Andreica, România*  
*Andrei Blaj, România*  
*Adrian Cârțu, România*  
*Cristian Cernat, România*  
*Liviu Ciortea, România*  
*Marius Dumitran, România*  
*Silviu-Ionuț Ganceanu, România*  
*Andrei Gönczi, România*  
*Stelian Kelemen, România*  
*Haryanto Lego, Statele Unite ale Americii*  
*Kiril Minkov, Bulgaria*  
*Mircea Pașoi, România*  
*Ciprian Păduraru, România*  
*Daniel Păsăilă, România*  
*Alexandru Popa, România*  
*Bogdan-Ștefan Popescu, România*  
*Claudiu Rad, România*  
*Sorin Stancu-Mara, România*  
*Alexandru-Cosmin Vlăjoagă, România*

### P030438: Extratereștri

Deși se observă imediat că problema cere determinarea celui mai lung subșir comun a două șiruri de caractere,

apar dificultăți datorate faptului că nu putem folosi versiunea clasică de rezolvare bazată pe metoda programării dinamice. Ca urmare, pentru a rezolva problema trebuia utilizat algoritmul de determinare a celui mai lung subșir comun folosind un spațiu de memorie liniar.

#### Au rezolvat corect...

Dintre cei 35 de participanți la această rundă a concursului de programare *Bursele Agora*, zece au reușit să obțină punctajul maxim:

*Chattopadhyay Amit, India*  
*Mugurel-Ionuț Andreica, România*  
*Andrei Blaj, România*  
*Andrei Cârțu, România*  
*Adrian Diaconu, România*  
*Vasanth Jai, India*  
*Haryanto Lego, Bulgaria*  
*Daniel Păsăilă, România*  
*Mihail-Cosmin Piț-Rada, România*  
*Adrian Vladu, România*

### P030439: Matrice magică

O posibilitate de rezolvare a acestei probleme constă în construirea a două matrice  $X$  și  $Y$  pe baza matricei  $A$ . Valorile  $x_{ij}$  vor reprezenta cele mai mari numere divizibile cu  $p$  care sunt mai mici decât  $a_{ij}$ , iar valorile  $y_{ij}$  vor reprezenta cele mai mari numere divizibile cu  $p$  care sunt mai mici decât  $a_{ij}$ . Vom numi matricele  $X$  și  $Y$  *matrice inferioară*, respectiv *matrice superioară*.

*Matricea inferioară* va fi bordată cu sumele pe linii și pe coloane ale matricei  $A$ , iar *matricea superioară* va fi bordată cu sumele pe linii și coloane ale matricei  $Y$ . Elementele celor două matrice se împart la  $p$  și se calculează diferența lor. Se obține astfel o matrice binară care este bordată de anumite sume. Astfel problema se reduce la determinarea muchiilor unui graf pentru care se cunosc gradele nodurilor. Așadar, am obținut o problemă care se rezolvă la fel ca problema *Pluto* propusă spre rezolvare la runda #05 a acestei ediții a concursului.

#### Au rezolvat corect...

La această a patruzecoa etapă a concursului am primit 25 de rezolvări. Zece dintre cei care au participat la această rundă au obținut punctajul maxim:

*Mugurel-Ionuț Andreica, România*  
*Adrian Cârțu, România*  
*Adrian Diaconu, România*  
*Marius Dumitran, România*  
*Silviu-Ionuț Ganceanu, România*  
*Kiril Minkov, Bulgaria*  
*Mircea Pașoi, România*  
*Csaba Pățaș, România*  
*Alexandru Popa, România*  
*Adrian Vladu, România*