Rundele #06 - #15

În cadrul acestui articol vom publica detalii despre desfășurarea rundelor 06-15 ale ediției 2003/2004 a concursului de programare Bursele Agora organizat de revista noastră.

P010416: Arena din Erathia

Vom împărți coordonatele creaturilor în mai multe mulțimi, fiecare dintre aceste mulțimi conținând coordonate ale unor creaturi care fac parte din aceeași rasă.

Pentru fiecare dintre aceste mulțimi vom determina cea mai îndepărtată pereche de puncte. Cea mai defavorizată rasă va fi cea pentru care distanța dintre cele mai îndepărtate puncte va fi maximă.

Cea mai simplă metodă de rezolvare a acestei probleme constă în determinarea distanțelor dintre toate perechile de puncte (vor fi $N \cdot (N - 1) / 2$ astfel de perechi) și alegerea celei mai mari dintre ele. Am notat cu N numărul punctelor dintr-o mulțime.

Așadar, problema se reduce la o simplă determinare a unei valori maxime dintre $N \cdot (N - 1) / 2$, valori care sunt calculate. Evident, ordinul de complexitate al acestui algoritm este $O(N^2)$.

Din nefericire, acest ordin de complexitate nu este acceptabil pentru această problemă, deoarece putem avea până la 5000 de puncte.

Pentru a obține un algoritm eficient, vom observa mai întâi faptul că puncte le care formează cea mai îndepărtată pereche se află, obligatoriu, pe înfășurătoarea convexă a mulțimii de puncte.

Vom demonstra această afirmație prin *metoda reducerii la absurd*. Să presupunem că unul dintre cele două puncte nu se află pe înfășurătoarea convexă, așadar se află în interiorul acesteia. Vom nota acest punct prin Q, iar celălalt punct prin Q. Dacă vom prelungi segmentul care unește punctele P și Q, atunci acesta va intersecta o latură a înfășurătorii convexe sau va trece printr-un alt punct al acesteia.

În acest din urmă caz se observă imediat că distanța de la P la Q este mai mică decât distanța de la P la punctul respectiv, deci punctele P și Q nu se pot afla la distanța maximă.

Pentru primul caz vom considera că latura intersectată are ca extremități punctele R și S și prelungirea segmentului care unește punctele P și Q intersectează latura respectivă în punctul T.

Pentru început se observă faptul că lungimea segmentului care unește punctele P și T este mai mare decât cea care unește punctele P și Q.

Vom considera acum triunghiul PRS (determinat de vârfurile P, R și S). Punctul T se va afla pe latura RS a acestui triunghi. Există o teoremă care afirmă că cel puțin una dintre laturile PR și PS are lungimea mai mare decât segmentul PT. Ca urmare una dintre laturile triunghiului va fi mai lungă decât segmentul PT care, la rândul său, este mai lung decât segmentul PQ.

Datorită faptului că această latură are ca extremități două vârfuri ale înfășurătorii convexe, deducem imediat că punctele P și Q nu se pot afla la distanța maximă.

În concluzie, primul pas necesar pentru a determina punctele aflate la distanța maximă este determinarea unei înfășurători convexe.

O variantă care poate fi utilizată în continuare este considerarea succesivă a vârfurilor înfășurătorii și determinarea, printr-o metodă similară cu căutarea binară, a celui mai îndepărtat vârf față de vârful considerat.

Se va considera nodul din stânga (sau dreapta) nodului ales. Atâta timp cât distanța crește, la fiecare pas k, vom "sări" la al 2^k-lea nod. În momentul în care distanța începe să scadă, vom reveni la nodul anterior și vom începe să înjumătățim "distanța" saltului.

Dacă am ajuns la pasul l, acum, la fiecare, pas vom descrește cu 1 puterea. Dacă la un anumit pas, puterea este p, vom efectua un salt la al 2^p-lea nod. Dacă distanța descrește vom reveni, iar dacă nu, vom continua de la vârful respectiv.

În final, vom obține vârful aflat la distanța maximă față de nodul considerat. După considerarea tuturor vârfurilor, vom obține distanța maximă între două puncte din mulțimea dată.

Algoritmul descris funcționează corect deoarece, parcurgând nodurile într-un anumit sens distanța începe să crească până la un moment dat, după care ea începe să scadă.

O a doua variantă constă în alegerea unui vârf P_1 al înfășurătorii și parcurgerea acesteia (începând cu punctul din stânga sau din dreapta sa) atâta timp cât distanța crește. Vom nota punctul respectiv prin P. În continuare vom trece la punctul P_2 (cel aflat pe înfășurătoare imediat lângă punctul P_1, în funcție de sensul ales) și vom continua parcurgerea.

Poate fi ușor observat faptul că nu are sens să mai verificăm toate punctele parcurse la pasul anterior (cele până la P) deoarece distanțele obținute vor fi mai mici. Așadar, vom continua de la punctul P atâta timp cât distanța va crește.

Vom trece apoi la punctul P_3 și vom repeta în continuare procedeul până în momentul în care am luat în considerare toate punctele de pe înfășurătoare.

Evident, în final vom cunoaște distanța maximă dintre două puncte de pe înfășurătoare și, implicit, distanța maximă dintre două puncte ale mulțimii date.

Determinarea înfășurătorii convexe are ordinul de complexitate $O(N \cdot \log N)$, dacă utilizăm scanarea *Graham* sau $O(N \cdot h)$, dacă utilizăm potrivirea *Jarvis*. (Am notat prin h numărul punctelor aflate pe înfășurătoarea convexă a mulțimii de puncte.)

În situația în care, după determinarea înfășurătorii, vom folosi metoda de căutare binară, distanța maximă va fi determinată într-un timp de ordinul $O(h \cdot \log h)$, datorită faptului că vom efectua h căutări binare, fiecare având ordinul de complexitate $O(\log h)$.

În cazul în care vom utiliza cea de-a doua metodă, aceasta va avea ordinul de complexitate $O(h)$, deoarece se poate observa faptul că prin avansări se va parcurge de cel mult două ori înfășurătoarea.

Ca urmare, ordinul de complexitate al algoritmului de determinare a unei perechi de puncte aflate la distanță maximă este același cu cel al algoritmului folosit pentru determinarea înfășurătorii convexe.

Au rezolvat corect...

Dintre cei 81 de participanți la etapa a șasea a concursului, 17 au reușit să obțină punctajul maxim:

Chattopadhyay Amit, India
Mugurel-Ionuț Andreica, România
Adrian Cârcu, România
Adrian-Romulus Ciripan, România
Adrian Diaconu, România
Silviu-Ionuț Gănceanu, România
Ivan Kazmenko, Federația Rusă
Haryanto Lego, Statele Unite ale Americii
Kiril Minkov, Bulgaria
Mihai Moalfă, România
Cosmin Molea, România
Andrei Păduraru, România
Daniel Păsăilă, România
Csaba Pătcaș, România
Claudiu Rad, România
Andrei-Ionuț Tudor, România
Sorin Vultureanu, România

P010417: Merlin

Problema s-a dovedit destul de dificilă, dovadă fiind numărul mic al concurenților care au reușit să obțină punctajul maxim.

O posibilă abordare, care ar fi dus la obținerea punctajului maxim, constă în folosirea *metodei greedy* pentru a obține o configurație care să conțină un număr cât mai mare de pietricele.

După obținerea acestei configurații, ea putea fi îmbunătățită cu ajutorul metodei *backtracking*.

Au rezolvat corect...

Dintre cei 58 de concurenți care au participat la această etapă a concursului, doar patru au reușit să obțină punctajul maxim:

Mugurel-Ionuț Andreica, România
Sergiu Hlihor, România
Daniel Păsăilă, România
Csaba Pătcaș, România

P010418: Distribuție

Pentru a identifica orașele deservite de prima tipografie (respectiv de a doua) vom determina, mai întâi, distanțele minime de la prima tipografie (orașul identificat prin 1) la toate orașele, respectiv de la a doua tipografie (orașul identificat prin 2) la toate orașele.

În acest scop putem utiliza un algoritm pentru determinarea drumurilor minime de la un nod al unui graf la toate celelalte (*Dijkstra*). Am considerat orașele ca fiind nodurile unui graf neorientat ale cărui muchii reprezintă străzile dintre orașe, iar costurile muchiilor sunt date de lungimile străzilor.

Pentru fiecare oraș i vom cunoaște distanțele d_{1i} și d_{2i} față de cele două tipografii.

Vom calcula valorile $d_i = d_{1i} - d_{2i}$ pentru toate orașele și vom sorta orașele în ordine crescătoare în funcție de aceste valori.

Primele N / 2 orașe (în ordinea determinată) vor fi deservite de prima tipografie, iar celelalte orașe de către cea de-a doua.

Au rezolvat corect...

Dintre cei 43 de participanți la cea de-a opta etapă, aproape jumătate (18) au reușit să obțină punctajul maxim:

Chattopadhyay Amit, India
Mugurel-Ionuț Andreica, România
Andrei Blaj, România
Iulian Blajuț, România
Adrian Diaconu, România
Ioan-Dan Docleanu, România
Silviu-Ionuț Gănceanu, România
Andrei-Gabriel Ion, România
Ivan Kazmenko, Federația Rusă
Vasanth Jai, India
Haryanto Lego, Statele Unite ale Americii
Kiril Minkov, Bulgaria
Mihai Moalfă, România
Dalian Valeriu Moraru, România
Mircea Pașoi, România
Daniel Păsăilă, România
Alexandru Peptan, România
Victor Podeanu, România
Tudor-Alexandru Rabaea, România

P010419: Cerc

Această problemă este foarte asemănătoare cu problema *Petrecere nebunatică* propusă spre rezolvare la etapa regională a concursului *ACM* din acest an. Rezolvarea acestei probleme a fost prezentată îm *GInfo* 13/8 (decembrie 2003).

Au rezolvat corect...

Dintre cei 57 de participanți la cea de-a noua etapă a concursului, doar cinci au obținut punctajul maxim:

Mugurel-Ionuț Andreica, România
Adrian Cârcu, România
Bogdan Nicolae, România
Csaba Pătcaș, România
Valentin Stanciu, România

P010420: Explozie

Problema se reduce la a determina numărul punctelor de coordonate întregi aflate în interiorul unei sfere de rază dată.

Pentru fiecare punct de coordonate întregi pozitive (x, y, 0) aflate în interiorul sferei vom determina cel mai mare număr întreg z astfel încât punctul de coordonate (x, y, z) se află în interiorul sferei.

Valoarea z se obține rezolvând ecuația $x^2 + y^2 + z^2 = r^2$, unde r este raza sferei. Vom considera partea întreagă [z] a soluției pozitive a acestei ecuații.

În cazul în care valorile x și y sunt nenule, vom adăuga valoarea $4 \cdot [z] + 2$ la numărul de puncte din interiorul sferei. Dacă doar una dintre valorile x și y este nulă, numărul punctelor adăugate va fi $2 \cdot [z] + 1$. În situația în care ambele valori sunt nule, valoarea adăugată este $[z] + 1$.

Au rezolvat corect...

La cea de-a zecea etapă a concursului nostru au participat 43 de concurenți. Dintre aceștia doar trei au obținut numărul maxim de puncte:

Mugurel-Ionuț Andreica, România
Adrian Cârcu, România
Ivan Kazmenko, Federația Rusă

P010421: Recunoașterea caracterelor

Pentru a compara litera care trebuie recunoscută cu o literă dată vom construi un graf bipartit. Nodurile din prima mulțime vor fi date de elementele cu valoarea 1 din matricea corespunzătoare primei litere, iar nodurile din a doua mulțime vor fi date de elementele cu valoarea 1 din matricea corespunzătoare celei de-a doua litere.

Fiecare nod din prima mulțime va fi unit printr-o muchie de toate nodurile din cea de-a doua. Costurile muchiei corespunzătoare unei perechi de noduri este dată de distanța *Manhattan* dintre punctele corespunzătoare, iar capacitatea sa este 1.

În acest graf bipartit vom determina un cuplaj maxim de cost minim. Costul acestui cuplaj reprezintă distanța dintre matrice.

Pentru a ne asigura că este respectată condiția care impune faptul ca unui element dintr-o matrice să îi corespundă $[p / q]$ sau, eventual, $[p / q] + 1$ puncte din cealaltă, capacitățile muchiilor care vor uni nodul sursă (cel introdus pentru determinarea cuplajului cu ajutorul fluxului maxim) de nodurile din prima mulțime, vor fi date de partea întreagă superioară a raportului p / q. Valorile p și q au semnificația din enunț. După determinarea tuturor cuplajelor maxime de cost minim, rezultatul va fi litera corespunzătoare cuplajului cu cel mai mic cost.

Au rezolvat corect...

Doar 36 de participanți au trimis soluții la această rundă. Dintre aceștia, cinci au reușit să obțină punctajul maxim.

Mugurel-Ionuț Andreica, România
Ivan Kazmenko, Federația Rusă
Kiril Minkov, Bulgaria
Csaba Pătcaș, România
Alexandru Peptan, România

P010422: Ciobănașul

Pentru a determina numărul pătratelor care au în vârfuri parcele nivelate vom considera toate perechile de linii.

Pentru fiecare pereche vom determina numărul k al coloanelor care conțin valoarea 1 în ambele linii. În acest scop putem utiliza conjuncția logică pentru a mări viteza de execuție. După determinarea valorii k, vom ști că pe cele două linii vom avea $k \cdot (k - 1) / 2$ pătrate.

Vom însuma valorile corespunzătoare fiecărei perechi de linii și vom afișa rezultatul.

Au rezolvat corect...

La această rundă au participat 48 de concurenți dintre care cinci au obținut punctajul maxim:

Chattopadhyay Amit, India
Adrian Cârcu, România
Silviu-Ionuț Gănceanu, România
Vasanth Jai, India
Mihail-Cosmin Piț-Rada, România

P010423: Joc

Pentru început vom observa că o grămadă care conține un număr de elemente care este multiplu de 4 poate fi "câștigată" de al doilea jucător.

Indiferent care este numărul de pietre pe care îl alege primul jucător (acesta nu va putea lua toate pietrele deoarece numărul total nu este prim, fiind un multiplu de 4), cel de-al doilea jucător va putea lua 1, 2 sau 3 pietre pentru ca numărul pietrelor rămase să fie din nou multiplu de patru. Evident, după mutarea primului jucător numărul rămas nu va fi multiplu de 4, deoarece numărul pietrelor luate trebuie să fie 1 sau un număr prim (deci nu poate fi multiplu de 4).

Așadar, strategia primului jucător este ca să obțină o configurație în care toate grămezile să conțină un număr de pietre care este multiplu de 4.

În acest scop se va utiliza strategia pentru jocul *NIM*, dar considerând doar resturile împărțirii la 4 ale numerelor de pietre din grămezi.

Primul jucător va avea o strategie sigură de câștig doar dacă are o strategie pentru această variantă "redusă" a jocului *NIM*.

Au rezolvat corect...

La această etapă au participat 31 de concurenți, dintre care 13 au obținut punctajul maxim.

Chattopadhyay Amit, India
Mugurel-Ionuț Andreica, România
Adrian Cârcu, România
Liviu Ciortea, România
Marius Dumitran, România
Silviu-Ionuț Gănceanu, România
Haryanto Lego, Statele Unite ale Americii
Mihai-Vlad Pantiș, România
Mircea Pașoi, România
Csaba Pătcaș, România
Alexandru Peptan, România
Mihail-Cosmin Piț-Rada, România
Adrian Vladu, România

P010424: Cifre

Această problemă poate fi foarte ușor rezolvată folosind *principiul cutiei lui Dirichlet*.

Vom concatena numărul format din cifrele date cu el însuși până în momentul în care restul împărțirii la n va fi 0 sau obținem un rest care a mai fost obținut la un pas anterior. Pe baza principiului, suntem siguri că vom ajunge în una dintre cele două situații după cel mult n pași.

Dacă am obținut restul 0, avem un rezultat. Dacă nu, rezultatul va fi dat prin simpla scădere a celor două numere care duc la obținerea aceluiași rest prin împărțirea la n. Dacă cele două resturi se obțin la iterațiile p și q, atunci diferența constă în simpla înlocuire a ultimelor $k \cdot (q - p)$ cifre ale rezultatului cu cifra 0 (care face parte cu siguranță dintre cifrele date).

Există un mic amănunt care poate duce la o mică greșeală. Rezultatul nu poate să înceapă cu cifra 0. Pentru a evita această situație este suficient ca, în șirul cifrelor date, să nu punem cifra 0 pe prima poziție.

Au rezolvat corect...

Douăzeci dintre cei 54 de participanți la această etapă au reușit să obțină numărul maxim de puncte.

Chattopadhyay Amit, India
Mugurel-Ionuț Andreica, România
Eduard-Gabriel Băzăvan, România
Liviu Ciortea, România
Adrian Diaconu, România
Marius Dumitran, România
Silviu-Ionuț Gănceanu, România
Ivan Kazmenko, Federația Rusă
Vasanth Jai, India
Haryanto Lego, Statele Unite ale Americii
Mihai Moalfă, România
Marius Nicolae, România
Mihai-Vlad Pantiș, România
Mircea Pașoi, România
Daniel Păsăilă, România
Csaba Pătcaș, România
Mihail-Cosmin Piț-Rada, România
Victor Podeanu, România
Sorin Stancu-Mara, România
Adrian Vladu, România

P010425: Stația spațială

Aceasta a fost cea mai simplă problemă dintre cele propuse spre rezolvare la primele 15 runde. Mai mult de trei sferturi dintre cei care au trimis soluții au reușit să o rezolve corect. Soluția acestei probleme este foarte simplă; se utilizează o strategie *greedy* care constă în considerarea primei nave (*USS Enterprise*) și eliminarea tuturor celor care urmează și se află la o distanță prea mică. Se trece la următoarea navă și se elimină din nou navele aflate la distanță prea mică. Procedeul continuă până în momentul în care se ajunge la ultima navă.

Au rezolvat corect...

Dintre cei 52 de concurenți care au participat la această rundă, 41 au obținut punctajul maxim:

Renald, Indonezia
Chattopadhyay Amit, India
Mugurel-Ionuț Andreica, România
Eduard-Gabriel Băzăvan, România
Iulian Blajuț, România
Adrian Cârcu, România
Dragoș Cavulea, România
Cristian Cernat, România
Liviu Ciortea, România
Adrian-Romulus Ciripan, România
Arthur Costea, România
Adrian Diaconu, România
Marius Dumitran, România
Silviu-Ionuț Gănceanu, România
Adrian Ghiuță, România
Andrei Gönczi, România
Sergiu Hlihor, România
Ing Ing, Indonezia
Kazmenko Ivan, Federația Rusă
Vasanth Jai, India
Andrei Janca, România
Johannes Slotta, Germania
Stelian Kelemen, România
Haryanto Lego, Statele Unite ale Americii
Kiril Minkov, Bulgaria
Marius Nicolae, România
Mircea Pașoi, România
Ciprian Păduraru, România
Daniel Păsăilă, România
Csaba Pătcaș, România
Alexandru Peptan, România
Vlad Petcu, România
Mihail-Cosmin Piț-Rada, România
Alexandru Popa, România
Claudiu Rad, România
Alexandru Stan, România
Valentin Stanciu, România
Sorin Stancu-Mara, România
Andrei-Ionuț Tudor, România
Adrian Vladu, România
Sorin Vultureanu, România