



Rundele #16 - #25

În cadrul acestui articol vom publica detalii despre desfășurarea rundelor #16-#25 ale ediției 2003/2004 a concursului de programare Bursele Agora, organizat de revista noastră.

P020424: Tăiere

Problema se rezolvă relativ ușor folosind o strategie *Greedy*. La fiecare pas vom încerca să eliminăm un grup astfel încât numărul rămas să fie cât mai mic posibil.

Grupul ales este primul care reprezintă un număr care este mai mare decât numărul reprezentat de grupul anterior (precedentele x cartonașe), și decât numărul reprezentat de grupul următor (următoarele x cartonașe).

Pentru a ne asigura că grupurile anterior și următor există întotdeauna, vom adăuga x cartonașe fictive la începutul și la sfârșitul șirului. Pe aceste cartonașe se va scrie cifra 0.

Aparent, apare o problemă cauzată de faptul că după eliminarea a x cartonașe, următoarele x eliminate nu ar forma un grup în șirul inițial. Din fericire, aceasta nu este o problemă reală deoarece în total sunt eliminate $2 \cdot x$ cartonașe aflate pe poziții consecutive, deci pot fi formate două grupuri a câte x cartonașe.

Să considerăm următorul șir de cartonașe:

1 2 9 6 5 4 3 1

din care trebuie eliminate două grupuri a câte trei cartonașe.

Inițial vom adăuga câte trei zerouri la începutul și la sfârșitul șirului. Astfel obținem:

0 0 0 1 2 9 6 5 4 3 1 0 0 0

Potrivit strategiei prezentate, primul grup eliminat este 9 6 5 și după efectuarea acestei operații obținem șirul:

0 0 0 1 2 4 3 1 0 0 0

La al doilea pas vom alege grupul 2 4 3 și vom obține șirul:

0 0 0 1 1 0 0 0

După eliminarea zerourilor de la începutul și de la sfârșitul șirului vom obține rezultatul căutat: 11.

Așadar, cartonașele din primul grup eliminat se află pe pozițiile 3, 4 și 5 în șirul inițial, iar cartonașele din al doilea grup se află pe pozițiile 2, 6 și 7 în acest șir, ceea ce nu este permis conform restricțiilor din enunț.

Totuși, avem șase cartonașe aflate pe pozițiile consecutive 2, 3, 4, 5, 6 și 7, care pot forma două grupuri: pozițiile 2, 3 și 4, respectiv pozițiile 5, 6 și 7.

Ca urmare, grupurile eliminate vor fi, de fapt, 296 și 543, dar rezultatul final este exact același: numărul obținut este 11.

Strategia prezentată permite întotdeauna obținerea celui mai mic număr posibil.

Au rezolvat corect...

Problema s-a dovedit a fi relativ ușor de rezolvat; 29 dintre cei 45 de concurenți care au participat la această rundă au reușit să obțină punctajul maxim:

Chattopadhyay Amit, India
Mugurel-Ionuț Andreica, România
Irinel Besliu, România
Andrei Blaj, România
Iulian Blaj, România
Adrian Cărcu, România
Liviu Ciortea, România
Adrian Diaconu, România
Marius Dumitran, România
Silviu-Ionuț Gănceanu, România
Ștefan Gheorghe, România
Andrei Gönczi, România
Sergiu Hlihor, România
Vasanth Jai, India
Andrei Janca, România
Johannes Slotta, Germania
Haryanto Lego, Statele Unite ale Americii
Kiril Minkov, Bulgaria
Marius Nicolae, România
Mircea Pașoi, România
Daniel Păsăilă, România
Csaba Pătaș, România
Alexandru Peptan, România
Mihail-Cosmin Piț-Rada, România
Alexandru Popa, România
Claudiu Rad, România
Valentin Stanciu, România
Sorin Stancu-Mara, România
Adrian Vladu, România

**P020425: Sume**

Această problemă poate fi rezolvată foarte ușor păstrând un șir de valori booleene care să indice dacă o anumită sumă poate sau nu fi obținută.

Datorită faptului că avem cel mult 100 de numere ale căror valori sunt cel mult 10000, suma maximă care poate fi obținută este 1000000, deci am avea nevoie de 1000001 valori. Ca urmare, va trebui să folosim un singur bit pentru reprezentarea unei sume, ceea ce reduce necesarul de memorie la 125001 de octeți pentru un șir.

Inițial vom considera că nu poate fi obținută decât suma 0. În continuare, la fiecare pas, vom lua în considerare un număr. Vom parcurge șirul sumelor obținute la pasul anterior (bit cu bit) și, pentru fiecare bit cu valoarea 1, la suma corespunzătoare vom adăuga numărul curent. Rezultatul operației va reprezenta o sumă care poate fi obținută cu siguranță, așadar pentru pasul următor va trebui să setăm la valoarea 1 bitul corespunzător noii sume.

În final vom număra biții cu valoarea 1 din întregul șir și vom scrie rezultatul.

Se observă că, dacă parcurgem șirul de la stânga la dreapta (începând cu valoarea 0), în momentul în care bitul corespunzător unei sume devine 1, dacă acesta este setat în același șir, atunci când se va ajunge la poziția respectivă nu vom putea ști dacă bitul a fost setat la pasul curent sau la un pas anterior.

Există cel puțin soluții simple care pot evita astfel de complicații. Se poate folosi un șir de biți auxiliar care va indica sumele obținute la începutul fiecărui pas și noile sume sunt indicate doar în acest șir. După încheierea execuției unui pas șirul auxiliar devine șirul curent. O a doua posibilitate este utilizarea unui șir auxiliar ale cărui valori sunt nule la începutul fiecărui pas. Din nou, noile sume sunt setate în șirul auxiliar, iar la încheierea execuției pasului șirul curent se obține printr-o simplă conjuncție logică între cele două șiruri, iar valorile tuturor biților din șirul auxiliar devin 0.

Totuși, există o soluție mult mai simplă și mai eficientă. Problema amintită anterior este evitată prin simpla parcurgere a șirului de biți de la dreapta spre stânga (dinspre suma maximă spre 0). Fiecare bit nou-setat se va afla la dreapta poziției curente, deci nu va mai fi întâlnit la pasul curent datorită deplasării spre stânga.

Au rezolvat corect...

Dintre cei 46 de concurenți care au participat la această etapă a concursului, doar șapte au reușit să obțină punctajul maxim:

Mugurel-Ionuț Andreica, România

Adrian Cârțu, România

Marius Dumitran, România

Vasanth Jai, India

Mircea Pașoi, România

Mihail-Cosmin Piț-Rada, România

Adrian Vladu, România

P020426: Warp

Problema se reduce la o simplă determinare a celui mai lung dintre drumurile minime dintre toate perechile de noduri ale unui graf neorientat.

Apare o singură diferență: costul unui drum este dat de produsul costurilor muchiilor componente și nu de suma lor.

Această dificultate este evitată foarte ușor prin simpla logaritmare a costurilor muchiilor (obținem astfel drumuri în care înmulțirile sunt transformate în adunări).

În final vom obține logaritmul celui mai lung dintre drumurile minime.

Valoarea drumului minim ar putea fi obținută printr-o exponențiere dar, datorită faptului că această valoare este foarte mare soluția nu este viabilă.

O posibilitate este păstrarea traseelor și reconstituirea drumului obținut. Rezultatul se obține prin înmulțirea costurilor muchiilor de pe acest drum (folosind operații cu numere mari).

Au rezolvat corect...

Doar 31 de concurenți au trimis rezolvări la această etapă; dintre aceștia, șapte au reușit să obțină punctajul maxim:

Mugurel-Ionuț Andreica, România

Adrian Cârțu, România

Silviu-Ionuț Gănceanu, România

Mihai-Vlad Pantiș, România

Mircea Pașoi, România

Mihail-Cosmin Piț-Rada, România

Sorin Stancu-Mara, România

P020427: Găuri

Problema poate fi foarte ușor rezolvată prin folosirea unui simplu algoritm de umplere. Cu ajutorul acestuia vom putea determina toate zonele.

După determinarea zonelor le vom elimina pe cele care nu sunt găuri (conțin cel puțin un element care se află pe prima linie, prima coloană, ultima linie sau ultima coloană) și alegem cea mai mare zonă rămasă.

Pentru simplitate, putem eticheta elementele care formează zonele folosind numere întregi distincte. După identificarea unei zone vom păstra numărul de elemente care o formează (dimensiunea sa) într-un vector.

Vom parcurge apoi prima linie, prima coloană, ultima linie și ultima coloană și vom determina toate etichetele care apar. Pentru fiecare etichetă determinată, vom "elimina" zona respectivă prin setarea dimensiunii sale la 0.

În final vom determina elementul maxim al șirului care reprezintă dimensiunile zonelor.

Bineînțeles, există o mulțime de alte modalități de identificare a zonelor și de eliminare a celor care nu sunt găuri.

De exemplu, se poate porni de pe marginile matricei (se iau în considerare toate elementele de pe prima linie, prima coloană și ultima coloană) și, folosind algoritmul de umplere, se determină zonele care ating marginile și se

atribuie valoarea 1 tuturor elementelor care fac parte din acestea.

Au rezolvat corect...

Dintre cei 29 de participanți la această rundă a concursului de programare *Bursele Agora*, zece au reușit să obțină punctajul maxim:

Adrian Cârcu, România
Liviu Ciortea, România
Marius Dumitran, România
Silviu-Ionuț Gânceanu, România
Kiril Minkov, Bulgaria
Mihai-Vlad Pantiș, România
Mircea Pașoi, România
Mihail-Cosmin Piț-Rada, România
Claudiu Rad, România
Adrian Vladu, România

P020428: Daneel

Călătoria lui *Daneel* este foarte "asemănătoare" cu cea pe care "a efectuat-o" *Bob* în 1998 la ediția a V-a a *Olimpiadei de Informatică a Europei Centrale* care s-a desfășurat la *Zadar*, în *Croația*.

Așadar, această problemă este foarte asemănătoare cu problema *Șosele* de la concursul desfășurat în urmă cu șase ani.

Singura diferență o constituie faptul că *Daneel* poate călători pe un drum în ambele sensuri, spre deosebire de *Bob* care putea călători într-un singur sens.

Enunțul original a fost publicat în *GInfo* 8/8 (noiembrie 1998), iar soluția în *GInfo* 9/1 (ianuarie 1999). Prezentăm în continuare o versiune prescurtată a descrierii soluției.

În primul rând vom reformula problema în termeni din teoria grafurilor:

Se consideră un multigraf ale cărui vârfuri sunt identificate prin numerele 1, 2, ..., n . Între două vârfuri pot exista mai multe muchii. Fiecărei muchii i se atașează o lungime și o taxă care trebuie plătită la traversarea sa. De asemenea, este cunoscută o sumă k . Trebuie să se determine lungimea celui mai scurt drum (dacă acesta există) de la nodul 1 la nodul n pentru care taxa totală plătită nu depășește valoarea k .

Pentru fiecare muchie vom memora extremitățile, lungimea și costul.

Vom construi o listă ale cărei elemente sunt înregistrări care conțin câte trei câmpuri:

- vârful curent la care s-a ajuns (pe un drum care pornește de la vârful 1);
- taxa care a trebuit plătită pe drumul prin care s-a ajuns la nodul respectiv;
- lungimea drumului care pornește de la nodul 1 și ajunge la nodul respectiv.

Vom păstra într-un vector distanțele minime ale drumurilor de la vârful 1 la fiecare dintre celelalte vârfuri care corespund unor drumuri al căror costuri sunt cel mult egale cu k .

Elementele listei sunt întotdeauna ordonate în funcție de cost. În momentul în care se încearcă inserarea unui element în listă, atunci acesta nu va fi adăugat dacă în listă există un alt element care conține informații despre același nod, dar pentru care atât costul, cât și distanța sunt mai mici.

Pentru simplitate, vom ordona muchiile în funcție de una dintre extremități și vom identifica pozițiile la care încep muchiile corespunzătoare fiecărui nod. Orice muchie va fi dublată pentru a permite traversarea sa în ambele sensuri.

Rezolvarea se bazează pe o parcurgere în lățime a grafului, plecând de la lista care conține un singur element pentru care nodul curent este 1, iar distanța și costul sunt ambele 0.

La fiecare pas vom extrage primul element al listei. Elementul nu se expandează dacă distanța corespunzătoare este mai mare decât distanța minimă pentru nodul respectiv. În caz contrar se determină vârfurile în care se poate ajunge din acest nod și se încearcă adăugarea în listă a unor elemente corespunzătoare acestor vârfuri.

Se calculează valorile pentru costuri și distanțe (adăugând distanțele și costurile pentru muchiile corespunzătoare) și se inserează elemente în liste numai dacă noul cost este cel mult egal cu k .

Algoritmul se încheie când lista devine vidă. Rezultatul va fi dat de distanța minimă corespunzătoare nodului identificat prin n .

Au rezolvat corect...

La această a douăzecea etapă a concursului am primit doar 23 de rezolvări. Cinci dintre participanți au obținut punctajul maxim:

Mugurel-Ionuț Andreica, România
Silviu-Ionuț Gânceanu, România
Kiril Minkov, Bulgaria
Mircea Pașoi, România
Sorin Stancu-Mara, România

P020429: Munte

Un număr are aspect de munte dacă pe primele poziții se află cifre în ordine crescătoare, iar pe ultimele poziții se află cifre în ordine descrescătoare.

Pentru a rezolva problema vom determina, pentru fiecare cifră în parte, lungimea celui mai lung subșir crescător care se termină cu cifra respectivă, precum și lungimea celui mai lung subșir descrescător care începe în dreptul cifrei respective.

Pentru fiecare cifră vom calcula sumele celor două lungimi, iar "vârful" muntelui va fi dat de cifra pentru care suma este maximă.





Soluția problemei va fi dată de diferența dintre numărul de cifre al numărului dat și suma respectivă, la care se va adăuga valoarea 1.

Valoarea 1 trebuie adăugată deoarece "vârful" muntelui este considerat de două ori: în șirul crescător și în șirul descrescător.

Au rezolvat corect...

Dintre cei 56 de concurenți care au trimis soluții la această problemă, 37 au obținut punctajul maxim:

Marius Airinei, România
Victor Ambrus, România
Chattopadhyay Amit, India
Mugurel-Ionuț Andreica, România
Eduard-Gabriel Băzăvan, România
Călin Bindea, România
Andrei Blaj, România
Iulian Blaj, România
Adrian Cărcu, România
Cristian Cernat, România
Liviu Ciortea, România
Andrei Deac, România
Adrian Diaconu, România
Marius Dumitran, România
Silviu-Ionuț Gânceanu, România
Claudiu-Dan Gheorghe, România
Andrei-Gabriel Ion, România
Kazmenko Ivan, Federația Rusă
Vasanth Jai, India
Andrei Janca, România
Haryanto Lego, Statele Unite ale Americii
Kiril Minkov, Bulgaria
Mihai Moalfă, România
Cosmin Molea, România
Mihai-Vlad Pantiș, România
Mircea Pașoi, România
Ciprian Păduraru, România
Daniel Păsăilă, România
Csaba Pătcăș, România
Alexandru Peptan, România
Vlad Petcu, România
Mihail-Cosmin Piț-Rada, România
Victor Podeanu, România
Tudor-Alexandru Rabaea, România
Renald, Indonezia
Silviu Julean, România
Adrian Vladu, România

P020430: Puncte

În primul rând vom observa că determinantul folosit pentru calcularea ariei unui triunghi determinat de trei puncte ale căror coordonate sunt întregi trebuie să fie un număr par.

De asemenea, se observă că paritatea determinantului este determinată de paritatea celor șase coordonate.

Din acest motiv vom împărți punctele în patru categorii (ambele coordonate pare, ambele coordonate impare, prima coordonată pară, iar a doua impară și prima coordonată impară, iar a doua pară). Vom număra punctele din fiecare categorie și vom calcula pe baza unei formule simple modul în care se pot alege puncte din diferite categorii pentru a obține un determinant par.

Algoritmul este liniar deoarece toate operațiile se efectuează în timp constant, cu excepția celei de împărțire a punctelor în cele patru categorii.

Au rezolvat corect...

La această rundă au participat 55 de concurenți dintre care 24 au obținut punctajul maxim:

Chattopadhyay Amit, India
Mugurel-Ionuț Andreica, România
Andrei Blaj, România
Adrian Cărcu, România
Cristian Cernat, România
Liviu Ciortea, România
Adrian Diaconu, România
Ioan-Dan Docleanu, România
Marius Dumitran, România
Silviu-Ionuț Gânceanu, România
Ștefan Gheorghe, România
Kazmenko Ivan, Federația Rusă
Vasanth Jai, India
Haryanto Lego, Statele Unite ale Americii
Adina Pampu, România
Mircea Pașoi, România
Daniel Păsăilă, România
Csaba Pătcăș, România
Alexandru Peptan, România
Mihail-Cosmin Piț-Rada, România
Tudor-Alexandru Rabaea, România
Renald, Indonezia
Cristian-George Strat, România

P020431: Era imperiilor

Pentru ca aria poligonului să fie maximă, punctele care îl determină trebuie să se afle pe același cerc.

Vom determina raza acestui cerc folosind *metoda căutării binare*. Pentru fiecare rază pe care o verificăm, vom încerca să amplasăm punctele pe cerc. Dacă reușim să închidem poligonul, am găsit raza corectă. Dacă nu, vom verifica dacă ultimul punct a "trecut" (în sensul considerat) de primul punct. Dacă ultimul punct trece de primul punct, atunci raza căutată este mai mare decât cea aleasă, iar dacă nu, raza căutată este mai mică.

După determinarea razei vom calcula aria poligonului folosind o simplă triangularizare (coordonatele sunt determinate deja de la pasul căutării binare).

Mai trebuie tratat cazul în care nu se poate forma nici un poligon. În această situație, aria care poate fi împrejmuită este, evident egală cu 0.

Au rezolvat corect...

La această etapă au participat 27 de concurenți, dintre care numai cinci au obținut punctajul maxim:

Liviu Ciorte, România
Marius Dumitran, România
Silviu-Ionuț Gânceanu, România
Daniel Păsăilă, România
Tudor-Alexandru Rabaea, România

P020432: Aladin

Se observă foarte ușor că orice configurație binară a primei linii identifică unic întregul covor cu două excepții:

- configurații de forma: 0 1 0 1 0 1 ...
- configurații de forma: 1 0 1 0 1 0 ...

Pentru aceste două tipuri de configurații, pentru fiecare linie avem două opțiuni (putem alege oricare dintre cele două configurații pe linia respectivă).

Ca urmare avem $2^m - 2$ configurații pe prima linie care identifică în mod unic întregul covor și alte 2^n configurații care pot fi obținute folosind pe prima linie una dintre cele două configurații amintite. În concluzie, numărul total de configurații posibile este $2^m + 2^n - 2$.

Au rezolvat corect...

Dintre cei 47 de participanți la această etapă, 28 au reușit să obțină numărul maxim de puncte.

Chattopadhyay Amit, India
Mugurel-Ionuț Andreica, România
Alexandru Anton, România
Eduard-Gabriel Băzăvan, România
Oana Belei, România
Andrei Blaj, România
Adrian Cârțu, România
Liviu Ciorte, România
Andrei Costăș, România
Adrian Diaconu, România
Marius Dumitran, România
Silviu-Ionuț Ganceanu, România
Kazmenko Ivan, Federația Rusă
Vasanth Jai, India
Slotta Johannes, Germania
Haryanto Lego, Statele Unite ale Americii
Dalian Valeriu Moraru, România
Marius Nicolae, România
Mircea Pașoi, România
Ciprian Păduraru, România
Daniel Păsăilă, România
Alexandru Peptan, România
Mihail-Cosmin Piț-Rada, România
Victor Podeanu, România
Calin Popa, România
Claudiu Rad, România
Renald, Indonezia
Adrian Vladu, România

P020433: Santinele

Problema se reduce la determinarea poziției unui cerc de rază dată care conține în interiorul său cât mai multe dintre punctele date.

Pentru a rezolva problema vom lua în considerare fiecare punct în care se află o santinelă și vom presupune că punctul respectiv se află pe circumferința cercului de rază r căutat.

În aceste condiții problema se reduce la a rezolva următoarea subproblemă, pentru fiecare punct.

Se cunosc coordonatele a n puncte. Să se determine un cerc care conține pe circumferința sa unul dintre punctele date, are raza r și conține în interiorul său cât mai multe dintre punctele date.

Pentru început vom sorta celelalte $n - 1$ puncte în funcție de valorile unghiurilor determinate de dreapta care unește cele două puncte și dreapta orizontală care trece prin punctul ales.

În continuare vom considera un cerc oarecare de rază r care trece prin punctul ales și numărăm punctele aflate în interiorul acestui cerc (inclusiv cele aflate pe circumferința cercului).

Cunoscând ordinea punctelor (stabilită în urma sortării anterioare) putem mișca centrul cercului determinat în sens trigonometric în jurul punctului ales, astfel încât la fiecare rotație numărul punctelor din interiorul cercului să crească sau să scadă cu 1.

După efectuarea celor $n - 1$ rotiri vom ști care este numărul maxim de puncte care se află în interiorul unui cerc a cărui rază este r și pe a cărui circumferință se află un punct dat.

Aplicând același procedeu pentru toate punctele, vom putea determina numărul maxim de puncte care se află în interiorul unui cerc de rază r .

Se observă că pentru fiecare punct efectuăm o sortare al cărei ordin de complexitate este $O(n \cdot \log n)$ și un număr de $O(n)$ rotiri, ordinul de complexitate al fiecărei operații de rotire fiind $O(1)$.

Ca urmare, ordinul de complexitate al operațiilor efectuate pentru fiecare punct ales este egal cu $O(n \cdot \log n) + O(n) = O(n \cdot \log n)$.

Datorită faptului că avem în total n puncte, ordinul de complexitate al acestui algoritm de determinare a numărului maxim de puncte care se află în interiorul unui cerc a cărui rază este cunoscută este: $O(n) \cdot O(n \cdot \log n) = O(n^2 \cdot \log n)$.

A rezolvat corect...

Aceasta s-a dovedit a fi cea mai dificilă problemă propusă în prima jumătate a primei faze a concursului nostru. Un singur concurent dintre cei 34 care au trimis rezolvări a reușit să obțină punctajul maxim:

Adrian Cârțu, România

