Configurații stabile
(se termină sau nu se termină?)

prof. Moț Nistor Eugen

Motto 1: Tot ce-a fost ori o să fie
În prezent le-avem pe toate
(Glossa)
Motto 2: Asta la vista, baby!
(Terminator 2)

Discutăm în continuare despre mulțimi de elemente care suportă anumite transformări. Ne interesează dacă după un anumit număr de transformări se ajunge la o situație finală dată. Fie $S=\{S_1,S_2,\ldots S_t\}$ mulțimea stărilor posibile. Dacă există o singură transformare $f:S\to S$, (deci starea următoare depinde doar de starea actuală) se știe că există n_0 și $p\in \mathbf{N}$, astfel ca $f^{(n+p)}=f^{(n)}$ pentru $n>n_0$, dacă avem o familie de transformări $f_1,f_2,..f_s$ $:S\to S$, e valabil un rezultat asemănător, $f_i^{(n+pi)}=f^{(p)}$ (am notat cu $f^{(k)}$ compunerea lui f cu el însuşi de k ori). Dacă starea următoare depinde de evoluția sistemului, deci de toate stările precedente, să zicem $f: S\times S\times \ldots\times S\to S$, lucrurile sunt mai complicate, (sau foarte complicate).

Mai concret vom rezolva probleme de genul:

- Avem 3 grămezi cu nuci. Luăm `2k` nuci dintr-o grămadă şi le punem în cealalte grămezi, `n` fiecare câte `k` nuci. Facem operația atât timp cât există o grămadă mai mare ca alta. Ajungem la trei grămezi egale sau nu se termină niciodată mutatul nucilor?

- Într-o matrice cu `1` şi `0` schimbăm pe `1` în `0` şi invers până când ajungem la o matrice fără `0`, se termină vreodată schimbările? (bineînțeles, răspunsul depinde de regulile adoptate)

- Avem `n` lămpi, unele aprinse altele stinse aşezate în cerc. La fiecare secundă lampa `i` îşi schimbă starea dacă lampa `i+1` e aprinsă (lampa `n+1` fiind lampa `1`). Pentru o configurație dată care va fi situația după `T` secunde?

- Se termină vreodată plimbatul pe la ghişee? (vezi problema Ghişee)

- Tranziția din România se termină sau nu?

- Avem un pachet cu `n` cărți de joc pe care le amestecăm după o regulă de genul: pac, pac, pac.

I. Două probleme binecunoscute

1. **Jocul vieții** - de pe lista lui Frâncu (reproduc integral enunțul şi rezolvarea)

O colonie este formata din N celule asezate pe un cerc; unele celule sunt vii si altele moarte. Definim evolutia cu o generatie a coloniei ca fiind procesul prin care toate celulele isi schimba simultan starea conform urmatoarelor reguli:

1) Intre doua celule moarte nu poate exista o celula vie.

2) Intre doua celule vii, orice celula vie se sufoca si moare.

3) Intre o celula moarta si una vie, o celula supravietuieste (sau daca era moarta, una noua se naste in loc).

Lămurire: Starea unei celule din generatia cu numarul k depinde exclusiv de starile celulelor vecine ei din generatia k-1.

De exemplu, codificand cu 0 o celula moarta si cu 1 o celula vie, iata cum evolueaza colonia 1001011 (N=7). Subliniem ca prima celula este vecina cu ultima, colonia avand forma circulara.

```
1001011 ==> 1110010 ==> 1011100 ==> 0010111 ==> ...
 Gen. 0      Gen. 1      Gen. 2      Gen. 3
```

Spunem ca o colonie se stabilizeaza daca exista o generatie S>=0 incepand de la care nici o celula nu isi mai schimba starea, adica aspectul coloniei este acelasi la generatiile S, S+1, S+2, ... Dandu-se o colonie cu N celule, sa se spuna daca ea se stabilizeaza si daca da, care este valoarea minima pentru S (valoarea minima in sensul ca generatia S-1 este diferita de generatia S).

DATE DE INTRARE

Fisierul COLONIE.IN contine pe prima linie numarul de celule, N<=100. Pe a doua linie fisierul contine o secventa de N caractere 0 (reprezentand celule moarte) si 1 (reprezentand celule vii), corespunzatoare generatiei 0. Caracterele NU sunt despartite prin spatii.

DATE DE IESIRE

Fisierul COLONIE.OUT contine pe prima linie mesajul DA sau NU, dupa cum colonia se stabilizeaza sau nu. Daca raspunsul este DA, pe linia a doua se va tipari valoarea lui S.

EXEMPLE

```
COLONIE.IN              COLONIE.OUT
4                       DA              // Pentru ca 1110 ==> 1010 ==>
1110                    2               // ==> 0000 ==> 0000 ==> ...
COLONIE.IN              COLONIE.OUT
5                       NU
11110
```

COMPLEXITATE RECOMANDATA: O(N^3).

Rezolvare

Problema 7 (Jocul vietii) a fost intr-o anumita masura un esec, mai ales pentru cei care n-au trimis nici o sursa. Imi pare foarte rau si promit ca pe viitor sa postez numai probleme cu teste "urate" gata facute. Care a fost buba: aceasta problema nu are teste "urate", in sensul ca orice configuratie cu N<=100 celule, ori se stabilizeaza in maximum 16 iteratii, ori nu se stabilizeaza.

Totusi, iata algoritmul "destept", care are complexitate O(N^2) nu O(N^3), lucru pe care l-am descoperit cu stupoare scriind programul.

Mai mult, raspunsul la intrebarea daca o colonie se stabilizeaza se poate da in O(N).

Ideea de pornire este ca exista 2^N configuratii distincte de N celule. Din acest motiv, conform principiului lui Dirichlet, este evident ca dupa cel mult 2^N generatii configuratiile incep sa se repete. Deci tot ce avem de facut este sa comparam generatia 2^N cu generatia 2^N+1 si aflam daca (,) colonia se stabilizeaza. Cum facem sa evoluam 2^N generatii ? Desigur, nu iterativ, ci evoluand in salturi. Pentru aceasta, sa facem niste notatii: fie C[0]..C[N-1] colonia originala si fie D[t][i] starea celulei i la momentul t.

Postulam ca:

D[2^k][i]=C[i-2^k] xor C[i+2^k] oricare k>=0

(aici indicii in C se iau modulo N).

Cu alte cuvinte, starea unei celule la momentul 1 este data de starea originala a vecinilor ei; starea la momentul 8 este data de starea originala a vecinilor ei de la 8 celule departare s.a.m.d.

Demonstratia se face prin inductie completa, desi dupa cum stim programatorii se multumesc adesea cu inductia incompleta :) Deci:

1) Pentru k=0 propozitia este adevarata:

D[2^0][i]=C[i-2^0] xor C[i+2^0], adica D[1][i]=C[i-1] xor C[i+1],

ceea ce este adevarat.

2) Presupunem ca propozitia este adevarata pentru k si demonstram ca este adevarata si pentru k+1. Pentru aceasta, remarcam ca generatia de la momentul 2^(k+1) se obtine plecand din configuratia de la momentul 2^k si evoluand cu inca 2^k generatii:

```
C ----------------> D[2^(k+1)]
 \                 /
   --> D[2^k] -->
```

Dar pentru 2^k iteratii afirmatia se verifica si putem scrie:

D[2^(k+1)][i]=D[2^k][i-2^k] xor D[2^k][i+2^k]

D[2^k][i-2^k]=C[i-2^k-2^k] xor C[i-2^k+2^k] = C[i-2^(k+1)] xor C[i]

D[2^k][i+2^k]=C[i+2^k-2^k] xor C[i+2^k+2^k] = C[i] xor C[i+2^(k+1)]

Din cele trei rezulta ca

D[2^(k+1)][i] = C[i-2^(k+1)] xor C[i] xor C[i] xor C[i+2^(k+1)]

= C[i-2^(k+1)] xor C[i+2^(k+1)] q.e.d.

Asadar pentru a calcula ceea ce ne intereseaza, si anume D[2^N],

scriem ca D[2^N][i]=C[i-2^N] xor C[i+2^N]. Nu va speriati ca 2^N are

vreo 30 de cifre cand N=100, pentru ca noua ne trebuie indicii (i-2^N)

modulo N si (i+2^N) modulo N, deci nu este nevoie de numere mari

(deocamdata).

D[2^N] se poate calcula in O(N) si astfel decidem daca (iar) colonia se stabilizeaza. Presupunem ca se stabilizeaza - cum decidem in cat timp ?

Sa notam D[2^N] = S, configuratia in care incremeneste colonia. Acum sa analizam sirul D[0]=C, D[1], D[2], ..., D[2^N].

Acest sir contine niste elemente diferite de S, apoi un sir de elemente egale cu S, deoarece din clipa in care apare prima generatie egala cu S, toate cele care ii urmeaza sunt egale cu S. Trebuie sa aflam indicele primei generatii egale cu S. Aceasta se face prin cautare binara. Intr-o prima faza, calculam D[2^(N-1)] si avem doua variante:

- Daca D[2^(N-1)]=S, atunci colonia se stabilizeaza intr-un moment anterior lui 2^(N-1);

- Daca D[2^(N-1)]<>S, atunci colonia se stabilizeaza intr-un moment ulterior lui 2^(N-1);

Si asa mai departe. Teoretic, complexitatea se calculeaza astfel:

Numarul generatiei in care colonia se stabilizeaza este cuprins intre 0 si 2^N-1, deci are O(N) cifre. Adunarile pe asemenea numere se fac in O(N).

Cautarea prin injumatatirea intervalului efectueaza log 2^N = N pasi.

Fiecare pas presupune o evolutie cu 2^tz generatii, care se face in O(N), si eventual o adunare pe numere mari, deci inca O(N). Per total O(N)*(O(N)+O(N))=O(N^2).

Cu "mica" observatie facuta la inceputul acestui mesaj, anume ca o configuratie cu 100 celule se stabilizeaza in cel mult 16 iteratii, putem reduce complexitatea la O(N). Dar asta-i alta poveste...

Variante ale acestei probleme găsiți în multe locuri (s-a dat şi la concursul organizat de GInfo).

Din aceeaşi sursă (sau altele, indicate la bibliografie) citiți problema Ghişee – veți afla dacă peregrinările noastre pe la ghişee se termină sau nu, ele pot fi uşor contabilizate cu un contor Tarjan sau contor Cozmâncă sau altele similare.

2. Celule

Într-un lanț infinit de celule sensibile fiecare celulă se poate găsi în două stări: "liniştită" şi "excitată". Dacă la un anumit moment o celulă a fost excitată, atunci ea emite un semnal care după un timp scurt (o milisecundă) ajunge la fiecare dintre cele două celule vecine cu ea. Fiecare celulă este excitată atunci şi numai atunci când la ea ajunge un semnal de la una dintre celulele vecine; dacă semnalele ajung deodată din amândouă părțile, atunci ele se anulează şi celulele nu se excită. Să presupunem că în momentul inițial este excitată numai una din celule. Câte celule se vor găsi în stare excitată după t milisecunde?

Rezolvare (după "Probleme de matematică traduse din revista sovietică Kvant")

Este suficient să urmărim cum se propagă excitația de la o singură celulă în primii 14-15

timpi pentru a observa următoarele reguli (numerotăm celulele : …–3, –2, –1, 0, 1, 2, 3, …, celula excitată inițial fiind celula 0):

- la momentul $t=2^k$, k=0,1,2,.. sunt excitate numai două celule: -2^k şi 2^k
- la momentul $t=2^k-1$ sunt excitate 2^k celule, de la -2^k+1 până la 2^k-1
- fie $0\le t\le 2^k$; atunci la momentul 2^k+t sunt excitate de două ori mai multe celule decât la momentul t (demonstrația – prin inducție matematică sau informatică – se observă că "undele de excitație" generate de cele două celule excitate la momentul 2^k nu se acoperă până la momentul $2^{k+1}-1$ şi de aceea fiecare undă este construită ca unda de excitație de la o singură celulă pentru $t<2^k$)

De exemplu, notând cu f(t) numărul celulelor excitate la momentul t, f(1000)=2f(488)= 4f(232)=8f(104)=16f(40)=32f(8)=64.

Folosind scrierea în baza 2 se poate spune mai simplu $f(t)=2^m$ unde m este numărul de cifre 1 din scrierea binară a lui t.

II. Câteva probleme de acelaşi gen, comentate

1. Sume pozitive

Se consideră o matrice de dimensiuni m×n cu elemente întregi. Singura operație permisă este schimbarea semnelor tuturor elementelor unei linii sau coloane cu –1. Găsiți numărul minim de operații care transformă matricea într-una în care suma elementelor de pe orice linie sau coloană este nenegativă, sau precizați că nu e posibilă obținerea unei astfel de matrice.

Rezolvare

Întotdeauna e posibilă obținerea unei matrice cu sume pozitive! Teoretic rezolvarea este simplă: căutăm o linie sau coloană cu suma negativă şi o înmulțim cu –1. Astfel, suma tuturor elementelor matricei creşte, iar matricele care se obțin sunt diferite între ele. Dar există numai 2^{mn} matrice posibile, deci nu putem face o infinitate de operații, ceea ce înseamnă că vom ajunge la o matrice care nu mai are sume negative. Problemele puse de implementare sunt: cum căutăm sumele negative, cum actualizăm sumele, cât de multe operații trebuie făcute şi mai ales cum găsim numărul minim de operații. Pentru moment dăm răspunsul la una din întrebări - câte operații sunt necesare: din rezolvarea teoretică a rezultat că pot fi cel mult 2^{mn} operații, dar cum avem doar m linii şi n coloane şi pe fiecare linie/coloană e suficient să facem o singură dată schimbarea de semne, numărul se reduce simțitor, la m+n. Se poate arăta însă că sunt suficiente (m+n)/2 operații

2. Rețea de puncte colorate

Se consideră mulțimea punctelor din plan de coordonate întregi, pe care le considerăm colorate în alb. Un număr de n puncte se colorează în roşu, după care, la fiecare unitate de timp culoarea tuturor punctelor din plan se schimbă simultan, după regulile :

- dacă un punct are majoritatea vecinilor coloraţi în roşu, punctul va deveni roşu
- dacă un punct are majoritatea vecinilor coloraţi în alb, punctul va deveni alb
- dacă un punct are acelaşi număr de vecini albi şi roşii îşi păstrează culoarea

După câte unități de timp dispar punctele roşii (dacă dispar)?

Prin vecini ai punctului (a,b) înțelegem punctele (x,y) pentru care |a–x|+|b–y|=1 (sau numai |b–y|=1, cum s-a dat la OIM)

Date de intrare

Fişierul `puncte.in` conține pe prima linie numărul n, pe următoarele n linii perechi de numere întregi separate de câte un spațiu, coordonatele punctelor roşii

Date de ieşire

În fişierul `puncte.out` se va scrie numărul 0 dacă nu dispar niciodată punctele roşii sau un număr natural nenul reprezentând cel mai mic număr de unități de timp după care nu vor mai exista puncte roşii.

Restricții

$1 \le n \le 1000$

Coordonatele punctelor sunt din intervalul `[-32000, 32000]`

Exemplu

```
rosii.in          rosii.out
3                    2
1 1
1 2
2 1
```

Rezolvare

În funcție de modul în care înțelegem vecinătatea punctele roşii dispar sau se pot menține la infinit (pentru anumite configurații). Problema originală,dată la o olimpiadă rusească de matematică lua în considerație doar vecinii de sus şi de jos, şi are o demonstrație că n puncte roşii dispar după n unități de timp (prin inducție matematică, cam neclară). Pentru alte cazuri problema rămâne drept studiu, ideea de bază este că nu se pot extinde punctele roşii în afara dreptunghiului în care se găsesc inițial.

3. Scatii zglobii

Pe n copaci aşezaţi în cerc sunt m scatii zglobii. La fiecare minut câte doi scatii zboară de pe copacul în care stau în copacul alăturat, în direcţii contrare (unul în sens orar, celălalt în sens antiorar). E posibil ca după un anumit timp scatiii să ajungă în anumite poziţii precizate? Dacă da, găsiţi o variantă de a realiza acest lucru (eventual numărul minim de zboruri necesare).

Exemplul 1: n=4, m=4, configuraţia iniţială (2,0,1,1) configuraţia finală (0,3,1,0).

Se poate ajunge din două "mutări"; prima: 1=>2 şi 4=>3 a doua: 1=>2 şi 3=>2

Exemplul 2: n=4, m=4, configuraţia iniţială (2,0,1,1), configuraţia finală (0,3,0,1). Nu se poate.

Exemplul 3 (exemplul din enunţul original al problemei): n=44, m=44, configuraţia iniţială (1,1,1,…,1) – deci câte un scatiu în fiecare pom, configuraţia finală (0,0,…44,…,0) – deci toţi scatiii adunaţi într-un pom. Nu se poate! (dar dacă n=m=43 se poate!).

Rezolvare

Nu prea are legătură cu tema iniţială (configuraţii stabile), dar e frumuşică. Se poate arăta că două configuraţii sunt echivalente (adică se poate ajunge de la una la cealaltă) dacă au loc următoarele relaţii: $(a_0, a_1,\ldots a_{n-1})$ echivalent cu $(b_0, b_1, \ldots, b_{n-1})$ dacă $a_0+a_1+\ldots+a_{n-1}=b_0+\ldots b_{n-1}$ (ceea ce e evident) şi $(a_1+2a_2+\ldots+(n-1)a_{n-1}) \bmod n = (b_1+2b_2+\ldots+(n-1)b_{n-1}) \bmod n$.

O modalitate de a trece de la o configuraţie la alta echivalentă, cu max. m·n mutări, se poate obţine dacă arătăm cum putem strânge toţi scatiii pe pomul 0, mai puţin unul care va fi pe pomul $r=(a_1+2a_2+\ldots+(n-1)a_{n-1}) \bmod n$: alegem un scatiu care va fi pereche pentru toţi ceilalţi care se deplasează spre pomul 0 (justificarea faptului că scatiul "plimbăreţ" ajunge pe pomul r e simplă: se arată că mutând doi scatii după regulile date valoarea expresiei $\sum_{k=0}^{n-1} k \cdot a_k \bmod n$ nu se modifică). Efectuând zborurile din pomul 0 în sens invers, putem duce scatiii în poziţiile dorite. Rămâne de discutat chestiunea numărului minim de operaţii.

4. Paritate

Se consideră n numere întregi, aşezate pe un cerc, toate având aceeaşi paritate. La fiecare unitate de timp se înlocuieşte fiecare număr cu media aritmetică dintre el şi cel de după el (în sens orar). După câte unităţi de timp nu mai avem aceeaşi paritate la toate numerele ?

Date de intrare : din prima linie a fişierului paritate.in se citeşte numărul n iar de pe liniile următoare se citesc n numere întregi având aceeaşi paritate.

Date de ieşire : în fişierul paritate.out se va scrie numărul 0 dacă numerele vor avea aceeaşi paritate la infinit sau un număr natural nenul reprezentând cel mai mic număr de unităţi de timp după care numerele din şir nu vor avea toate aceeaşi paritate.

Restricţii : $2 \le n \le 5000$

numerele sunt întregi din intervalul [-32000, 32000]

Exemplu

paritate.in	paritate.out
4	0
7	
9	
7	
9	

Rezolvare

Mai întâi să examinăm în ce caz poate continua la infinit operaţia respectivă. Evident, când toate numerele devin egale. Altă posibilitate există? Nu, pentru că media dintre numărul maxim şi cel de după el ne dă un număr mai mic dacă numerele nu sunt egale. Câte operaţii se fac până devin toate egale? Cel mult una! Acum celălalt caz. Ştim că $\min\{x_i, x_{i+1}\} \le (x_i+x_{i+1})/2 \le \max\{x_i,x_{i+1}\}$, deci şi $\min\{x_1,x_2,\ldots,x_n\} \le (x_i+x_{i+1})/2 \le \max\{x_1,x_2,\ldots,x_n\}$. Prin urmare diferenţa dintre cel mai mare şi cel mai mic număr de pe cerc poate cel mult să rămână aceeaşi sau să scadă. Dacă cunoaştem diferenţa iniţială dintre cel mai mare şi cel mai mic număr ştim numărul maxim de operaţii care pot fi făcute până obţinem două numere de paritãţi diferite sau până toate numerele devin egale. E nevoie de atâtea operaţii ? Pentru enunţul dat nu are importanţă, dar dacă schimbăm puţin restricţiile şi luăm numere din intervalul $[-10^9, 10^9]$?

III. Dacă v-aţi prins

1. Ordine

Considerăm n persoane de înălţimi diferite $h_1,h_2,\ldots,h_n$ formând un cerc. Numim antiordine între 4 persoane consecutive de pe cerc, să le numim a,b,c,d, situaţia în care persoanele din interior (b şi c) au înălţimile în ordine inversă faţă de persoanele din margine (a şi d), adică $h_a<h_d$ şi $h_b>h_c$ sau invers ($h_a>h_d$ şi $h_c<h_d$), h_i fiind desigur înălţimea persoanei i. Odată la fiecare secundă un grup de 4 persoane consecutive aflate în antiordine procedează la inversarea poziţiei persoanelor din mijloc. Se termină sau nu inversările? Dacă da, după câte secunde?

2. Ghişee

În ţara Iaroman birocraţia este foarte bine organizată. Dimineaţa se prezintă la ghişee, numerotate cu numere naturale de la 1 la n, n solicitanţi, câte unul la fiecare ghişeu. Instrucţiunile prevăd clar ce trebuie să facă funcţionarul de la ghişeul i cu un eventual solicitant: să-l trimită la ghişeul j ($1\le i,j\le n$). Considerăm că deplasările de la un ghişeu la altul se fac într-o unitate de timp. În fişierul ghisee.in se află numărul n pe prima linie iar pe următoarele n linii un şir de n numere naturale cuprinse între 1 şi n (numărul de pe linia i+1 arată unde va fi trimisă o persoană care se prezintă la ghişeul i). Scrieţi în fişierul ghisee.out două numere naturale p,q ($p<q$) reprezentând cele mai mici valori cu proprietatea că după p, respectiv q unităţi de timp fiecare din cele n persoane se află la aceleaşi ghişee (mai exact pozitia persoanei i la timpul p=pozitia persoanei i la timpul q, pentrui=1,2,...,n).

Restricţii: $2\le n\le 5000$, numărul de persoane care se prezintă la un ghişeu e nelimitat.

Exemplu: n=5 şi 3 1 4 3 2 => răspunsul 3 5

3. Uniformizare

Se scriu pe un cerc N cifre de 0 şi 1. Şirul de cifre se transformă astfel: între două cifre identice se scrie 0, între două cifre diferite se scrie 1, apoi se şterg numerele iniţiale. Cu cele N rămase se procedează la fel. Se poate obţine un şir format din N cifre 0?

Rezolvările la aceste 3 probleme sunt uşor de găsit dacă le-aţi studiat pe cele anterioare, aşa că nu mai e nevoie de nici un comentariu.

În concluzie, dacă întâlniți o astfel de problemă încercați una din ideile de mai sus:

- căutați un invariant al sistemului la transformările date
- căutați o margine superioară a numărului de transformări
- operațiile pe biți sunt utile
- efectuați transformări în salturi
- folosiți căutarea binară

Sper că explicațiile date la fiecare problemă sunt suficient de clare ca să suplinească lipsa algoritmilor, eventual puteți vedea sursele din anexă.

IV. Probleme deschise

1. Avem o grămadă de mărgele. Dacă numărul de mărgele e par aruncăm jumătate din ele, dacă e impar, mai adăugăm de două ori numărul de mărgele plus una. Repetăm operația până când rămâne o singură mărgea. Se termină sau nu jocul cu mărgele ?

2. Se consideră o matrice cu valori întregi. Se efectuează de mai multe ori operația următoare: se adună la un element şi la vecinii lui un număr oarecare. Se poate obține o matrice cu toate elementele egale?

3. Tranziția din România (spre ce?) cât mai ține ???

V. Bibliografie

1. I. Cuculescu – Olimpiadele de matematică ale elevilor, Ed. Tehnică Bucureşti 1984
2. H. Banea – Probleme de matematică traduse din revista sovietică Kvant, Ed. Didactică şi Pedagogică Bucureşti 1983
3. T. Cormen, C. Leiserson, R. Rivest – Introducere în algoritmi, Ed. Agora Cluj 2000
4. M.Oltean - Programarea jocurilor matematice, Ed Albastră Cluj 1996
5. D. Knuth – Tratat de programare a calculatoarelor. Algoritmi fundamentali, Ed. Tehnică Bucureşti 1975
6. GInfo 1998 - 2002