

# Jocuri combinatoriale

## Definiția 1

Prin **joc** se înțelege un șir de decizii (acțiuni, mutări), luate de părți ale căror interese sunt contrare.

Jocurile pot fi clasificate după numărul de jucători, după natura mutărilor (libere sau aleatoare, cum ar fi zaruri, cărți de joc, etc), după cantitatea de informație disponibilă (cunoaștem sau nu informații despre toate mutările precedente?).

Jocurile despre care discutăm în continuare sunt jocuri pentru 2 jucători, care mută alternativ, mutările sunt libere (nu există nimic aleator) și informația este completă (cunoaștem toate mutările efectuate).

## Definiția 2

Spunem că un jucător are **strategie sigură de câștig** dacă acesta va câștiga, indiferent de modul în care ar încerca adversarul să îi împiedice victoria.

## Exemple de strategii de joc

### Strategia simetriei

Un jucător imită mutările celuilalt jucător, efectuând o mutare simetrică cu a adversarului (simetria se stabilește în funcție de o axă de simetrie sau de o axă de simetrie)

### Exemplul 1

O bandă de hârtie este împărțită în  $n$  căsuțe. Alternativ, doi jucători X și Y hașurează câte  $k$  căsuțe adiacente, nehașurate. Cel care nu mai poate muta pierde. Inițial mută X. Se știe că  $n$  și  $k$  au aceeași paritate. Se cere:

- Să se determine dacă jucătorul X are strategie sigură de câștig.
- În caz afirmativ se cere să se programeze mutările lui X, cele ale lui Y fiind citite de la intrare.

### Soluție

Inițial X va hașura căsuțele din centrul benzii, împărțind în acest fel banda de hârtie în două benzi de dimensiuni egale, putând în acest fel să imite simetric mutările lui Y.

### Exemplul 2

Fie o tablă dreptunghiulară împărțită în  $n \times m$  căsuțe. Alternativ, doi jucători X și Y hașurează câte două căsuțe adiacente, nehașurate până în acel moment. Cel care nu mai poate muta pierde. Inițial mută jucătorul Y. Cel puțin unul dintre numerele  $m, n$  este un număr par. Se cere:

- Să se determine dacă jucătorul X are strategie sigură de câștig.
- În caz afirmativ se cere să se programeze mutările lui X, cele ale lui Y fiind citite de la tastatură.

### *Soluție*

Vom căuta o axă de simetrie (pentru că cel puțin unul dintre  $m$  și  $n$  este par). Să alegem ca axa de simetrie linia care este paralelă cu marginea tablei care are un număr impar de pătrățele (dacă aceasta există) și este egal distanțată de cele două margini. Dacă Y joacă de o parte a axei de simetrie, X va juca în cealaltă parte, dar dacă Y va hașura două pătrățele, unul de o parte a axei, iar celălalt de cealaltă parte, atunci X nu mai are cum să îl imite simetric, de aceea este evidentă necesitatea unei noi axe de simetrie, lucru care este posibil, doar dacă ambele numere  $n, m$  sunt pare. Dacă doar unul din numerele  $n, m$  este par, atunci Y, va putea hașura primele două patratele pe linia (coloana) din centru, având o patratică de o parte a axei, iar cealalta patratică de cealalta parte a axei, iar apoi va imita simetric mutarile lui X.

În concluzie, X are strategie sigură de câștig, dacă ambele numere  $n, m$  sunt pare, iar o poziție  $(p, q)$ , va avea simetrica de coordonate  $(n+1-p, m+1-q)$ .

### **Strategia perechilor**

Ideea strategiei constă în gruparea mutărilor pe perechi de mutări. La fiecare pas, unul dintre jucători va putea efectua mutarea pereche a mutării efectuate de adversar.

Ca și în strategia simetriei, este posibil ca prima mutare să nu poată fi împerecheată, cu o altă mutare, dar apoi (după ce această mutare este efectuată) să fie posibilă realizarea acestui lucru.

### **Exemplul 1**

Se dă o foaie de hârtie dreptunghiulară, împărțită în  $n \times m$  căsuțe. Alternativ, doi jucători X și Y decupează una dintre căsuțele care este învecinată pe orizontală, sau verticală cu ultima căsuță decupată. Cel care nu mai poate muta pierde. Inițial mută X și va decupa o căsuță oarecare. O căsuță poate fi decupată o singură dată, deci nu este permisă parcurgerea de două ori a aceleiași căsuțe. Se cere:

- Să se determine dacă X are strategie sigură de câștig.
- În caz afirmativ să se programeze mutarile lui X, cele ale lui Y fiind citite de la intrare.

### *Solutie*

Ideea de bază este gruparea mutărilor în perechi. În acest caz încercăm să acoperim tabla cu piese de domino. Dacă tabla are un număr impar de căsuțe, atunci nu este posibilă o acoperire completa a acesteia, dar în schimb o putem acoperi, dacă eliminăm, spre exemplu colțul din stanga sus, deoarece pe linia 1, vom rămâne cu un număr par de casute, pe care le putem acoperi complet, fără să suprapunem piese, iar în rest mai rămâne o suprafață dreptunghiulară care are un număr par de casute, care pot fi acoperite cu dominouri.

Dacă tabla are un număr par de casute, atunci cel de-al doilea jucător (Y) are strategie sigură de câștig, deoarece la fiecare pas, va juca în acea căsuță, care se afla sub același domino cu casuta decupată de X. Dacă tabla are un număr impar de casute, atunci jucătorul care mută primul (X) are strategie sigură de câștig, inițial el decupând casuta care nu a fost acoperită de nici un Domino, apoi la fiecare pas decupând casuta pereche casutei decupate de Y.

### **Exemplul 2**

Se da o foaie de hârtie dreptunghiulară, împărțită în  $n \times n$  căsuțe. Alternativ, doi jucători X și Y decupează una din casutele care este L-învecinată cu ultima casuta decupată. Două casute se numesc L-învecinate, dacă există o săritură a calului care să treacă de la una la alta (dintr-o singură mutare). Cel care nu mai poate muta pierde. O casuta poate fi decupată o singura data, deci nu este permisă parcurgerea de doua ori a aceleiași căsuțe. Se cere:

- a) Sa se determine dacă X are strategie sigură de castig.
- b) In caz afirmativ sa se programeze mutarile lui X, cele ale lui Y fiind citite de la intrare.

### *Solutie*

O mutare este identică cu o săritură a calului, într-o poziție nouă, neparcursă. Trebuie să găsim o împărțire a casutelor tablei în perechi, astfel încât fiecare casuta să facă parte dintr-o singură pereche. Să definim o pereche de casute, ca fiind două casute astfel încât un cal poate să sara direct de pe una pe alta. Deci dacă reușim să găsim un drum al calului, care să treacă o singura data prin fiecare casuta, atunci problema este rezolvată, deoarece casuta cu număr de ordine impar în cadrul drumului, va fi prima casuta a Domino-ului în forma de L, definit anterior, iar casutele cu număr de ordine par vor fi cea de-a doua casuta a Domino-ului. Pentru  $n$  par, jucătorul Y are strategie sigură de castig, la fiecare pas el jucând în casuta pereche celei în care a jucat X.

Pentru  $n$  impar, jucătorul X are strategie sigură de castig, inițial el decupând poziția din centru, apoi la fiecare pas va juca mutarea pereche a mutării executate de Y.

Determinarea unui drum al calului care să treacă o singura data prin fiecare casuta, se face poate prin următorul procedeu:

- Se pleacă din poziția (1,1). Se poate pleca la fel de bine din orice alta poziție.
- La fiecare pas se sare în acea casuta din care sunt cele mai puține posibilități de a sari, mai puțin în casuta din centru.

## 2.3 Strategia parității

Denumirea acestei strategii vine de la existența a două tipuri de configurații (pară, respectiv impară) în care se poate afla la un moment dat jocul (partida). Ele sunt definite, în avantajul unuia dintre jucători (să-l numim pe acesta X, iar pe celălalt Y), astfel la fiecare pas, jucatorul X, va putea face o mutare convenabilă care să transforme o configurație de joc impară într-o configurație de joc pară. Spre deosebire de X, jucatorul Y, va putea să treacă la fiecare mutare, doar dintr-o configurație pară într-una impară. Starea finală a jocului este o configurație pară.

Mai exact, jocul îndeplinește (printre altele) următoarele condiții:

1. Se joacă între doi parteneri, care muta alternativ. Cel care nu mai poate muta pierde.
2. Putem defini o partiție cu două clase (A și B) a mulțimii configurațiilor jocului, astfel încât:

- a) Configurația finală să aparțină mulțimii A.
- b) Orice mutare aplicată unei configurații din A, va transforma această configurație într-o configurație din B. Pentru configurația finală nu există nici o mutare care să o transforme într-o altă configurație.
- c) Există o mutare convenabilă, care realizează transformarea unei configurații din B într-o configurație din A.

În aceste condiții va câștiga (dacă joacă fără greșeli) acel jucător care la momentul în care este la mutare va găsi jocul într-o configurație impară. Pentru a câștiga el trebuie să treacă într-o configurație pară, lucru posibil, datorită modului cum am definit aceste poziții.

### Exemplul 1

Se dă o stivă cu  $n$  monede. Alternativ, doi jucători X și Y pot extrage din stivă maximum  $k$  monede. Cel care nu mai poate muta, pierde. Inițial mută X. Se cere:

- a) Să se determine dacă X are strategie sigură de câștig.
- b) În caz afirmativ să se programeze mutările lui X, cele ale lui Y fiind citite de la intrare. (*Jocul Bachet, 1612*)

### Soluție

Strategia câștigătoare se bazează pe împărțirea configurațiilor de joc în două mulțimi disjuncte, și anume:

- configurația *pară* se definește ca fiind acel  $n$ , cu proprietatea ca  $n \% (k+1) == 0$
- configurația *impară* se definește ca fiind acel  $n$ , cu proprietatea ca  $n \% (k+1) != 0$

### Teorema 1

În urma oricărei mutări, aplicate unei configurații pare, se trece într-o configurație impară.

*Demonstrație*

Configurația pară este caracterizată de faptul ca  $n$  este divizibil cu  $k+1$ . La fiecare pas putem extrage  $p$  monede,  $p$  cuprins între 1 și  $k$ . Deci în noua configurație  $n$  nu va fi divizibil cu  $k+1$ .

## Teorema 2

Dintr-o configurație impară, se poate trece, în urma unei mutări convenabile, într-o configurație pară.

### *Demonstrație*

Să presupunem că în stivă avem  $n$  monede. Cea mai apropiată configurație pară (cu mai puține monede decât  $n$ ) se obține extrăgând  $n \% (k+1)$  monede.

**Deci jucătorul X fiind primul la mutare, va câștiga (mutând corect, fără greșeli) dacă  $n \% (k+1) \neq 0$ .**

## Exemplul 2

Se dă o stivă cu  $n$  monede. Dacă numărul monedelor din stivă, este mai mare decât un număr natural dat  $p$ , atunci din stivă se pot extrage maximum  $k+1$  monede, în caz contrar se pot extrage maximum  $k$  monede. Alternativ, doi jucători X și Y, extrag din stivă monede, respectând condiția de mai sus. Inițial mută X. Se cere:

- Să se determine dacă X are strategie sigură de câștig.
- În caz afirmativ să se programeze mutările lui X, cele ale lui Y fiind citite de la intrare.

### *Soluție*

Să determinăm cel mai mare număr natural  $w$ , mai mic sau egal cu  $p$ , astfel încât  $w \% (k+1) = 0$ . Deci cel care va ajunge să ia primul din stivă cu  $w$  monede, va pierde. Deci pentru ca Y să fie obligat să extragă dintr-o stivă cu  $w$  monede, X va trebui să mute în așa fel încât să lase în stivă  $w$  monede. Asta înseamnă că X trebuie să câștige pentru o stivă de  $n-w$  monede, din care se pot extrage la fiecare pas, maxim  $k+1$  monede, lucru realizabil doar dacă  $n-w$  este multiplu de  $k+1$ .

## Exemplul 3.

### Jocul NIM

Se dau  $n$  grămezi, fiecare conținând un anumit număr de pietre. Doi jucători vor începe să ia alternativ din pietre, astfel: la fiecare pas, jucătorul aflat la mutare trebuie să îndepărteze un număr nenul de pietre dintr-o singură grămadă. Câștigătorul este cel care ia ultima piatră.

### Cerință

Pentru  $t$  configurații de joc date, să se determine dacă jucătorul care ia primele pietre are strategie sigură de câștig.

### Date de intrare

Pe prima linie a fișierului de intrare `nim.in` se va afla numărul  $t$  de configurații. Pe următoarele  $2 \cdot t$  linii se vor afla descrierile jocurilor, astfel: pe linia  $2 \cdot i$  se va afla numărul  $n_i$  de grămezi care alcătuiesc jocul  $i$ , iar pe linia  $2 \cdot i + 1$  se vor afla  $n_i$  numere, dimensiunile grămezilor.

### Date de ieșire

În fișierul de ieșire `nim.out` se vor afișa  $t$  linii, pe linia  $i$  aflându-se mesajul "DA", dacă primul jucător are strategie sigură de câștig în jocul  $i$ , respectiv "NU", în caz contrar.

### Restricții

- $1 \leq t \leq 100$
- $1 \leq n_i \leq 10\,000$
- Numărul de pietre din oricare grămadă este natural pozitiv mai mic sau egal cu  $2 \cdot 10^9$

### Exemplu

| nim.in  | nim.out |
|---------|---------|
| 2       | NU      |
| 4       | DA      |
| 1 3 5 7 |         |
| 3       |         |
| 4 8 17  |         |

### Soluție

Pentru cazul trivial în care numărul de grămezi este egal cu 1, primul jucător are evident strategie de câștig, el putând lua toate pietrele din grămadă.

Dacă numărul de grămezi este egal cu 2, primul jucător are strategie de câștig atunci când numărul de pietre din prima grămadă este diferit de numărul de pietre din cea de-a doua, strategia lui fiind cea de a aduce tot timpul grămada mai mare la numărul de pietre al grămezii mai mici, și cum jocul este finit, înseamnă că primul jucător o să aducă jocul în starea  $(0, 0)$ .

Dacă numărul de grămezi este mai mare decât doi strategia se complică și nu se mai observă cu "ochiul liber". Dar ideea este similară.

### O strategie posibilă ar fi următoarea:

Considerăm reprezentările binare ale numărului de pietre din fiecare grămadă.

Efectuăm operația XOR (sau exclusiv  $\wedge$ ) pe aceste reprezentări.

**Dacă rezultatul acestei operații este diferit de 0, primul jucător va avea strategie sigură de câștig.**

**Dacă rezultatul acestei operații este 0, al II-lea jucător va avea strategie sigură de câștig (deci primul jucător este într-o stare de pierdere).**

Să demonstrăm că primul jucător are strategie sigură de câștig dacă  $rez = p_1 \wedge p_2 \wedge \dots \wedge p_n \neq 0$ .  
Ideea este de a muta astfel încât să aducă pe al doilea jucător într-o stare de pierdere, adică pentru care  $rez = p_1 \wedge p_2 \wedge \dots \wedge p_n = 0$ .

Fie  $i$  cel mai semnificativ bit egal cu 1 din  $rez$ . Alegem o grămadă  $j$  care are în reprezentarea sa binară acest bit egal cu 1.

Voi extrage din această grămadă  $p_j - (p_j \wedge rez)$  pietre ( $p_j \wedge rez$  este mai mic decât  $p_j$ , pentru că bitul  $i$  se anulează) După această mutare  $p_1 \wedge p_2 \wedge \dots \wedge p_n = 0$ , iar al doilea jucător va trebui să extragă un număr de pietre, ceea ce va conduce iarăși la o stare pentru care  $p_1 \wedge p_2 \wedge \dots \wedge p_n \neq 0$  (adică primul jucător va avea ce muta).

Dacă suntem într-o stare cu  $rez = 0$ , orice mutare ar face jucătorul care este la rând va aduce jocul într-o stare cu  $rez \neq 0$  (să spunem că din gămada  $j$  va luam  $x$  piese; bitul cel mai semnificativ din  $x$  se schimbă sigur)

De exemplu, dacă avem o gramadă cu o piatră, o gramadă cu trei pietre, o gramadă cu cinci pietre și o gramadă cu șapte pietre:

0

000

00000

000000

Atunci vom avea:

1 = (0001)

3 = (0011)

5 = (0101)

7 = (0111)

$rez = (0000)$

Conform propoziției de mai sus această stare este de pierdere.

Să considerăm un alt exemplu

$N = 3$

$p = (4, 8, 17)$

Reprezentările binare sunt

$p_1$ : 00100

$p_2$ : 01000

$p_3$ : 10001

$rez$ : 11101

Cel mai semnificativ bit 1 din  $rez$  corespunde lui  $2^4$

Aleg o grămadă  $j$  pentru care  $p_j$  are bitul corespunzător lui  $2^4$  egal cu 1 (aceasta trebuie să existe și este  $p_3$ )

Extrag din grămadă 3:  $p_3 - (rez \wedge p_3)$  piese, adică în grămadă  $p_3$  rămân  $p_3 \wedge rez$  piese

rez: 11101

p3: 10001

rez<sup>p3</sup>: 01100=12 piese

Extrag p3-(rez<sup>p3</sup>)=17-12=5 piese din grămada 3 și se obține p3=12:

p1: 00100

p2: 01000

p3: 01100

rez:00000

Orice mutare ar face al doilea jucător, ca determina ca rez să devină iarăși  $\neq 0$ .

## Probleme similare

### Problema 1

Pe o tablă de șah, care are  $n \cdot m$  căsuțe, sunt plasați pe prima linie  $n$  pioni albi și pe ultima linie  $n$  pioni negri. Fiecare dintre cei doi jucători poate muta un singur pion, care îi aparține, un număr strict pozitiv de căsuțe în sus sau în jos, astfel încât să nu ajungă vreun pion alb să fie mai jos decât pionul negru de pe aceeași coloană. Pierde jucătorul care nu mai poate muta.

### Soluție

Această problemă este o deghizare a jocului NIM, numărul de pătrățele libere între pionul alb și pionul negru de pe coloana  $i$  putând fi considerat numărul de pietre din grămada  $i$ . Singura diferență este că se pot adăuga pietre la grămadă (existând posibilitatea mutării înapoi).

Această problemă se rezolvă ușor, jucătorul care are strategie de câștig putând evita asemenea mutări. O astfel de mutare poate fi utilă numai pentru jucătorul care este într-o poziție de pierdere.

Când jucătorul care nu are strategie de câștig mută înapoi  $x$  casuțe, celălalt jucător va muta pionul propriu de pe aceeași coloană cu  $x$  căsuțe în față, astfel ajungându-se la aceeași stare cu cea existentă cu două mutări anterior (considerând diferența pozițiilor).

### Problema 2

Această problemă a fost propusă spre rezolvare participanților la barajul pentru selecția lotului național din 1997.

Pe o linie sunt plasate la coordonate întregi  $2 \cdot n$  piese roșii și albastre.

Fiecare piesă roșie poate fi mutată în dreapta oricâte poziții astfel încât să nu sară peste o piesă albastră, iar piesele albastre pot fi mutate oricâte poziții la stânga astfel încât să nu depășească vreo piesă roșie. Piesele vor alterna: roșu, albastru, roșu, albastru etc. Pierde jucătorul care nu mai poate muta.



## Soluție

Această problemă poate fi, de asemenea, redusă la jocul NIM. Diferențele pozițiilor perechilor de piese roșii și albastre consecutive constituie numărul de pietre al grămezilor din jocul NIM.

## Problema 3 (Misere NIM)

Se consideră  $n$  grămezi de pietre, jucătorii mută alternativ, fiecare jucător extrăgând oricâte pietre dintr-o singură grămadă. Cel care ia ultima piatră pierde jocul.

Strategia acestui joc este similară cu cea aplicată în jocul NIM cu câteva mici diferențe. Jucătorul care are strategie de câștig în poziția curentă în cadrul jocului NIM face aceeași mutare pe care ar face-o în cazul jocului NIM, exceptând cazul în care această mutare lasă doar grămezi cu o singură piatră și numărul acestor grămezi este par.

În această situație, dacă ar trebui să ia  $x$  pietre, jucătorul poate lua  $x - 1$  pietre din grămada actuală, pentru ca numărul de grămezi să fie impar și adversarul să facă ultima mutare.

## Jocul TZIANSITZI (*Concursul de Informatică prin E-mail, Oradea, 1997*)

Jocul TZIANSITZI, ceea ce în traducere înseamnă "alegerea pietrelor", este un joc popular din China. Doi jucători extrag alternativ pietre din două grămezi. La fiecare mișcare un jucător poate extrage același număr de pietre din ambele grămezi sau un număr oarecare de pietre din una dintre grămezi. Câștigă jucătorul care ia și ultima piatră.

## Cerința

Se cere să testați dacă, pentru o configurație dată, primul jucător are sau nu strategie de câștig și dacă da să programați mutările primului jucător, mutările celui de al doilea jucător fiind citite de la tastatură.

## Date de intrare

Programul nu citește date din nici un fișier.

## Date de ieșire

Programul nu va scrie date în nici un fișier.

## Interacțiune

Programul vostru va interacționa cu un program al comisiei care rulează în paralel. În cadrul unui set de test (test-case), programul vostru va trebui să dea răspunsul corect pentru mai multe teste. Pentru fiecare test din set, interacțiunea are următorul format:

1. Programul vostru citește o linie de la intrarea standard, care conține două numere naturale  $n$   $m$  separate printr-un spațiu ( $n$  este numărul de pietre din prima grămadă,  $m$  este numărul de pietre din a doua grămadă pentru testul respectiv). Dacă  $n$  și  $m$  sunt nule, înseamnă că setul de test s-a terminat.
2. Apoi, programul vostru va determina dacă primul jucător are sau nu strategie sigură de câștig. În caz afirmativ, va scrie o linie care conține doar valoarea 1. În caz contrar (deci dacă primul jucător nu are strategie sigură de câștig) se va scrie o linie care conține doar valoarea 0 (zero) și apoi o linie care conține cuvântul DONE.

3. Dacă primul jucător are strategie sigură de câștig, programul vostru va efectua un număr de runde. O rundă constă din scrierea unei linii la ieșirea standard, care descrie operația efectuată de primul jucător în runda respectivă. O operație este descrisă prin numărul gramezii din care jucătorul extrage pietre (1 pentru prima gramadă, 2 pentru cea de a doua, respectiv 3 pentru ambele gramezi), urmat de un spațiu, apoi de un număr natural nenul care reprezintă numărul de pietre extrase. Apoi, se va citi o linie de intrare standard care descrie operația efectuată de al doilea jucător (operația fiind specificată în același mod). Când programul vostru a executat ultima mutare, afișează o linie specială, conținând numai cuvântul DONE

### **Punctaj**

Programul vostru va primi punctajul corespunzător setului de teste curent dacă și numai dacă a rezolvat corect toate testele conținute în setul respectiv.

Programul vostru primește 0 puncte pe un anumit test în următoarele situații :

- afișează în cadrul unei runde o operație ce nu poate fi executată la momentul respectiv;
- afișează că primul jucător are strategie sigură de câștig atunci când nu are (sau invers);
- atunci când primul jucător are strategie sigură de câștig, programul vostru nu câștigă;
- formatul de afișare impus în problema nu este respectat.

### **Instrucțiuni de programare**

Programatorii în C/C++ trebuie să execute flush după ce au terminat de scris o linie completă la ieșire. Acest lucru este valabil și pentru linia conținând DONE

În cazul programelor scrise în C++ și utilizarea iostreams, se va folosi următoarea secvență pentru citirea de la intrarea standard și scrierea la ieșirea standard:

```
cin>>x;
```

```
cout<<op<<"\n"<<flush;
```

În cazul programelor în C sau C++ și utilizarea scanf și printf, se va folosi următoarea secvență pentru citirea de la intrarea standard și scrierea la ieșirea standard:

```
scanf ("%d", &x);
```

```
printf ("%d\n",op); fflush (stdout);
```

În cazul programelor în Pascal se va folosi următoarea secvență pentru citire de la intrarea standard se utilizează readln, iar pentru afișarea unei linii se utilizează writeln :

```
readln(x);
```

```
writeln(op);
```

### **Restricții și precizări**

$1 \leq n \leq 10^{16}$

Într-un set de teste sunt cel mult 7 teste.

### **Exemplu de interacțiune**

| Operație    | Semnificație                                     |
|-------------|--------------------------------------------------|
| Citeste 1 2 | n=1, m=2                                         |
| Scrie 0     | Primul jucător nu are strategie sigură de câștig |
| Scrie DONE  | S-a terminat primul test                         |
| Citeste 5 6 | n=5, m=6                                         |

|             |                                                       |
|-------------|-------------------------------------------------------|
| Scrie 1     | Primul jucator are strategie sigura de castig         |
| Scrie 2 3   | Primul jucator extrage 3 pietre din gramada 2         |
| Citeste 1 1 | Al doilea jucator extrage 1 pietre din gramada 1      |
| Scrie 3 2   | Primul jucator extrage 2 pietre din ambele gramezi    |
| Citeste 3 1 | Al doilea jucator extrage 1 pietre din ambele gramezi |
| Scrie 1 1   | Primul jucator extrage 1 pietre din gramada 1         |
| Scrie DONE  | Acest test s-a terminat                               |
| Citeste 0 0 | Acest test-case s-a terminat                          |

## Soluție

Rezolvarea se bazează pe reprezentarea numerelor naturale în sistemul de numerație Fibonacci.

### Definiție

Perechea  $(n, m)$ , cu  $n < m$  se numește distinsă dacă:

1. reprezentarea *Fibonacci* a lui  $n$  se termină cu un număr par de zerouri;
2. reprezentarea *Fibonacci* a lui  $m$  se obține din reprezentarea *Fibonacci* a lui  $n$  prin adăugarea la sfârșit a unui 0.

### Exemple:

1.  $n = 1$  (1, în sistemul *Fibonacci*),  $m = 2$  (10, în sistemul *Fibonacci*)
2.  $n = 3$  (100, în sistemul *Fibonacci*),  $m = 5$  (1000, în sistemul *Fibonacci*)
3.  $n = 4$  (101, în sistemul *Fibonacci*),  $m = 7$  (1010, în sistemul *Fibonacci*)

### Propoziție

Orice număr natural aparține unei singure perechi distinsă.

### Demonstrație

Dacă reprezentarea numărului în sistemul de numerație Fibonacci se termină cu un număr par de zerouri, al doilea număr din pereche se obține adăugând un 0 final, altfel al doilea număr din pereche se obține prin suprimarea unui 0 final.

### Propoziție

În șirul  $\{m - n \mid (n, m) \text{ pereche distinsă}\}$  fiecare număr natural apare o singură dată.

### Demonstrație

Fie  $d \in \mathbb{N}$ , iar  $x_p x_{p-1} \dots x_1$ , reprezentarea *Fibonacci* a lui  $d$ .

$$d = x_p x_{p-1} \dots x_1 = x_p \cdot f_p + x_{p-1} \cdot f_{p-1} + \dots + x_1 \cdot f_1$$

Dacă reprezentarea Fibonacci a lui  $d$  se termină cu un număr impar de zerouri, atunci

$$n = x_p x_{p-1} \dots x_1 0 = x_p \cdot f_{p+1} + x_{p-1} \cdot f_p + \dots + x_1 \cdot f_2 \quad (5)$$

$$m = x_p x_{p-1} \dots x_1 00 = x_p \cdot f_{p+2} + x_{p-1} \cdot f_{p+1} + \dots + x_1 \cdot f_3. \quad (6)$$

$$m - n = x_p \cdot (f_{p+2} - f_{p+1}) + x_{p-1} \cdot (f_{p+1} - f_p) + \dots + x_1 \cdot (f_3 - f_2) = x_p \cdot f_p + x_{p-1} \cdot f_{p-1} + \dots + x_1 \cdot f_1 = d$$

Dacă reprezentarea *Fibonacci* a lui  $d$  se termină cu un număr par de zerouri, deci  $x_1 = x_2 = x_{2k} = 0, x_{2k+1} \neq 0$ , atunci:

$$n = x_p x_{p-1} \dots x_{2k+2} \underbrace{0101 \dots 01}_{01 \text{ de } k+1 \text{ ori}} = x_p \cdot f_{p+1} + x_{p-1} \cdot f_p + \dots + x_{2k+2} \cdot f_{2k+3} + (f_{2k+1} + f_{2k-1} + \dots + f_3 + f_1)$$

$$m = x_p x_{p-1} \dots x_{2k+2} \underbrace{0101 \dots 010}_{01 \text{ de } k+1 \text{ ori}} = x_p \cdot f_{p+2} + x_{p-1} \cdot f_{p+1} + \dots + x_{2k+2} \cdot f_{2k+4} + (f_{2k+2} + f_{2k} + \dots + f_4 + f_2)$$

$$m - n = x_p \cdot (f_{p+2} - f_{p+1}) + x_{p-1} \cdot (f_{p+1} - f_p) + \dots + x_{2k+2} \cdot (f_{2k+4} - f_{2k+3}) + (f_{2k+2} - f_{2k+1}) + \dots + (f_2 - f_1) = x_p \cdot f_p + x_{p-1} \cdot f_{p-1} + \dots + x_{2k+2} \cdot f_{2k+2} + f_{2k+1} = d, \text{ conform proprietății (3).}$$

Unicitatea se poate demonstra, aplicând formulele (5), (6) în sens invers (formula (5) corespunde cazului în care  $n$  se termină cu 0, formula (6), pentru cazul în care  $n$  nu se termină cu 0).

### *Teoremă*

Dacă  $n$  și  $m$ , numărul de pietre din prima, respectiv cea de a doua grămadă nu formează o pereche distinsă, atunci primul jucător are strategie sigură de câștig (câștigă dintr-o singură mutare, sau poate aduce jocul printr-o singură mutare într-o situație în care  $(n, m)$  formează o pereche distinsă.

### *Schița de demonstrație*<sup>1</sup>

1. Dacă  $n = m$  sau  $n = 0$  primul jucător câștigă dintr-o singură mutare.
2. Dacă  $0 < n < m$ , determinăm perechea distinsă  $(n, p)$  și perechea distinsă  $(a, b)$  astfel încât  $b - a = m - n$ .  
Dacă  $p < m$ , atunci se poate trece cu o singură mutare de la perechea  $(n, m)$  la perechea distinsă  $(n, p)$ .  
Dacă  $p > m$ , deducem  $p - n > b - a = m - n$ , deci perechea distinsă  $(n, p)$  este "mai mare" decât perechea distinsă  $(a, b)$ , deci  $p > m, a > n$ . Deducem  $n - a = m - b$ , deci dintr-o singură mișcare putem trece la perechea distinsă  $(a, b)$ .
3. Dacă perechea  $(n, m)$  este distinsă, după orice mișcare ea va înceta să mai fie distinsă.

<sup>1</sup> O demonstrație riguroasă se găsește în Iaglom, A.M.; Iaglom, I.M. - *Probleme neelementare tratate elementar*. Editura Tehnică, 1983.