



# Etapa FINALĂ

În numărul trecut am publicat enunțurile celor cinci probleme propuse spre rezolvare la faza finală a concursului de programare organizat de Ginfo. Vă prezentăm acum descrierile soluțiilor oficiale. Implementările sunt disponibile pentru download la [www.ginfo.ro/revista/11\\_7](http://www.ginfo.ro/revista/11_7).

## CPAR7P1: Bile

Problema poate fi rezolvată folosind metoda *programării dinamice*. Vom construi o matrice  $cost$ ; elementul  $cost_{ij}$  va indica lungimea minimă totală a barelor folosite pentru a lega bilele cuprinse între pozițiile  $i$  și  $j$ . În final, elementul  $cost_{1,2n}$  va conține lungimea totală a barelor necesare pentru a lega toate bilele.

Este evident că toate elementele de pe diagonala principală vor avea valoarea 0. Inițial vom considera că celelalte elemente (doar cele aflate deasupra diagonalei principale sunt semnificative) au valoarea  $\infty$ .

Pentru două bile vecine (aflate pe pozițiile  $i$  și  $i + 1$ ) care au culori diferite putem afirma că valoarea  $cost_{i,i+1}$  este 3 deoarece avem nevoie de două bare verticale și una orizontală, toate având lungimea 1.

Nu are rost să determinăm valorile corespunzătoare pozițiilor  $i$  și  $i + k$ , pentru care  $k$  este par, deoarece nu avem posibilitatea de a lega un număr impar de bile. Așadar, la pașii următori vom determina valorile corespunzătoare pozițiilor  $i$  și  $i + k$ , pentru care  $k$  este impar.

Există două posibilități de a determina valoarea minimă corespunzătoare pozițiilor  $i$  și  $j$ .

Dacă bilele au culori diferite, atunci putem să legăm cele două bile cu o bară orizontală de lungime  $j - i$  și două verticale care au o înălțime cu 1 mai mare decât cea mai mare înălțime a barelor verticale de pe pozițiile cuprinse între  $i + 1$  și  $j - 1$ .

Se observă că mai avem nevoie de o matrice ( $height$ ), elementul  $height_{ij}$  indicând înălțimea maximă a barelor verticale cuprinse între pozițiile  $i$  și  $j$ .

O altă posibilitate de obținere a unei valori pentru  $cost_{ij}$  (indiferent de culoarea bilelor de pe pozițiile  $i$  și  $j$ ) este de a alege o poziție intermediară  $k$  și a aduna valorile  $cost_{ik}$  și  $cost_{kj}$ . Valoarea  $height_{ij}$  va fi egală cu maximul valorilor  $height_{ik}$  și  $height_{kj}$ . Vom alege toate valorile  $k$  pentru care  $k - i$  este un număr par.

Evident, vom alege cea mai mică valoare pe care o determinăm folosind cele două posibilități.

Folosind această metodă vom completa toate elementele  $cost_{ij}$  pentru care există o posibilitate de a lega toate

bilele cuprinse între  $i$  și  $j$ . Pentru toate celelalte elemente se va păstra valoarea  $\infty$ .

După determinarea valorii minime trebuie reconstituite legăturile realizate. Pentru aceasta trebuie să determinăm modul în care au fost calculate valorile  $cost_{ij}$ . Va trebui să știm dacă am folosit prima sau a doua metodă și, în cazul celei de-a doua metode, care este valoarea  $k$  aleasă. Vom păstra încă o matrice ( $inter$ ), iar elementele  $inter_{ij}$  vor avea valoarea  $-1$  dacă a fost folosită prima posibilitate sau valoarea  $k$  dacă a fost folosită cea de-a doua.

Legăturile între pozițiile  $i$  și  $j$  se vor afișa printr-o rutină recursivă; inițial ea va fi apelată pentru pozițiile 1 și  $2 \cdot n$ .

În cazul în care se folosește prima posibilitate pentru pozițiile  $i$  și  $j$ , vom "tipări" legătura dintre  $i$  și  $j$  și apoi vom apela recursiv rutina pentru pozițiile  $i + 1$  și  $j - 1$ .

În cazul în care se folosește cea de-a doua posibilitate pentru pozițiile  $i$  și  $j$ , nu vom tipări nimic și vom apela recursiv rutina pentru pozițiile  $i$  și  $k$  și pentru pozițiile  $k + 1$  și  $j$ .

## CPAR7P2: Plăți și încasări

Enunțul problemei poate fi "tradus" astfel:

*Se dă un șir de numere întregi, dintre care cel puțin unul este pozitiv. Să se determine subsecvența de sumă maximă a acestui șir. În cazul în care există mai multe astfel de subsecvențe trebuie determinată cea mai lungă dintre acestea. Dacă există mai multe subsecvențe de sumă maximă care au aceeași lungime, trebuie aleasă cea care începe cu elementul având indicele cel mai mic.*

Această problemă (împreună cu soluția) a fost prezentată în articolul *Secvența de sumă maximă*, scris de **Clara Ionescu** și publicat în numărul 11/2 (februarie 2001) al revistei noastre.

## CPAR7P3: Robot

Pentru a păstra traseul robotului vom folosi o matrice ale cărei elemente vor reprezenta pătrățelele caroiului. Fiecare element va păstra informații referitoare la laturile pătrățelei corespunzătoare: existența unei linii deasupra, sub,



în stânga sau în dreapta pătrățelului. Cea mai simplă posibilitate de implementare este folosirea a patru biți, fiecare reprezentând una dintre cele patru laturi și având valoarea 1 dacă există linie pe acea latură și 0 în caz contrar.

După construirea matricei, trebuie determinat dreptunghiul vid de arie minimă care se formează. Pentru simplificare, vom trata patru cazuri:

1. dreptunghiuri de lungime și înălțime 1.
2. dreptunghiuri de lungime 1 și înălțime diferită de 1.
3. dreptunghiuri de lungime diferită de 1 și înălțime 1.
4. dreptunghiuri de lungime și înălțime diferite de 1.

Pentru primul caz va trebui doar să căutăm pătrățele care au linii pe toate cele patru laturi ale lor. Dacă găsim un astfel de pătrățel, atunci soluția problemei este un dreptunghi de arie 1.

Pentru al doilea caz va trebui să căutăm pătrățele care au linii pe trei dintre laturi (deasupra, sub și la stânga). Vom parcurge apoi pătrățelele din dreapta sa care au linii deasupra și dedesubt.

Când întâlnim primul pătrățel care nu respectă această condiție, e posibil să fi ajuns la ultimul pătrățel al dreptunghiului. Dacă acesta are linii deasupra, sub și la dreapta, atunci am găsit un nou dreptunghi. Dacă aria acestuia este mai mică decât cea a celui mai mic dreptunghi determinat anterior, vom păstra noua soluție.

Pentru al treilea caz va trebui să căutăm pătrățele care au linii pe trei dintre laturi (deasupra, la stânga și la dreapta). Vom parcurge apoi pătrățelele de sub el, care au linii la stânga și la dreapta.

Când întâlnim primul pătrățel care nu respectă această condiție, e posibil să fi ajuns la ultimul pătrățel al dreptunghiului. Dacă acesta are linii dedesubt, la stânga și la dreapta, atunci am găsit un nou dreptunghi. Dacă aria acestuia este mai mică decât cea a celui mai mic dreptunghi determinat anterior, vom păstra noua soluție.

Pentru al patrulea caz vom căuta pătrățele care au linii pe două dintre laturi (la stânga și deasupra). Vom parcurge apoi pătrățelele din dreapta sa care au linii numai deasupra.

Când întâlnim un pătrățel care nu respectă această condiție, e posibil să fi ajuns la latura din dreapta a dreptunghiului. Acest pătrățel trebuie să aibă linii pe latura de deasupra și pe cea din dreapta. Vom parcurge acum pătrățelele aflate sub pătrățelul inițial, care au linii numai la stânga.

Când întâlnim un pătrățel care nu respectă această condiție, e posibil să fi ajuns la latura de jos a dreptunghiului. Acest pătrățel trebuie să aibă linii pe latura de dedesubt și pe cea din stânga. Dacă toate condițiile au fost îndeplinite, am reușit să determinăm un posibil dreptunghi.

În plus, trebuie să verificăm dacă acesta este unul valid. Pentru aceasta vom parcurge pătrățelele de pe latura de jos și vom verifica dacă nu au și alte linii cu excepția celei de dedesubt.

Analog, pătrățelele de pe latura din dreapta trebuie să aibă linii numai pe latura din dreapta. În sfârșit, pătrățelul

din colțul dreapta-jos trebuie să aibă linii pe latura din dreapta și pe cea de dedesubt.

Dacă și aceste condiții sunt îndeplinite, atunci am găsit un nou dreptunghi. Dacă aria acestuia este mai mică decât cea a celui mai mic dreptunghi determinat anterior, vom păstra noua soluție.

O observație importantă este că nu trebuie să verificăm decât pătrățelele de pe laturi, nu și pe cele din interior. Aceasta se datorează continuității liniei care formează traseul robotului. Pentru ca o linie să se afle în interiorul unui dreptunghi, aceasta ar fi trebuit să pornească din exteriorul acestuia, deci s-ar fi intersectat cu cel puțin una dintre laturi și dreptunghiul nu ar mai fi fost valid.

#### CPAR7P4: Star Trek

Este evident că pentru ca cercurile în care pot acționa *tri-corder*-ele să fie disjuncte, trebuie să alegem cei mai apropiați doi membri ai echipajului.

După determinarea acestei perechi, raza maximă de acțiune poate fi calculată înjumătățind distanța dintre ele. Deoarece raza de acțiune trebuie să fie un număr întreg, valoarea obținută va fi trunchiată.

Observăm că, dacă razele sunt egale cu jumătatea distanței, cele două cercuri sunt tangente. Din acest motiv, în cazul în care valoarea obținută este un număr întreg, va trebui să mai scădem o unitate (acest caz apare în exemplul din enunț).

Așadar, problema se reduce la determinarea celei mai apropiate perechi de puncte cunoscând coordonatele a  $n$  puncte din plan. Această problemă a fost propusă spre rezolvare la etapa a doua a ediției trecute a concursului nostru. Enunțul a apărut în numărul 10/7 (noiembrie 2000) al *GInfo*, iar soluția în numărul 11/1 (ianuarie 2001).

#### CPAR7P5: Windows XP

Această problemă poate fi rezolvată folosind o strategie de tip *greedy*.

Pentru început vom crea o "listă de așteptare" pentru fiecare aplicație, listă care va conține toate aplicațiile care trebuie instalate în prealabil pentru ca programul de instalare să poată fi lansat în execuție.

Este evident că vom începe instalarea tuturor aplicațiilor care nu necesită instalarea prealabilă a altor aplicații.

Apoi, la fiecare pas, vom determina toate aplicațiile al căror proces de instalare s-a încheiat și le vom elimina din listele de așteptare ale aplicațiilor care nu au fost încă instalate.

După aceea vom verifica dacă există aplicații pentru care listele de așteptare sunt vide. Dacă găsim astfel de aplicații putem să lansăm în execuție programul de instalare corespunzător.

La fiecare pas vom memora aplicațiile ale căror programe de instalare se află în execuție. În final vom afișa numărul de pași parcurși și, corespunzător fiecărui pas, lista aplicațiilor al căror proces de instalare este în desfășurare.