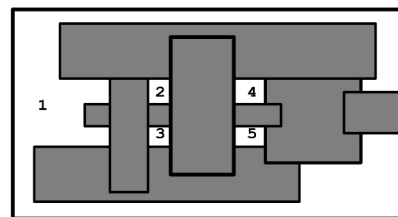


Descrierea soluției problemei 1 – colaj

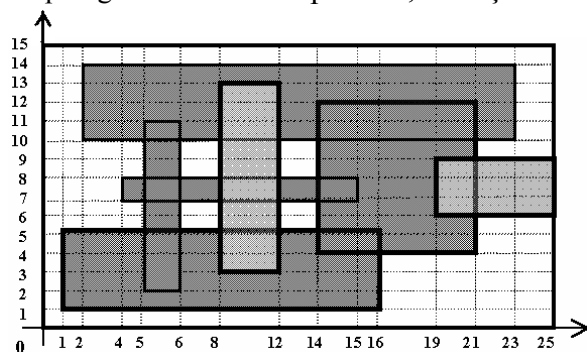
Propunător: prof Carmen MINCĂ

Se consideră următorul exemplu

colaj.in	colaj.out	Explicație
7 25 15 1 1 16 5 14 4 21 12 4 7 15 8 19 6 25 9 2 10 23 14 5 2 6 11 8 3 12 13	5	Colajul realizat de echipa A este cel din desenul alăturat. Se observă 5 suprafețe albe distincte conținute de colaj.



O soluție se poate obține pe baza metodei împărțirii în “zone elementare”, care nu se intersectează cu nici un dreptunghi. Pentru exemplul dat, se obține următoarea împărțire în zone elementare:



Sunt reprezentate și vârfurile planșei: $(0; 0)$, $(25; 0)$, $(0, 15)$ și $(25; 15)$. Se observă că aceste zone elementare formează o matrice A având un număr de linii, respectiv coloane, din mulțimea $\{2n-1, 2n, 2n+1\}$. Numărul minim de linii $2n-1$, respectiv coloane: $2n-1$, se obține dacă există dreptunghiuri cu laturile pe marginile de jos și sus, respectiv din stânga și dreapta, ale planșei. Numărul de linii, respectiv coloane, crește cu 1 dacă niciun dreptunghi nu este situat pe marginea de jos, respectiv din stânga, a planșei.

Numărul de linii, respectiv coloane, crește cu 1 dacă niciun dreptunghi nu este situat pe marginea de sus, respectiv din dreapta, a planșei.

Fie x_0 , respectiv y_0 , valoarea care se adaugă la $2n-1$ pentru a se obține numărul de coloane, respectiv linii, ale matricei A. Pentru matricea din exemplu, $x_0=1$, $y_0=2$.

0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	1	0	1	0	1	1	1	1	0	0
0	0	0	0	1	0	1	0	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	1	0	1	1	1	1	1	1
0	0	0	0	1	0	1	0	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Elementul $A[i][j]$ al matricei va avea valoarea 1 dacă și numai dacă există cel puțin un dreptunghi care să conțină zona elementară respectivă.

Matricea este inversată față de axa Ox.

Pentru exemplul dat, matricea A are $2n+1$ linii și $2n$ coloane, iar conținutul ei este cel alăturat.

Pentru construcția matricei vom folosi doi vectori X și Y. Vectorul X va reține toate abscisele vârfurilor dreptunghiurilor. Vectorul Y va reține toate ordonatele vârfurilor dreptunghiurilor.

Zonele elementare din desen corespund câte unui element $A[i][j]$ din matrice și sunt dreptunghiuri care au coordonatele vârfurilor opuse: $(X[i], Y[j])$ și $(X[i+1], Y[j+1])$ $i, j=1, 2, \dots, 2n$.

Dacă nu există dreptunghiuri incluse în interiorul unui alt dreptunghi, atunci vectorii au câte $2n$ componente, fiecare având toate valorile distincte. Altfel sunt memorate coordonatele dreptunghiurilor care un sunt incluse într-un alt dreptunghi.

Se sortează crescător vectorii. Pentru exemplul dat, vectorii vor avea conținutul:

$X = (1, 2, 4, 5, 6, 8, 12, 14, 15, 16, 19, 21, 23)$

$Y = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)$

Se memorează coordonatele vârfurilor dreptunghiurilor din exemplul dat într-un vector V cu n elemente de tip structură:

```
struct dr { int a,b,c,d; int pa,pb,pc,pd;};
```

unde:

1. (a,b) sunt coordonatele vârfului stânga-jos al dreptunghiului
2. (c,d) sunt coordonatele vârfului dreapta-sus al dreptunghiului
3. pa este poziția pe care apare a în vectorul sortat X
4. pb este poziția pe care apare b în vectorul sortat Y
5. pc este poziția pe care apare c în vectorul sortat X
6. pd este poziția pe care apare d în vectorul sortat Y

Se inițializează matricea A cu 0. Pentru fiecare dreptunghi, se marchează cu 1 toate zonele acoperite de acesta:

```
for (k=1; k<=n; k++)  
    for (j=v[k].pa; j<v[k].pc; j++)  
        for (i=v[k].pb; i<v[k].pd; i++) a[i+y0][j+x0]=1;
```

Se borpdează matricea cu 1, pentru a nu ieși în exteriorul ei în timpul aplicării algoritmului FILL. Se caută fiecare element din matrice cu valoarea 0, se umple atât elementul cât și vecinii acestuia cu valoarea 1 și se numără zonele care au valoarea 0. Acest număr va fi numărul de suprafețe albe din colaj.

După aplicarea FILL-ului matricea A din exemplu va avea toate valorile egale cu 1.

Sursele `colajC.cpp` și `colajP.pas` constituie implementarea a acestei soluții.

O soluție care obține un punctaj parțial constă în a contrui o matrice A cu p linii și m coloane, elementele ei memorând valori 0 și 1.

Fiecărui dreptunghi cu vârful stânga-jos, respectiv dreapta-sus, de coordonate (x_a, y_a) , respectiv (x_b, y_b) , îi corespunde în matricea A o zonă în care toate elementele au valoarea 1:

```
for (l=y_a; l<y_b; l++)  
    for (c=x_a; c<x_b; c++) a[l][c]=1;
```

Pornind de la primul element din matrice cu valoarea 0, aplicând algoritmului FILL, umplem atât elementul cât și vecinii acestuia cu valoarea 1 și numărăm zonele care au valoarea 0. Acest număr nr va fi numărul de suprafețe albe din colaj.

Dezavantajul utilizării acestei metode constă în spațiul de memorie disponibil, insuficient pentru memorarea matricei A. Nu întotdeauna este posibilă memorarea unei matrice cu $m \times p$ ($m, p < 8000$) și cu elemente de tip char/byte. Prin “comprimarea” dreptunghiurilor utilizând metoda împărțirii în “zone elementare”, se reduce spațiul necesar memorării matricei de tip char la cel mult 201×201 octeti.

Un alt dezavantaj rezultă din timpul mare de executare, datorat aplicării algoritmului recursiv FILL pentru o matrice cu un număr mare de componente, și a dimensiunii mici a memoriei corespunzătoare segmentului de stivă. (vezi sursa `colaj_40.cpp` care obține 40p).

O îmbunătățire a punctajului se poate obține atunci când valorile m și p permit declararea unei matrice A de tip char care să memoreze toate zonele corespunzătoare dreptunghiurilor, prin eliminarea anumitor linii și coloane. Necesitând un număr mare de operații datorate ștergerilor liniilor, respectiv coloanelor, identice din A se depășește timpul de execuție.

Se elimină anumite linii ale matricei A astfel: dacă liniile $l_i, l_{i+1}, \dots, l_k$ din A sunt identice, reținem linia l_i , restul fiind șterse, nefiind necesare, realizându-se astfel o comprimare pe linii a matricei A, fiecare grupă de linii identice fiind înlocuită cu o singură linie de acest tip.

Analog, dacă coloanele $c_i, c_{i+1}, \dots, c_k$ din A sunt identice, atunci se păstrează în A doar coloana c_i , restul se șterg nefiind necesare, realizând astfel o comprimare pe coloane a matricei A, pentru fiecare grupă de coloane identice păstrându-se în A o singură coloană de acest tip. (vezi sursa `colaj_50.cpp` care obține 50p).