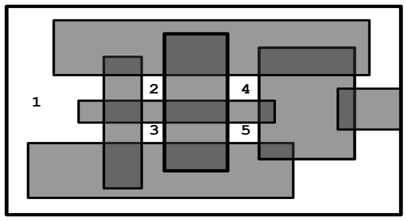
Descrierea soluţiei problemei 1 – colaj

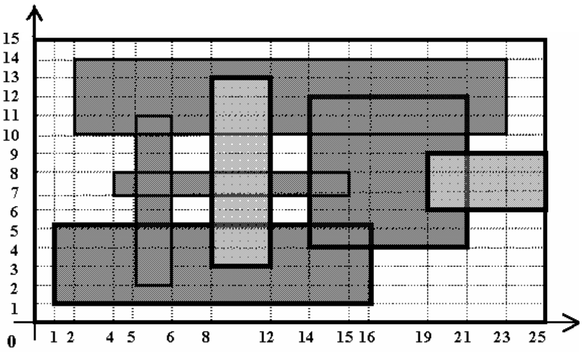
Propunător: prof Carmen MINCĂ

Se consideră următorul exemplu

colaj.in	colaj.out	*Explicaţie*
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** suprafeţe albe distincte conţinute de colaj.



O soluţie se poate obţine pe baza metodei împărţirii în "zone elementare", care nu se intersectează cu nici un dreptunghi. Pentru exemplul dat, se obţine următoarea împărţire în zone elementare:



Sunt reprezentate şi vârfurile planşei: (0; 0), (25;0), (0, 15) şi (25; 15). Se observă că aceste zone elementare formează o matrice A având un număr de linii, respectiv coloane, din mulţimea {2n-1, 2n, 2n+1}.

Numărul minim de linii 2n-1, respectiv coloane: 2n-1, se obţine dacă există dreptunghiuri cu laturile pe marginile de jos şi sus, respectiv din stânga şi dreapta, ale planşei. Numărul de linii, respectiv coloane,creşte cu 1 dacă niciun dreptunghi nu este situat pe marginea de jos, respectiv din stânga, a planşei.

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

Fie x0, respectiv y0, valoarea care se adaugă la 2n-1 pentru a se obţine numărul de coloane, respectiv linii, ale matricei A. Pentru matricea din exemplu, x0=1, y0=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 şi numai dacă există cel puţin un dreptunghi care să conţină zona elementară respectivă.

Matricea este inversată faţă de axa Ox.

Pentru exemplul dat, matricea A are 2n+1 linii şi 2n coloane, iar conţinutul ei este cel alăturat.

Pentru construcţia matricei vom folosi doi vectori X şi Y. Vectorul X va reţine toate abscisele vârfurilor dreptunghiurilor. Vectorul Y va reţine toate ordonatele vârfurilor dreptunghiurilor.

Zonele elementare din desen corespund câte unui element A[i][j] din matrice şi sunt dreptunghiuri care au coordonatele vârfurilor opuse: (X[i],Y[j]) şi (X[i+1],Y[j+1]) i,j=1,2,…,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 conţinutul:

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 poziţia pe care apare a în vectorul sortat X
4. pb este poziţia pe care apare b în vectorul sortat Y
5. pc este poziţia pe care apare c în vectorul sortat X
6. pd este poziţia pe care apare d în vectorul sortat Y

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

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

Sursele `colajC.cpp` şi `colajP.pas` constituie implementarea a acestei soluţii.

O solutie care obţine un puntaj parţial constă în a contrui o matrice A cu p linii şi m coloane, elementele ei memorând valori `0` şi `1`.

Fiecărui dreptunghi cu vârful stânga-jos, respectiv dreapta-sus, de coordonate (xa,ya), respectiv (xb,yb), îi corespunde în matricea A o zonă în care toate elementele au valoarea 1:

```
for(l=ya;l<yb;l++)
 for(c=xa;c<xb;c++) a[l][c]=1;
```

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

Dezavantajul utilizării acestei metode constă în spaţiul de memorie disponibil, insuficient pentru memorarea matricei A. Nu întotdeauna este posibilă memorarea unei matrice cu `mxp( m,p<8000)` şi cu elemente de tip char/byte. Prin "comprimarea" dreptunghiurilor utilizând metoda împărţirii în "zone elementare", se reduce spaţiul necesar memorării matricei de tip char la cel mult 201x201 octeti.

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

O îmbunătăţire a punctajului se poate obţine atunci când valorile m şi p permit declararea unei matrice A de tip char care să memoreze toate zonele corespunzătoare dreptunghiurilor, prin eliminarea anumitor linii şi coloane. Necesitând un număr mare de operaţii datorate ştergerilor liniilor, respectiv coloanelor, identice din A se depăşeşte timpul de execuţie.

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

Analog, dacă coloanele $c_i,c_{i+1},....,c_k$ din A sunt identice, atunci se păstrează în A doar coloana c_i, restul se şterg 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 obţine `50p`).