

Descrierea soluției - defrag

*prof. Cheșcă Ciprian
Liceul Tehnologic "Costin Nenițescu" Buzău*

Pentru rezolvarea primei cerințe se determină câți clusteri sunt ocupați pe fiecare pistă. Se poate utiliza în acest scop o matrice având numărul de linii egal cu numărul de piste și numărul de coloane egal cu numărul de sectoare al platanului. La fiecare citire de cluster ocupat se setează în matrice poziția dată de pista și sectorul clusterului pe 1 (inițial fiind pe 0). În același timp se contorizează în primul element de pe fiecare linie numărul de clusteri ocupați. Prin parcurgerea elementelor de pe prima coloană a matricei se determină valoarea cerută la prima cerință, prin scădere față de numărul total de sectoare.

Pentru a doua cerință se procedează astfel:

Imaginându-ne că s-a realizat mutarea unor clusteri și s-a obținut o zonă contiguă, această zonă va avea lungimea egală cu numărul de clusteri ocupați pe fiecare pistă. Această zonă poate începe în orice poziție a pistei (de la 1 la ns), însă numai unele dintre aceste secvențe au un număr minim de clusteri liberi.

Așadar se preia numărul de clusteri ocupați pe fiecare linie, calculat anterior, pe care să-l notăm cu nco_i (număr clusteri ocupați pe pista i) și se analizează toate secvențele de câte nco_i elemente, începând cu poziția 1 până la o poziție în care zona se termină pe ultimul sector, și se determină numărul minim de clusteri de 0 pentru toate secvențele. Această valoare este valoarea cerută la cerința 2.

O soluție care implementează algoritmul descris mai sus obține aproximativ jumătate din punctaj, deoarece nu ține cont de circularitatea așezării clusterilor.

Pentru a rezolva și circularitatea așezării clusterilor se realizează o copie a matricii clusterilor pe „orizontală” prin dublarea numărului de sectoare. În acest mod se poate testa și dacă secvențele cu clusteri consecutivi pot începe chiar de pe ultimul sector, formând astfel o zonă contiguă cu clusteri aflați în primele sectoare.

Pentru optimizarea algoritmului descris anterior, se poate proceda astfel: se determină numărul de clusteri liberi din secvența care începe pe primul sector și are lungimea nco_i , apoi la fiecare nouă secvență de aceeași lungime mutată cu o poziție către dreapta (sau stânga), se adună sau se scade câte un element în funcție de ce se întâmplă la capete, deoarece celelalte elemente din secvență nu se modifică.

prof. Carmen Popescu
Colegiul Național "Gheorghe Lazăr" Sibiu

prof. Gorea Claudiu
Colegiul Național „Al. Papiu - Ilarian” Tg. Mureș

La citirea datelor se construiește o matrice cu p linii și s coloane cu $a[i][j]=1$ dacă clusterul j din pista i este ocupat $a[i][j]=0$ în caz contrar

În plus în $a[i][0]$ memorăm câte clustere sunt ocupate pe pista i ($a[i][0]$ = suma elementelor din linia i)

Răspunsul la punctul a) se obține simplu numărând pentru câte linii avem $a[i][0]=0$

Pentru punctul b) pentru fiecare linie practic după defragmentare vom obține o zonă compactă de lungime $k=a[i][0]$, nu ne interesează unde este aceasta poziționată. Această zonă este normal să se aleagă în

acea zona din pista unde sunt cele mai multe clustere ocupate. De aceea vom calcula sumele

$$\begin{aligned} b[1] &= a[i][1] + a[i][2] + \dots + a[i][k] \\ b[2] &= a[i][2] + a[i][3] + \dots + a[i][k+1] \\ &\dots \\ b[s-k+1] &= a[i][s-k+1] + a[i][s-k+2] + \dots + a[i][n] \end{aligned}$$

și

$$\begin{aligned} c[1] &= a[i][s-k+2] + a[i][s-k+3] + \dots + a[i][s] + a[i][1] \\ c[2] &= a[i][s-k+3] + a[i][s-k+4] + \dots + a[i][s] + a[i][1] + a[i][2] \\ &\dots \\ c[k-1] &= a[i][s] + a[i][1] + a[i][2] + \dots + a[i][k-1] \end{aligned}$$

Se observă că

$$b[j] = b[j-1] - a[i][j-1] + a[i][j+k-1]$$

pentru $j=1, 2, \dots, s-k+1$

și

$$\begin{aligned} c[1] &= b[s-k+1] - a[i][s-k+1] + a[i][1] \\ c[j] &= c[j-1] - a[i][s-k+j] + a[i][j] \\ &\text{pentru } j=2, 3, \dots, k-1 \end{aligned}$$

Se poate observa ușor că nu avem nevoie de vectorii b și c , ci doar de o variabilă simplă în care memorăm suma curentă și pe care o actualizăm la fiecare pas.

Răspunsul la a doua cerință este k - maximul dintre $\{b_1, b_2, \dots, b_{s-k+1}, c[1], c[2], \dots, c[k-1]\}$