

Problema miting – descrierea soluției

Autor: prof. Constantin Gălățan
C. N. „Liviu Rebreanu” Bistrița

Cerința 1. (20 puncte)

Se mențin patru variabile: **imin**, **jmin**, **imax**, **jmax** reprezentând coordonatele colțului stânga sus, respectiv colțul dreapta jos a dreptunghiului care include toate literele cuvântului **cuv**. Aceste variabile se actualizează odată cu citirea matricei de caractere. Aria maximă va fi:

$$A = (imax - imin + 1) * (jmax - jmin + 1)$$

Cerința 2. (80 de puncte)

Problema impune restricția ca două litere aflate în mașini diferite sau două secvențe de litere aflate în două mașini diferite să se poată reuni numai dacă toate literele aflate în cele două mașini se pot la rândul lor rearanja ca o secvență a cuvântului. Numerotăm zonele accesibile ale matricei astfel încât numerele de ordine **1 ... nc** să corespundă zonelor corespunzătoare pozițiilor inițiale ale literelor cuvântului, respectând ordinea din cuvânt. Pentru restul zonelor accesibile ordinea de numerotare nu este importantă.

Să presupunem că dorim să reunim în aceeași mașină, literele **cuv[i...j]**, unde **i** și **j** sunt poziții în cuvânt. Costul reunirii depinde și de zona **x** în care dorim să le aducem. Definim **c[i][j][x]** consumul minim pentru a unifica **cuv[i...j]** în zona cu numărul de ordine **x**. Fie **nc** numărul de caractere ale cuvântului. Secvența **cuv[i...j]** obținută într-o mașină în zona **x** provine prin reunirea literelor provenind din două mașini: prima având literele **cuv[i...k]**, iar a doua literele **cuv[k + 1...j]**. Consumul minim de combustibil **c[i][j][x]** necesar pentru ca două mașini să ajungă în mod **independent** în zona **x** (Fig 1) este:

$$c[i][j][x] = \min(c[i][k][x], c[k+1][j][x])$$
$$1 \leq i < nc, i < j, i < k < j, nc < x \leq n * n$$

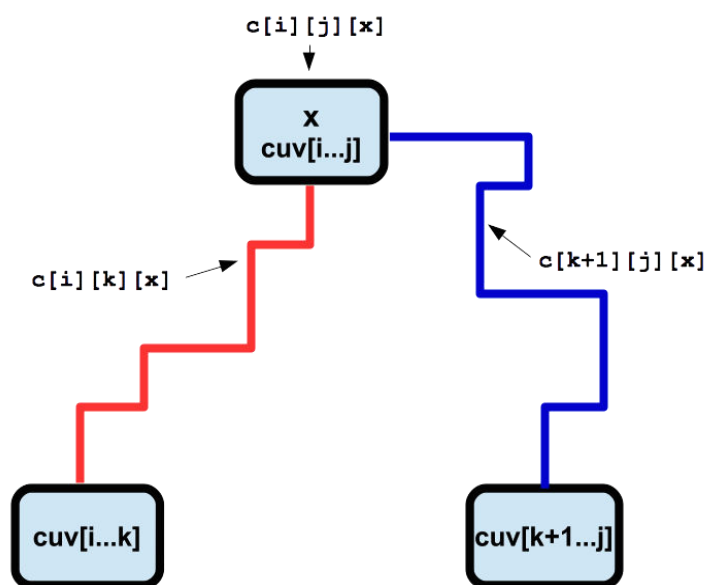


Fig. 1

Însă drumul minim al celor două mașini până în zona \mathbf{x} poate avea o porțiune comună începând cu zona \mathbf{y} . În acest caz se literele se reunifică în zona \mathbf{y} . Una dintre cele două mașini se oprește în \mathbf{y} , iar consumul transportului până în \mathbf{x} scade corespunzător consumului unei mașini pe drumul comun (Fig 2).

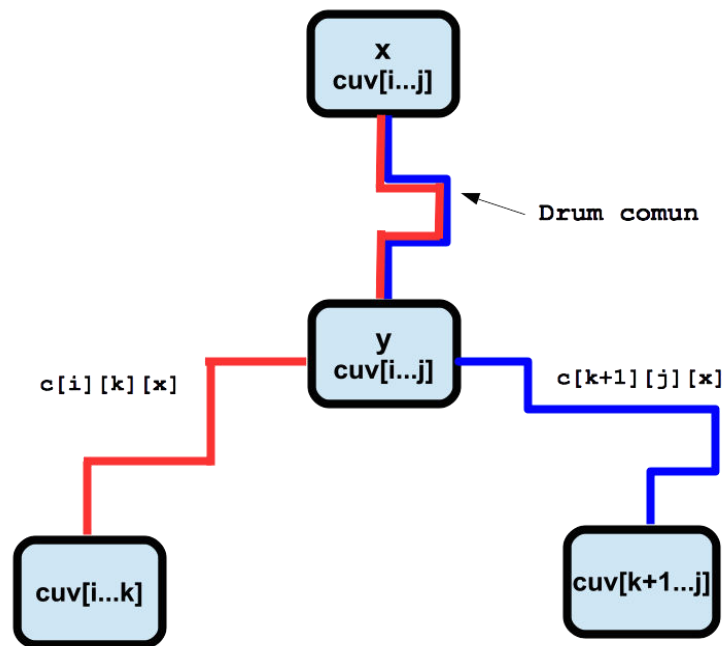


Fig. 2

Valoarea $c[i][j][x]$ se poate ajusta (eventual scade) rulând un algoritm de cost minim (Lee). Dacă \mathbf{y} e zonă vecină cu \mathbf{x} , atunci:

$$c[i][j][x] = \min(c[i][j][x], c[i][j][y] + 1)$$

În rezumat: mai întâi se determină consumurile minime pentru transportul fiecărei litere a cuvântului în fiecare celulă \mathbf{x} a matricei cu un algoritm de tip Lee. Pe baza acestor rezultate, se determină consumurile minime pentru a reuni câte două litere consecutive ale cuvântului în fiecare celulă \mathbf{x} a matricei. Urmează determinarea consumurilor minime pentru reunirea a trei litere consecutive folosind valori calculate pentru secvențe mai scurte, și așa mai departe. În final, se determină consumurile minime pentru a reuni toate literele cuvântului în fiecare celulă \mathbf{x} . Răspunsul este minimul acestor ultime valori.

Generarea tuturor secvențelor de lungimi $1, 2, \dots, nc$ necesită $nc \cdot (nc-1) / 2$ operații. Pentru fiecare asemenea operație, se aplică un algoritm de tip Lee, de complexitate $O(n^3)$. Complexitatea finală este $O(nc^2 \cdot n^3)$.