Autor: *prof.Constantin Gălățan*

Colegiul Național „*Liviu Rebreanu*” Bistrița

Fadema – Descrierea soluției

Dreptunghiul cu proprietatea din enunţ este o submatrice delimitată de două coloane **i** şi **j**, cu $1 \le i \le j \le m$ ale matricei date, pe care o numim **a**. Se va parcurge această submatrice căutând cea mai lungă secvenţă de linii alternante (valori **1** şi **0**). Pentru fiecare linie a acestei submatrice nu trebuie să existe nicio valoare identică cu valoarea de pe linia anterioară aflată pe aceeaşi coloană.

Complexitate $O(n^6)$ – 20 de puncte

Se generează în $O(n^4)$ toate submatricile cuprinse între liniile **i1** și **i2**, respectiv coloanele **j1** și **j2**. Fiecare asemenea submatrice se parcurge în $O(n^2)$ și se verifică validitatea sa.

Complexitate $O(n^4)$ – 40 de puncte

Se construieşte şirul **r[1]**, **r[2]**, ..., **r[n]** cu valori 0, 1 sau 2. **r[k]** $(1 \le k \le n)$ va fi 0 dacă **a[k][i]** este 0, **r[k] = 1** dacă **a[k][i] = 1** şi **r[k] = 2** dacă linia **k** a submatricei nu respectă condiţia de alternanţă. În continuare, se va căuta în şirul **r** cea mai lungă secvenţă alternantă. Fie **L** lungimea acesteia. Atunci numărul de elemente a submatricei alternante este **L * (j – i + 1)**. Soluţia va fi maximul acestor valori:

$$sol = \max(L * (j - i + 1)), \text{ cu } 1 \le i \le j \le m.$$

Complexitate $O(n^3)$ – 65 de puncte

Se procedează ca mai sus, doar că validitatea liniei **k** se verifică pe măsură ce se modifică lungimea liniei, adică odată cu creşterea valorii **j**. Aceasta duce la eliminarea unei parcurgeri de linie, iar complexitatea algoritmului scade cu un ordin de mărime.

Complexitate $O(n^2)$ – 90 puncte

Calculăm submatricea de dimensiune maximă ce s-ar putea obţine pornind spre dreapta şi în jos de la fiecare element al matricei şi respectă cerinţa din enunţ. Facem apoi acelaşi calcul spre sânga şi în jos.

Realizăm suma dimensiunilor celor două submatrici pentru fiecare element al matricei şi maximul dintre aceste sume este valoarea căutată.

Pentru a calcula submatricea de dimensiune maxima ce s-ar putea obţine pornind spre dreapta şi în jos de la fiecare element al matricei avem nevoie de:

a) Lungimea maximă **lSt[i][j]** a unui şir de elemente cu proprietatea din enunţ pornind de la fiecare element `a[i][j]` al matricei spre stânga. Aceste lungimi pot fi calculate în $O(n^2)$ parcurgân matricea de la dreapta spre stânga. Pentru fiecare element `a[i][j] != a[i][j+1]` avem `lSt[i][j]= lSt[i][j+1]+1` și *`lSt[i][j]=1`* pentru elementele cu `a[i][j] == a[i][j+1]` și pentru `j=`*m*.

b) Lungimea maximă `lJos[i][j]` a unui şir de elemente cu proprietatea din enunţ pornind de la fiecare element `a[i][j]` al matricei în jos. Aceste lungimi pot fi calculate în $O(n^2)$ parcurgând matricea de jos în sus analog punctului a).

c) Produsul dintre lungimea maximă `lJos[i][j]` și minimul lungimilor maxime `lSt` al elementelor care aparţin subșirului `lJos[i][j]` (adică minimul valorilor `lSt[k][j]` al elementrlor `a[k][j]` pentru `k=i, i+1, …,i+ lJos[i][j]`) reprezintă dimensiunea submatricei. Aceste minime pot fi calculate în $O(n^2)$ pentru toată matricea parcurgând-o de jos în sus și păstrând minimul dintre `lSt[i][j]`și `lSt[i][j+1]` pentru toate elementele `a[i][j] != a[i][j+1]` iar pentru elementele cu `a[i][j] == a[i][j+1]` și pentru `j=m` păstram `lSt[i][j]`.

Analog se procedează și pentru submatricea de dimensiune maxima ce s-ar putea obţine pornind spre stânga şi în jos de la fiecare element al matricei.

Observație: la punctajele de mai sus se adaugă 10 puncte din oficiu