**Tema 7 - Progtamare Dinamică**

### 1. Problema înmulţirii matricelor: Având un şir de matrice *A*1, *A*2, …, *An*, cu dimensiunile p1×p2, p2×p3, ..., pn×pn+1, care trebuie înmulţite, în ce ordine efectuăm înmulţirile astfel încât să avem un număr minim de operaţii? Adică, un exemplu de parantezare a șirului de matrice, astfel încât rezultatul înmulțirii să fie optimal. (4p)

**Observaţie**: Având matricele *A*1 şi *A*2 cu dimensiunile *n*×*m* respectiv *m*×*p*, numărul de înmulţiri efectuate este: *n\*m\*p*:

pentru i = 1,n

pentru j = 1,p

*rezultat(i,j) = *

Se consideră costul înmulţirii matricelor egal cu nr. de înmulţiri efectuate = *n\*m\*p*.

Deci pentru matricele *A*1*, A*2*, A*3 cu dimensiunile: 10×100, 100×5, 5×50, atunci:

- dacă se efectuează (*A*1\*(*A*2\**A*3)), se obţine costul total:

(100\*5\*50)+(10\*100\*50)=25000+50000 = 75000

- dacă se efectuează (*A*1\**A*2)\**A*3)) se obţine costul: (10\*100\*5)+(10\*5\*50)= 5000+2500= 7500

Se observă că, în al doilea caz se obţine un cost mult mai mic.

Se pune problema cum se parantezează şirul *A*1, *A*2, …, *An* pentru a obţine un cost minim.

**Rezolvare cu metoda programării dinamice**:

Notaţie: vom nota cu *Aij* (*i* ≤ *j*) matricea obţinută din produsul: *Ai*\**Ai*+1\*…\**Aj*

(1) Caracterizarea soluţiei optime: dacă *A*1*n* este obţinut printr-o parantezare optimă, avem un *k*, 1≤*k*≤*n* şi *A*1*n* = *A*1*k\*Ak*+1*n*, dar *A*1*k* este obţinut printr-o parantezare oprimă pentru produsul *A*1\**A*2\*…\**Ak* şi *Ak*+1*n* este obţinut printr-o parantezare oprimă pentru *Ak*+1\*…\**An*.

(Altfel s-ar obţine o contradicţie privind costul minim al produsului final)

Deci o soluţie optimă pentru problema curentă conţine soluţii optime pentru subproblemele care o compun.

(2) Determinarea unei relaţii de recurenţă:

Vom utiliza matricea *m*, care pe poziţia *m*(*i,j*) va conţine costul minim al unei parantezări pentru *Aij* = *Ai*\**Ai*+1\*…\**Aj*, de dimensiune *pi* × *pj*+1 unde (*i* < *j*), *m*(*i,i*)=0 pentru orice *i*, 1 ≤*i* ≤*n*. (nu ne interesează decât partea de deasupra diagonalei principale!)

*m(i,j)* =

unde

- dimensiunile matricei *Ai* sunt *pi*× *pi*+1.

- *Ai*\**Ai*+1\*…\**Aj* s-a obţinut din *Aik*\**Ak*+1*j*

Deci valorile *m*(*i,j*) reprezintă costul soluţiilor optime pentru subprobleme, *m*(1,*n*) este valoarea optimă pentru costul produsului *A*1*A*2.....*An*.

Vom folosi şi o matrice *s*, unde *s*(*i*,*j*) = *k* (*k* acel indice pentru care *m*(*i*,*j*) după definiţia de mai sus este optim). Această matrice ne ajută să construim efectiv o soluţie optimă, după ce am calculat valoarea soluţiei optime.

Costul minim se va afla în *m*(1,*n*).

(3) Algoritmul de tip „bottom-up” pentru completarea matricelor *m* şi *s*:

Vom calcula pas cu pas *m*(*i*,*j*), întâi pentru valori vecine: exemplu consider *n* = 4 și p1=14, p2=7, p3=6, p4=11, p5=9. Atunci:

- întâi iniţializez *m*(*i*,*i*)=0, pentru orice *i*, 1≤ *i* ≤*n*

- calculez apoi:

*m*(1,2) = *p*1\**p*2\**p*3=588, *m*(2,3) = *p*2\**p*3\**p*4=462, *m*(3,4) = *p*3\**p*4\**p*5=594,

*s*(1,2) = 1, *s*(2,3) =2, *s*(3,4) = 3

*m*==

- calculez apoi:

*m*(1,3) = min{(*m*(2,3)+ *p*1\**p*2\**p*4), (*m*(1,2)+ *p*1\**p*3\**p*4)}=min{1533, 1512}=1512, s(1,3)=2,

*m*(2,4) = min{(*m*(3,4)+ *p*2\**p*3\**p*5), (*m*(2,3)+ *p*2\**p*4\**p*5)}= min{972, 1155}=972, s(2,4)=2,

*m*==

- calculez apoi:

*m*(1,4) = min{(*m*(2,4)+ *p*1\**p*2\**p*5), (*m*(1,2)+*m*(3,4)+ *p*1\**p*3\**p*5), (*m*(1,3)+*p*1\**p*4\**p*5}=)}= min{1854, 1938,2898}=1854, s(1,4)=1,

*m*==

s=

Parantezarea optimală: (A1\*(A2\*(A3\*A4)))

Intrare: Numărul de matrice și dimensiunile acestora

Ieșire: Șirul de matrice parantezat.

2. **Problema subșirului crescător de lungime maximală:** Se consideră un şir (vector) de elemente *v* = (*v*(*i*), *i* = 0,*n*-1), să se determine un subşir crescător de lungime maximală. (2p)

**Exemplu**: *v* = {6, 2, 3, 1, 8, 9, 7, 8, 10}, *n*=9

Rezolvare: Un subșir crescător de lungime maximală este 2, 3, 8, 9, 10

Intrare: vectorul v

Ieșire: lungimea maximă a unui subșir crescător și un exemplu de astfel de subșir

### 3. Problema triunghiului: Se consideră un triunghi de numere naturale (o matrice subdiagonală) format din *n* linii (pe prima linie se află un singur element, pe a doua două, …, pe ultima am *n* elemente). Cu elementele triunghiului se formează sume astfel:

- se pornește cu elementul de pe prima linie

- la pasul *k* se adaug elementul de pe linia *k*, care se află sub elementul ales anterior (pe linia *k*-1) sau elementul plasat diagonal la dreapta sub elementul anterior (adică dacă la pasul k-1 s-a les *a*[*k*-1][*j*], la pasul k se alege a[*k*][*j*] sau *a*[*k*][*j*+1])

Problema care se pune este de a obţine cea mai mare sumă în acest fel. (1p)

**Exemplu**:

1

3 5

2 4 3

5 1 6 3

Sumele pe care le pot construi:

*s*1 = 1 + 3 + 2 + 5 = 11

*s*2 = 1 + 3 + 2 + 1 = 7

*s*3 = 1 + 3 + 4 + 1 = 8

*s*4 = 1 + 3 + 4 + 6 = 14

*s*5 = 1 + 5 + 4 + 1 = 11

*s*6 = 1 + 5 + 4 + 6 = 16

*s*7 = 1 + 5 + 3 + 6 = 15

*s*8 = 1 + 5 + 3 + 3 = 12

şi suma maximă este 16.

Intrare: matricea A

Ieșire: Suma maximă și un exemplu de soluție

4. **Problema sumei în dreptunghi**: Se consideră un dreptunghi de numere, reprezentat printr-o matrice A de dimensiuni m×n. Se construiește o sumă în dreptunghi în modul următor. Se pornește din colțul stânga sus. Dacă la pasul k-1 s-a adunat la sumă elementul A[k-1][j], la următorul pas se adaugă unul dintre elementele A[k-1][j+1], A[k][j] sau A[k][j+1], adică elementul de la drepata lui A[k-1][j], cel de dedesubt sau cel de la dreapta celui de dedesubt. În final trebuie obținută ce mai mare sumă posibilă. (2p)

**Exemplu**: Se consideră dreptunghiul

3 11 4 25 6

12 6 15 9 31

3 21 1 10 2

5 19 32 3 9

Suma maximă posibilă este 105 și se poate obține în modul următor: 3 + 12 + 6 +21 + 19 + 32 + 3 + 9.

1. Intrare: matricea A
2. Ieșire: Suma maximă și un exemplu de soluție

### 5. Subşir comun maximal: Având două şiruri *X* = (*x*1, *x*2, …, *xm*) şi *Y* = (*y*1, *y*2, …, *yn*) să se determine un subşir comun de lungime maximă. (2p)

**Exemplu:**  Se consideră cele două șiruri X și Y

*X* = (2, 5, 5, 6, 3, 1, 7, 4, 3)

*Y* = (4, 5, 2, 3, 7, 9, 3, 2, 7, 3)

O soluţie posibilă este: *Z* = (2, 3, 7, 3)

Intrare: vectorii X și Y

Ieșire: lungimea maximă a unui subșir comun și un exemplu de astfel de subșir.

6. **Algoritmul Floyd-Warshall** de determinare a celui mai scurt drum între oricare două noduri ale unui graf: Se dă un graf neorientat (sau orientat) reprezentat printr-o matrice de adiacență. Să se determine cele mai scurte drumuri între oricare două noduri. (4p)

Se consideră un graf G=(V, Γ), în care V este mulţimea vârfurilor şi Γ mulţimea arcelor dintre acestea. Se numerotează fiecare vârf din V cu un număr de la 1 la N, N=|V|. Pentru oricare perche de vârfuri (i, j) notăm prin c(i, j) costul muchiei de la i la j. c(i, j) = ∞ dacă nu există muchie între i şi j.

Notăm prin drumMinim(i,j,k) cel mai scurt drum între i şi j care utilizează doar vârfuri intermediare din mulţimea {1,2,...,k}.

Dacă există un drum mai scurt de la i la j, care trece prin vârful k+1, atunci acesta se obţine concatenând cel mai scurt drum de la i la k+1, care are vârfuri intermediare din {1,2,...,k} cu cel mai scurt drum de la k+1 la j care are vârfuri intermediare în {1,2,...,k}. Astfel de poate deduce relaţia de recurenţă



**Determinarea drumului de lungime minimă:**

Se realizează utilizând o matrice *predecesor*(i,j), care iniţial este i pentru vârfuri adiacente şi ∞ în rest.

*predecesor*(i,j) = nodul predecesor lui j pe drumul de lungime minimă de la i la j

Pe parcursul calculului iterativ al distanţei minime între (i,j) se actualizează

*predecesor*(i,j)=*predecesor*(k,j) cu indicele k din formula de recurenţă.

***Exemplu***:

graf2

,

, 

, 

, 

, 

, 

Cele mai scurte drumuri:

Cale\_minima(1,2) = 1 2

Cale\_minima(1,3) = 1 3

Cale\_minima(1,4) = 1 2 4

Cale\_minima(2,3) = 2 4 3

Cale\_minima(2,4) = 2 4

Cale\_minima(3,4) = 3 4

Intrare: Matricea de adiacență.

Ieșire: Pentru oricare două noduri lungimea celui mai scurt drum între ele și un exemplu de astfel de drum.

7. ***Problema palindroamelor:*** Se consideră un şir X = (x0,x1,...,xn-1) de lungime n. Să se determine numărul minim de palindroame în care poate fi împărţit acest şir. (4p)

***Exemplu:***

X=anaban numărul minim de palindroame: 2 a\_naban

X= abaccbcb numărul minim de palindroame: 3 aba\_cc\_bcb

Intrare: șirul de caractere X

Ieșire: numărul minim de palindroame și un exemplu de astfel de palindroame.