Metoda Programării Dinamice

-3-

Problema discretă a rucsacului

Se consideră un rucsac de capacitate (greutate) maximă G (**număr** natural) şi n obiecte caracterizate prin:

- greutăţile lor (**numere naturale**) $g_1,\ldots,g_n$;
- câştigurile $v_1,\ldots,v_n$ obţinute la încărcarea lor în totalitate în rucsac.

Un obiect **nu poate fi fracționat**.

Se cere o modalitate de încărcare de obiecte în rucsac, astfel încât câştigul total să fie maxim.

Problema discretă a rucsacului

Caz particular

Date n obiecte cu ponderile w_1, w_2,…, w_n și o limită W, să se selecteze o submulțime de obiecte cu suma ponderilor maximă, fără a depăși însă ponderea W

Problema discretă a rucsacului

Caz particular

Date n obiecte cu ponderile w_1, w_2,..., w_n și o limită W, să se selecteze o submulțime de obiecte cu suma ponderilor maximă, fără a depăși însă ponderea W

Interpretări

- Submulțime de sumă maximă mai mică sau egală cu o valoare M dată (v. Greedy)
- n activități cu duratele w_1, w_2, ...,w_n necesită o resursă. Știind că timpul maxim de funcționare a resursei este W, să se selecteze o submulțime de activități care țin resursa ocupată un timp cât mai lung (maxim)

Problema discretă a rucsacului

Exemplu:

```
G = 8
n = 4 obiecte
g:  3  4   4   6
v:  3  9  10  18
```

Problema discretă a rucsacului

Exemplu:

```
G = 8
n = 4 obiecte
g:  3  4   4   6
v:  3  9  10  18
```

Greedy – în ordinea descrescătoare a raportului v/g

- Alege întâi obiectul 4 de greutate 6
- Nu se mai poate pune nici un alt obiect întreg în rucsac
- Câştigul Greedy: **18**

Problema discretă a rucsacului

Exemplu:

```
G = 8
n = 4 obiecte
g:  3  4   4   6
v:  3  9  10  18
```

Greedy – în ordinea descrescătoare a raportului v/g

- Alege întâi obiectul 4 de greutate 6
- Nu se mai poate pune nici un alt obiect întreg în rucsac
- Câştigul Greedy: **18**

Soluţia optimă:

- Alegem obiectele 2 şi 3
- Câştigul total 10 + 9 = **19**

Problema discretă a rucsacului

- **Principiu de optimalitate**

  Dacă **S** este soluție optimă pentru greutatea **g** și obiectele **{1,2,…,n} care**

  - **conţine n,**

  - **nu conţine n**

Problema discretă a rucsacului

- **Principiu de optimalitate**

  Dacă **S** este soluție optimă pentru greutatea **g** și obiectele **{1,2,...,n} care**

  - **conţine n** atunci **S – {n}** este soluție optimă pentru

    greutatea $\mathbf{g - g_n}$ și

    obiectele **{1,2,..., n−1}**

  - **nu conţine n** atunci **S** este soluție optimă pentru

    greutatea **g** și

    obiectele **{1,2,..., n−1}**

Problema discretă a rucsacului

- Subproblemă
- Soluție
- Știm direct
- Relație de recurență
- Ordinea de calcul
- Afișarea obiectelor din soluția optima

Problema discretă a rucsacului

```
for(int g = 0; g<= G;g++)  c[0][g]= 0;

for(int i = 1; i<= n;i++){
        c[i][0]=0;
        for(int gr = 1; gr <= G; gr++){
                if (g[i]<= gr)







                  else
                         c[i][gr]=c[i-1][gr];
        }
}
cout<<"Castigul total " << c[n][G];
```

Problema discretă a rucsacului

```
for(int g = 0; g<= G;g++)  c[0][g]= 0;

for(int i = 1; i<= n;i++){
      c[i][0]=0;
      for(int gr = 1; gr <= G; gr++){
            if (g[i]<= gr)
                  if (v[i]+c[i-1][gr-g[i]]>c[i-1][gr])
                        c[i][gr]=v[i]+c[i-1][gr-g[i]];
                  else
                        c[i][gr]=c[i-1][gr];
            else
                  c[i][gr]=c[i-1][gr];
      }
}
cout<<"Castigul total " << c[n][G];
```

Problema discretă a rucsacului

- **Afişarea obiectelor**
  - din relația de recurență

Problema discretă a rucsacului

Afişarea obiectelor – recursiv

```
void afis(int i,int gr){
    if(i==0 || gr==0)
        return;
    if((g[i]<=gr)&&(c[i][gr]==v[i]+c[i-1][gr-g[i]])){
        afis(i-1,gr-g[i]);
        cout<<i<<" "<<endl;
    }
    else
        afis(i-1,gr);
}
```

`afis(n,G);`

Problema discretă a rucsacului

Afişarea obiectelor – nerecursiv

```
gr = G;
i = n;
while(gr>0 && i>0){
    if((g[i]<=gr)&&(c[i][gr]==v[i]+c[i-1][gr-g[i]])){
                cout<<i<<" ";
                gr=gr-g[i];
    }
    i--;
}
```

Problema discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g]=\begin{cases} c[i-1][g], \text{ daca } g_i>g \\ \max\{v_i+c[i-1][g-g_i], c[i-1][g]\}, \text{ altfel} \end{cases}$$

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0								
0								
0								
0								

Problema discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g]=\begin{cases} c[i-1][g], \text{ daca } g_i>g \\ \max\{v_i+c[i-1][g-g_i], c[i-1][g]\}, \text{ altfel} \end{cases}$$

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3					
0								
0								
0								

Problema discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g]=\begin{cases} c[i-1][g], \text{ daca } g_i>g \\ \max\{v_i+c[i-1][g-g_i],c[i-1][g]\}, \text{ altfel} \end{cases}$$

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3
0								
0								
0								

Problema discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g]=\begin{cases} c[i-1][g], \text{ daca } g_i>g \\ \max\{v_i+c[i-1][g-g_i], c[i-1][g]\}, \text{ altfel} \end{cases}$$

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3
0	0	0	3	9				
0								
0								

c[2][4] = soluția optimă pentru obiectele {1,2} și g=4

c[2][4] =max{v_2 + c[1][0], c[1][4] } = max{9 +0, 3 } = 9

Problema discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g] = \begin{cases} c[i-1][g], \text{ daca } g_i > g \\ \max\{v_i + c[i-1][g-g_i], c[i-1][g]\}, \text{ altfel} \end{cases}$$

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3
0	0	0	3	9	9	9	12	
0								
0								

Problema discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g] = \begin{cases} c[i-1][g], \text{ daca } g_i > g \\ \max\{v_i + c[i-1][g-g_i], c[i-1][g]\}, \text{ altfel} \end{cases}$$

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3
0	0	0	3	9	9	9	12	12
0	0	0	3	10				
0								

Problema discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g] = \begin{cases} c[i-1][g], \text{ daca } g_i > g \\ \max\{v_i + c[i-1][g-g_i], c[i-1][g]\}, \text{ altfel} \end{cases}$$

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3
0	0	0	3	9	9	9	12	12
0	0	0	3	10	10	10	?	
0								

Problemа discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g] = \begin{cases} c[i-1][g], \text{ daca } g_i > g \\ \max\{v_i + c[i-1][g-g_i], c[i-1][g]\}, \text{ altfel} \end{cases}$$

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3
0	0	0	3	9	9	9	12	12
0	0	0	3	10	10	10	13	
0								

Problema discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g] = \begin{cases} c[i-1][g], \text{ daca } g_i > g \\ \max\{v_i + c[i-1][g-g_i], c[i-1][g]\}, \text{ altfel} \end{cases}$$

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3
0	0	0	3	9	9	9	12	12
0	0	0	3	10	10	10	13	19
0	0	0	3	10	10	18	18	19

câștig optim c[4][8]

Problema discretă a rucsacului

- **Exemplu** G = 8, n = 4 obiecte

g: 3 4 4 6

v: 3 9 10 18

$$c[i][g]=\begin{cases} c[i-1][g], \text{ daca } g_i>g \\ \max\{v_i+c[i-1][g-g_i], c[i-1][g]\}, \text{ altfel} \end{cases}$$

Soluție:

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3
0	0	0	3	9	9	9	12	12
0	0	0	3	10	10	10	13	19
0	0	0	3	10	10	18	18	19

Problemă discretă a rucsacului

- **Exemplu** `G = 8, n = 4 obiecte`

```
g:   3   4    4    6
v:   3   9   10   18
```

$$c[i][g] = \begin{cases} c[i-1][g], & \text{daca } g_i > g \\ \max\{v_i + c[i-1][g-g_i], c[i-1][g]\}, & \text{altfel} \end{cases}$$

Soluție:

c:

0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3
0	0	0	3	9	9	9	12	12
0	0	0	3	10	10	10	13	19
0	0	0	3	10	10	18	18	19

La pasul i obiectul i a fost luat ⇔ c[i][g] > c[i−1][g]

Problema discretă a rucsacului

- O(nG)

Distanțe de editare. Alinierea secvențelor

- **Putem măsura similaritatea între secvențe (ADN) prin**
  - **Elemente comune** – cel mai lung subșir comun pentru două secvențe
  - **Distanțe de editare** – numărul minim de inserări și modificări (eventual și ștergeri) de caractere necesar pentru transforma prima secvență în cea de a doua

\+ aplicații în procese de căutare de cuvinte – sugestii de cuvinte similare

Alinierea secvențelor

Exemplu

Aliniere – punerea poziţiilor (caracterelor) din cele două secvenţe a şi b în corespondenţă 1 la 1, cu posibilitatea de a insera spaţii (păstrând ordinea literelor)

```
a = AGGGCT      b = AGGCA

AGGGCT
AGG-CA
```

penalizarea = penalizarea spațiului + penalizarea pentru diferența T/A

Alinierea secvențelor

Exemplu

Aliniere – punerea poziţiilor (caracterelor) din cele două secvenţe a şi b în corespondenţă 1 la 1, cu posibilitatea de a insera spaţii (păstrând ordinea literelor)

```
a = AGGGCT      b = AGGCA

AGGGCT
AGG-CA
```

penalizarea = penalizarea spațiului + penalizarea pentru diferența T/A

sau

```
AGGGCT-
AGG-C-A
```

penalizarea = 3*penalizarea spațiului

Alinierea secvențelor

Date două secvențe, $\mathbf{X} = \mathbf{x}_1\mathbf{x}_2\ldots\mathbf{x}_n$ și $\mathbf{Y} = \mathbf{y}_1\mathbf{y}_2\ldots\mathbf{y}_m$ aliniem secvențele inserând în ele caracterul ‘ ‘ astfel încât secvențele să devină de aceeași lungime și penalizând pozițiile pe care diferă secvențele obținute.

- **Formulare echivalentă**:

Aliniere = formarea de perechi (x_i,y_j) astfel încât fiecare caracter apare în cel mult o pereche şi nu există perechi încrucișate:

– dacă avem perechile (x_i,y_j) şi (x_k,y_t) şi $i<k \Rightarrow j < t$

AGGGCT

AGG-CA

Alinierea secvențelor

Scorul (penalizarea) alinierii = suma penalizărilor alinierilor de caractere diferite şi alinierilor caracter–spaţiu (**scorul Needleman–Wunsch**).

- Penalizări diferite pentru diferențe de litere, spaţiu (de exemplu diferența A–G poate fi mai gravă decât A–T)
- Notaţii:
  - p_{spatiu}
  - p_{XY} – penalizarea alinierii caracterului X cu caracterul Y

Alinierea secvențelor

- ADN – alfabet A,C,G,T
- **Asemănări ADN** – poate semnifica apropiere în arborele genealogic
- **Esențial să fie rapizi** – se aplică pentru volum mare de date

Alinierea secvențelor

- **Principiu de optimalitate:**

  $\mathbf{x_1x_2...x_n}$

  $\mathbf{y_1y_2...y_m}$

aliniere cu penalizare minimă

Evidenţiem ultima operație din aliniere

Cazuri

Alinierea secvențelor

- **Principiu de optimalitate:**

$\mathbf{x}_1\mathbf{x}_2\ldots\mathbf{x}_n$

$\mathbf{y}_1\mathbf{y}_2\cdots\mathbf{y}_m$

aliniere cu penalizare minimă

Evidenţiem ultima pereche din aliniere

<u>Cazuri</u>

- $\mathbf{x}_n$ aliniat cu $\mathbf{y}_m$
- $\mathbf{x}_n$ aliniat cu spaţiu
- $\mathbf{y}_m$ aliniat cu spaţiu

Alinierea secvențelor

- Subprobleme
- Știm direct
- Relație de recurență
- Ordinea de calcul
- Determinarea unei soluții

Alinierea secvențelor

- **Exemplu** $p_{spaţiu} = 2,$

$p_{AC} = p_{GT} = 1$

$p_{XY} = 3$ pentru $X \neq Y$ în rest

```
GATC   ->   G-ATC
TCAG        TCAG-   scor 6
```

c:

0	1	2	3	4

Alinierea secvențelor

- **Exemplu** $p_{spaţiu} = 2$,

$p_{AC} = p_{GT} = 1$

$p_{XY} = 3$ pentru $X \neq Y$ în rest

```
GATC    ->    G-ATC
TCAG          TCAG-    scor 6
```

c:

0	1	2	3	4
0	**2**	**4**	**6**	**8**
2				
4				
6				
8				

$j * p_{spaţiu}$

Alinierea secvențelor

- **Exemplu** $p_{spaţiu} = 2,$

  $p_{AC} = p_{GT} = 1$

  $p_{XY} = 3$ pentru $X \neq Y$ în rest

```
GATC   ->   G-ATC
TCAG        TCAG-   scor 6
```

		0	1	2	3	4	
		0	**2**	**4**	**6**	**8**	**G**
c:		**2**	**1**				**T**
		4					
		6					
		8					

Alinierea secvențelor

- **Exemplu**

$p_{spaţiu} = 2,$

$p_{AC} = p_{GT} = 1$

$p_{XY} = 3$ pentru $X \neq Y$ în rest

```
GATC   ->   G-ATC
TCAG        TCAG-    scor 6
```

c:

	0	1	2	3	4
	0	2	4	6	8
	2	1	3		
	4				
	6				
	8				

G

TC

```
aliniem G cu C, rămâne de aliniat secvenţa vidă cu T cu cost c[0][1]
aliniem - cu C, rămâne de aliniat secvenţa G cu T cu cost c[1][1]
aliniem G cu -, rămâne de aliniat secvenţa vidă cu TC cu cost c[0][2]
```

Alinierea secvențelor

- **Exemplu**

$p_{spaţiu} = 2,$

$p_{AC} = p_{GT} = 1$

$p_{XY} = 3$ pentru $X \neq Y$ în rest

```
GATC    ->    G-ATC
TCAG          TCAG-    scor 6
```

c:

0	1	2	3	4
0	2	4	6	8
2	1	3	5	
4				
6				
8				

G

TCA

aliniem G cu A, rămâne de aliniat secvenţa vidă cu TC cu cost c[0][2]

aliniem - cu A, rămâne de aliniat secvenţa G cu TC cu cost c[1][2]

aliniem G cu -, rămâne de aliniat secvenţa vidă cu TCA cu cost c[0][3]

Alinierea secvențelor

- **Exemplu**

$p_{spaţiu} = 2,$

$p_{AC} = p_{GT} = 1$

$p_{XY} = 3$ pentru $X \neq Y$ în rest

```
GATC    ->    G-ATC
TCAG          TCAG-    scor 6
```

c:

0	1	2	3	4
0	2	4	6	8
2	1	3	5	6
4	3	2	3	5
6	4	4	5	4
8	6	4	5	?

Alinierea secvențelor

- **Exemplu** $p_{spaţiu} = 2$,

  $p_{AC} = p_{GT} = 1$

  $p_{XY} = 3$ pentru $X \neq Y$ în rest

```
GATC    ->    G-ATC
TCAG          TCAG-    scor 6
```

c:

0	1	2	3	4
0	2	4	6	8
2	1	3	5	6
4	3	2	3	5
6	4	4	5	4
8	6	4	5	6

Alinierea secvențelor

- **Exemplu**

$p_{spaţiu} = 2,$

$p_{AC} = p_{GT} = 1$

$p_{XY} = 3$ pentru $X \neq Y$ în rest

```
GATC   ->   G-ATC
TCAG        TCAG-   scor 6
```

Soluţia:

c:

0	1	2	3	4
0	2	4	6	8
2	1	3	5	6
4	3	2	3	5
6	4	4	5	4
8	6	4	5	6

Alinierea secvențelor

- **Exemplu**

$p_{spaţiu} = 2,$

$p_{AC} = p_{GT} = 1$

$p_{XY} = 3$ pentru $X \neq Y$ în rest

```
GATC    ->    G-ATC
TCAG          TCAG-    scor 6
```

Soluţia:

C:

0	1	2	3	4	
0	2	4	6	8	↑ aliniere GT
2	1	3	5	6	↑ aliniere -C
4	3	2	3	5	↑ aliniere AA
6	4	4	5	4	↑ aliniere TG
8	6	4	5	6	↑ aliniere C-

Distanţa de ediare – Levenstein

- **Similar (generalizare)**

```
carte

antet
```

- **Laborator**

Probleme de numărare

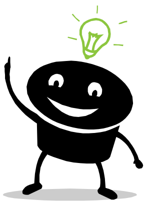
Probleme de numărare

- **Numărul de şiruri binare de lungime n care nu conţin două valori egale cu 1 pe poziţii consecutive**

```
000
001
010
100
101
```

Probleme de numărare

- **Numărul de şiruri binare de lungime n care nu conţin două valori egale cu 1 pe poziţii consecutive**



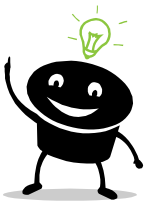
Analizăm structura unui şir soluţie evidenţiind primul element

- Începe cu 0

- Începe cu 1

Probleme de numărare

- **Numărul de şiruri binare de lungime n care nu conţin două valori egale cu 1 pe poziţii consecutive**



Analizăm structura unui şir soluţie evidenţiind primul element

- **Începe cu 0** – poate continua cu orice şir binar valid de lungime n-1
- **Începe cu 1** – poate continua cu orice şir binar valid de lungime n-1 care începe cu 0

Probleme de numărare

- **Numărul de şiruri binare de lungime n care nu conţin două valori egale cu 1 pe poziţii consecutive**

- **Subprobleme**
  - Nr[0][i] – numărul de şiruri binare valide care încep cu 0
  - Nr[1][i] – numărul de şiruri binare valide care încep cu 1
- **Recurenţe**

- **Soluţie**

Probleme de numărare

- Numărul de şiruri de lungime n peste alfabetul {1,2,3} care respectă constrângerile:

  – orice 1 are pe poziţiile alăturate în stânga şi dreapta valoarea 3

  – orice 2 are pe cel puțin una dintre poziţiile din stânga valoarea 3

```
3132331332 – DA
3132231332 – NU
 132331332 – NU
```

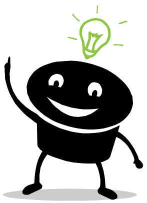
Temă

Probleme de numărare

- Numărul permutări ale mulţimii {1,2,..., n} care au exact k inversiuni (n, k date)

Probleme de numărare

- **Numărul permutări ale mulţimii {1,2,..., n} care au exact k inversiuni (n, k date)**



Analizăm structura unei permutări soluţie $x_1x_2...x_n$ evidenţiind ultimul element x_n

- n
- n-1
- n-2
- ...
- 1

Probleme de numărare

- **Numărul permutări ale mulţimii {1,2,..., n} care au exact k inversiuni (n, k date)**

Analizăm structura unei permutări soluţie $x_1x_2...x_n$ evidenţiind ultimul element x_n

- **n** – în $x_1x_2...x_{n-1}$ sunt k inversiuni, deoarece nu există inversiune de forma (x,n)

Probleme de numărare

- **Numărul permutări ale mulțimii {1,2,..., n} care au exact k inversiuni (n, k date)**

  Analizăm structura unei permutări soluţie $x_1x_2...x_n$ evidențiind ultimul element x_n

  - **n** – în $x_1x_2...x_{n-1}$ sunt k inversiuni, deoarece nu există inversiune de forma (x, n)
  - **n-1** – în $x_1x_2...x_{n-1}$ sunt k-1 inversiuni, deoarece unica inversiune determinată de x_n = n-1 este (n, n-1)
  - ... **n-k...1**

Unde ne oprim?

Probleme de numărare

- **Numărul permutări ale mulțimii {1,2,..., n} care au exact k inversiuni (n, k date)**

Analizăm structura unei permutări soluţie $x_1x_2...x_n$ evidențiind ultimul element x_n

- **n** – în $x_1x_2...x_{n-1}$ sunt k inversiuni, deoarece nu există inversiune de forma (x, n)
- **n-1** – în $x_1x_2...x_{n-1}$ sunt k-1 inversiuni, deoarece unica inversiune determinată de $x_n = n-1$ este (n, n-1)
- ... **n-k...1**

Unde ne oprim? $\Rightarrow$ cazuri $k<n$, $k \geq n$

Probleme de numărare

- **Numărul permutări ale mulțimii {1,2,..., n} care au exact k inversiuni (n, k date)**

Dacă k<n atunci

$$nr[n][k] = \overset{x_n=n}{nr[n-1][k]} + \overset{x_n=n-1}{nr[n-1][k-1]} + \ldots + \overset{x_n=n-k}{nr[n-1][0]} =$$

$$= nr[n-1][k] + nr[n][k-1]$$

Dacă n≤ k ≤ n(n−1)/2 atunci

$$nr[n][k] = \overset{x_n=n}{nr[n-1][k]} + \overset{x_n=n-1}{nr[n-1][k-1]} + \ldots + \overset{x_n=1}{nr[n-1][k-n+1]} =$$

$$= nr[n-1][k] + nr[n][k-1] - nr[n-1][k-n]$$

Altfel nr[n][k]=0

Probleme de numărare

- **Numărul permutări ale mulţimii $\{1,2,\ldots, n\}$ care au exact k inversiuni (n, k date)**

  - Subprobleme
    - Nr[i][t] – numărul de permutări cu i elemente având t inversiuni, $i \leq n$, $t \leq k$
  - Recurențe

    Nr[i][t] = Nr[i-1][t] + Nr[i-1][t-1]+…+Nr[i-1][max{t-i+1,0}]

    $t<=i(i-1)/2$
  - Soluţie Nr[n][k]

Probleme de numărare

- **Numărul permutări ale mulţimii {1,2,..., n} care au exact k inversiuni (n, k date)**

```
for(i=0;i<=n;i++)
         for(t=0;t<=k;t++)
                  nr[i][t]=0;
for(i=0;i<=n;i++) nr[i][0]=1;
for(i=1;i<=n;i++)
        for(t=1;t<=k;t++)
             if(t<i)
                  nr[i][t]=nr[i-1][t]+nr[i][t-1];
             else
                  if(t<=i*(i-1)/2)
                           nr[i][t]=nr[i-1][t]+nr[i][t-1]-nr[i-1][t-i];
                  else
                           nr[i][t]=0;
```

Probleme de numărare

- **Numărul permutări ale mulţimii {1,2,..., n} care au exact k inversiuni (n, k date)**
  - Mahonian numbers

Nr[n][k]=T(n,k) = coeficientul lui x^k din produsul

$$\prod_{i=0}^{n-1}(1+x+\ldots+x^i)$$

Nr[n][k]=Nr[n][$\binom{n}{2}-k$]

1										
1	1									
1	2	2	1							
1	3	5	6	5	3	1				
1	4	9	15	20	22	20	15	9	4	1

Alte tipuri de probleme

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Fie $G = (N, T, S, P)$ o gramatică independentă de context şi $w \in T^*$.

Se cere să se determine dacă $w \in L(G)$.

- Gramatica este în forma normală a lui Chomsky: producţiile au numai formele

  $A \to BC$ şi $A \to a$,

  cu $A, B \in N$ şi $a \in T^*$.

- Algoritmul CYK (Cocke-Younger-Kasami)

Horia Georgescu. Tehnici de programare. Editura Universităţii din Bucureşti 2005

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu

$P_1: \; S \rightarrow AB$

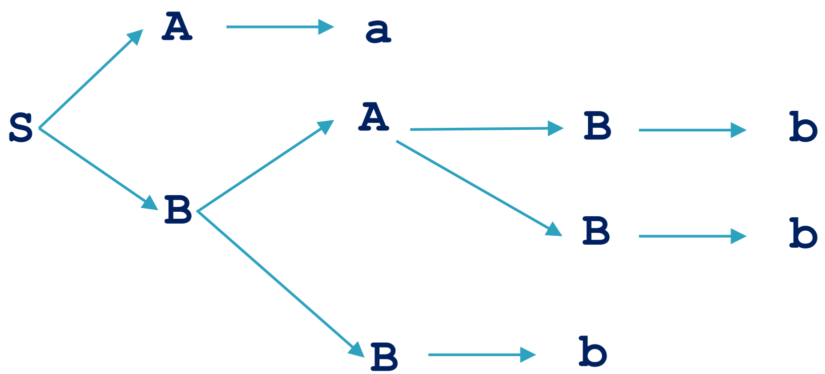
$P_2: \; A \rightarrow BB$

$P_3: \; A \rightarrow a$

$P_4: \; B \rightarrow AB$

$P_5: \; B \rightarrow b$

$w = abbb$



Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

$w = a_1 a_2 \ldots a_n$

$$\begin{array}{cc} (a_1 a_2 \ldots a_k) & (a_{k+1} \ldots a_n) \\ \Uparrow & \Uparrow \\ B & C \end{array}$$

$w \in L(G) \Leftrightarrow$ există $S \rightarrow BC$ în P astfel încât

$B \Rightarrow a_1 a_2 \ldots a_k$ (derivare în oricâţi paşi)

$C \Rightarrow a_{k+1} \ldots a_n$

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

$w=a_1a_2...a_n$

Subproblemă: $M(i,j) = \{ A \in N \mid A \Rightarrow a_i...a_j \}$, unde $\Rightarrow$ semnifică derivare în oricâţi paşi

Soluţie:

Ştim:

Recurenţe:

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

$w=a_1a_2...a_n$

Subproblemă: $M(i,j) = \{ A \in N \mid A \Rightarrow a_i...a_j \}$, unde $\Rightarrow$ semnifică derivare în oricâţi paşi

Soluţie: $w \in L(G) \Leftrightarrow S \in M(1,n)$

Ştim: $M(i,i) = \{ A \in N \mid A \to a_i \in P \}$

Recurenţe:

$$M(i,j) = \{ A \in N \mid \exists\ k \in i..j-1, \text{ astfel încât } \exists B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$$

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Recurenţe:

$M(i,j) = \{ A \in N \mid \exists\ k \in i..j-1$, astfel încât

$\exists B \in M(i,k)$ şi $C \in M(k+1,j)$ cu $A \to BC \in P \}$

Calcul M(i,j):

```
M(i,j) ← ∅
for k=i,j-1
    for A→BC ∈ P cu B∈M(i,k)și C∈M(k+1,j)
        M(i,j) ←  M(i,j) ∪ {A}
```

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

P_1: $S \to AB$

P_2: $A \to BB$

P_3: $A \to a$

P_4: $B \to AB$

P_5: $B \to b$

$w = abbb$

	a 1	*b* 2	*b* 3	*b* 4
1	A P_3 k=1			
2		B P_5 k=2		
3			B P_5 k=3	
4				B P_5 k=4

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

P_1: S → AB

P_2: A → BB

P_3: A → a

P_4: B → AB

P_5: B → b

w = abbb

	a 1	*b* 2	*b* 3	*b* 4
1	A P_3 k=1			
2		B P_5 k=2		
3			B P_5 k=3	
4				B P_5 k=4

M(1,2): k=1 M(1,1)={A}
M(2,2)={B}
Căutăm producții ? → AB

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k)$ şi $C \in M(k+1,j)$ cu $A \rightarrow BC \in P \}$

P_1: $S \rightarrow AB$

P_2: $A \rightarrow BB$

P_3: $A \rightarrow a$

P_4: $B \rightarrow AB$

P_5: $B \rightarrow b$

w = abbb

	a 1	*b* 2	*b* 3	*b* 4
1	A P_3 k=1	S P_1 k=1 B P_4 k=1		
2		B P_5 k=2		
3			B P_5 k=3	
4				B P_5 k=4

M(1,2): k=1 M(1,1)={A}
M(2,2)={B}
Căutăm producţii $Q \rightarrow AB$
$\Rightarrow P_1, P_4 \Rightarrow Q \in \{S, B\}$

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\ k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

P_1: S → AB

P_2: A → BB

P_3: A → a

P_4: B → AB

P_5: B → b

w = abbb

	a 1	b 2	b 3	b 4
1	A P_3 k=1	S P_1 k=1 B P_4 k=1		
2		B P_5 k=2	A P_2 k=2	
3			B P_5 k=3	
4				B P_5 k=4

M(2,3): k=2 M(2,2)={B}

M(3,3)={B}

Productie de forma Q → BB

⇒ P_2 ⇒ N ∈ {A}

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

P_1: S → AB

P_2: A → BB

P_3: A → a

P_4: B → AB

P_5: B → b

w = abbb

	a 1	*b* 2	*b* 3	*b* 4
1	A P_3 k=1	S P_1 k=1; B P_4 k=1		
2		B P_5 k=2	A P_2 k=2	
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(3,4): k=3 M(3,3)={B}

M(4,4)={B}

Productie de forma Q → BB

$\Rightarrow P_2 \Rightarrow Q \in \{A\}$

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

P_1: $S \to AB$

P_2: $A \to BB$

P_3: $A \to a$

P_4: $B \to AB$

P_5: $B \to b$

$w = abbb$

	a 1	*b* 2	*b* 3	*b* 4
1	A P_3 k=1	S P_1 k=1; B P_4 k=1	A P_2 k=2	
2		B P_5 k=2	A P_2 k=2	
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,3): k=1: M(1,1)={A}
M(2,3)={A}
Productie $Q \to AA$
$\Rightarrow$ Nu exista

k=2: M(1,2)={S,B}
M(3,3)={B}
Productie $Q \to SB$
$\Rightarrow P_2 \Rightarrow$ Nu exista
Productie $Q \to BB$
$\Rightarrow P_2 \Rightarrow Q \in \{A\}$

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

P_1: S → AB

P_2: A → BB

P_3: A → a

P_4: B → AB

P_5: B → b

w = abbb

	a 1	b 2	b 3	b 4
1	A P_3 k=1	S P_1 k=1 B P_4 k=1	A P_2 k=2	
2		B P_5 k=2	A P_2 k=2	S P_1 k=3 B P_4 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(2,4): k=2: M(2,2)={S,B}
M(3,4)={A}
Productie N → SA
⇒ Nu exista
Productie N → BA
⇒ Nu exista

k=3: M(2,3)={A}
M(4,4)={B}
Productie N → AB
⇒ P_1, P_4 ⇒ N ∈ {S, B}

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

P_1: $S \to AB$

P_2: $A \to BB$

P_3: $A \to a$

P_4: $B \to AB$

P_5: $B \to b$

$w = abbb$

	1 (*a*)	2 (*b*)	3 (*b*)	4 (*b*)
1	A P_3 k=1	S B P_1 P_4 k=1 k=1	A P_2 k=2	?
2		B P_5 k=2	A P_2 k=2	S B P_1 P_4 k=3 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,4): ?

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

P_1: $S \to AB$

P_2: $A \to BB$

P_3: $A \to a$

P_4: $B \to AB$

P_5: $B \to b$

w = abbb

	a 1	b 2	b 3	b 4
1	A P_3 k=1	S B P_1 P_4 k=1 k=1	A P_2 k=2	S B ? P_1 P_4 k=1 k=1
2		B P_5 k=2	A P_2 k=2	S B P_1 P_4 k=3 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,4):

k=1: M(1,1)={A}

M(2,4)={S, B}

Productie $Q \to SA$ $\Rightarrow$ Nu exista

Productie $Q \to AB$ $\Rightarrow$ P_1, P_4

$\Rightarrow Q \in \{S, B\}$

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

$P_1: S \to AB$

$P_2: A \to BB$

$P_3: A \to a$

$P_4: B \to AB$

$P_5: B \to b$

$w = abbb$

	1 (*a*)	2 (*b*)	3 (*b*)	4 (*b*)
1	A P_3 k=1	S P_1 k=1; B P_4 k=1	A P_2 k=2	S P_1 k=1; B P_4 k=1; ?
2		B P_5 k=2	A P_2 k=2	S P_1 k=3; B P_4 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,4):

k=2: M(1,2)={S, B}
M(3,4)={A}
Productie N → SA ⇒ Nu exista
Productie N → BA ⇒ Nu exista

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \to BC \in P \}$

P_1: S → AB

P_2: A → BB

P_3: A → a

P_4: B → AB

P_5: B → b

w = abbb

	1 (*a*)	2 (*b*)	3 (*b*)	4 (*b*)
1	A P_3 k=1	S P_1 k=1; B P_4 k=1	A P_2 k=2	S P_1 k=1; B P_4 k=1; ?
2		B P_5 k=2	A P_2 k=2	S P_1 k=3; B P_4 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,4):

k=3: M(1,3)={A}

M(4,4)={B}

Productie Q → AB ⇒ P_1, P_4

⇒ Q ∈ {S, B} sunt deja in M(1,4)

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

Exemplu: $M(i,j) = \{ A \in N \mid \exists\, k \in i..j-1,\ B \in M(i,k) \text{ şi } C \in M(k+1,j) \text{ cu } A \rightarrow BC \in P \}$

P_1: $S \rightarrow AB$

P_2: $A \rightarrow BB$

P_3: $A \rightarrow a$

P_4: $B \rightarrow AB$

P_5: $B \rightarrow b$

$w = abbb$

	1 (*a*)	2 (*b*)	3 (*b*)	4 (*b*)
1	A P_3 k=1	S P_1 k=1; B P_4 k=1	A P_2 k=2	S P_1 k=1; B P_4 k=1
2		B P_5 k=2	A P_2 k=2	S P_1 k=3; B P_4 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

$S \in M(1,4) \Rightarrow w \in L(G)$

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

O derivare:

P_1: $S \rightarrow AB$

P_2: $A \rightarrow BB$

P_3: $A \rightarrow a$

P_4: $B \rightarrow AB$

P_5: $B \rightarrow b$

w = abbb

	a 1	*b* 2	*b* 3	*b* 4
1	A P_3 k=1	S P_1 k=1; B P_4 k=1	A P_2 k=2	S P_1 k=1; B P_4 k=1
2		B P_5 k=2	A P_2 k=2	S P_1 k=3; B P_4 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,4, S): $S \rightarrow AB$ (P_1, pentru k=1)
M(1, 1, A)
M(2, 4, B)

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

O derivare:

P_1: S → AB

P_2: A → BB

P_3: A → a

P_4: B → AB

P_5: B → b

w = abbb

	1 (*a*)	2 (*b*)	3 (*b*)	4 (*b*)
1	A P_3 k=1	S P_1 k=1; B P_4 k=1	A P_2 k=2	S P_1 k=1; B P_4 k=1
2		B P_5 k=2	A P_2 k=2	S P_1 k=3; B P_4 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,4, S): S → AB (P_1, pentru k=1)
M(1, 1, A): A → a
M(2, 4, B)

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

O derivare:

P_1: $S \to AB$

P_2: $A \to BB$

P_3: $A \to a$

P_4: $B \to AB$

P_5: $B \to b$

w = abbb

	a 1	b 2	b 3	b 4
1	A P_3 k=1	S B P_1 P_4 k=1 k=1	A P_2 k=2	S B P_1 P_4 k=1 k=1
2		B P_5 k=2	A P_2 k=2	S B P_1 P_4 k=3 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,4, S): $S \to AB$ (P_1, pentru k=1)

M(1, 1, A): $A \to a$

M(2, 4, B): $B \to AB$ (P_4, pentru k=3)

M(2, 3, A):

M(4, 4, B):

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

O derivare:

P_1: $S \to AB$

P_2: $A \to BB$

P_3: $A \to a$

P_4: $B \to AB$

P_5: $B \to b$

$w = abbb$

	a 1	*b* 2	*b* 3	*b* 4
1	A P_3 k=1	S B P_1 P_4 k=1 k=1	A P_2 k=2	S B P_1 P_4 k=1 k=1
2		B P_5 k=2	A P_2 k=2	S B P_1 P_4 k=3 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,4, S): $S \to AB$ (P_1, pentru k=1)

M(1, 1, A): $A \to a$

M(2, 4, B): $B \to AB$ (P_4, pentru k=3)

M(2, 3, A):

M(4, 4, B):

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

O derivare:

$P_1: S \rightarrow AB$

$P_2: A \rightarrow BB$

$P_3: A \rightarrow a$

$P_4: B \rightarrow AB$

$P_5: B \rightarrow b$

$w = abbb$

	a 1	*b* 2	*b* 3	*b* 4
1	A P_3 k=1	S B P_1 P_4 k=1 k=1	A P_2 k=2	S B P_1 P_4 k=1 k=1
2		B P_5 k=2	A P_2 k=2	S B P_1 P_4 k=3 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

$M(1,4, S)$: $S \rightarrow AB$ (P_1, pentru k=1)

    $M(1, 1, A)$: $A \rightarrow a$

    $M(2, 4, B)$: $B \rightarrow AB$ (P_4, pentru k=3)

        $M(2, 3, A)$: $A \rightarrow BB$ (P_2, pentru k=2)

            $M(2, 2, B)$:

            $M(3, 3, B)$:

        $M(4, 4, B)$:

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

O derivare:

P_1: S → AB

P_2: A → BB

P_3: A → a

P_4: B → AB

P_5: B → b

w = abbb

	a 1	b 2	b 3	b 4
1	A P_3 k=1	S B P_1 P_4 k=1 k=1	A P_2 k=2	S B P_1 P_4 k=1 k=1
2		B P_5 k=2	A P_2 k=2	S B P_1 P_4 k=3 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

M(1,4, S): S → AB (P_1, pentru k=1)
    M(1, 1, A): A → a
    M(2, 4, B): B → AB (P_4, pentru k=3)
        M(2, 3, A): A → BB (P_2, pentru k=2)
            M(2, 2, B): B → b
            M(3, 3, B): B → b
        M(4, 4, B): B → b

Verificarea apartenenţei unui cuvânt la limbajul generat de o gramatică independentă de context

O derivare:

$P_1: S \to AB$

$P_2: A \to BB$

$P_3: A \to a$

$P_4: B \to AB$

$P_5: B \to b$

$w = abbb$

	a 1	*b* 2	*b* 3	*b* 4
1	A P_3 k=1	S B P_1 P_4 k=1 k=1	A P_2 k=2	S B P_1 P_4 k=1 k=1
2		B P_5 k=2	A P_2 k=2	S B P_1 P_4 k=3 k=3
3			B P_5 k=3	A P_2 k=2
4				B P_5 k=4

$S \to AB$

$A \to a$

$B \to AB$

$A \to BB$

$B \to b$

$B \to b$

$B \to b$