1.2 Tehnici de generare şi enumerare

În această secţiune sunt prezentate metode de enumerare a aranjamentelor combinatoriale introduse în secţiunea precedentă. Fiecare metodă de enumerare corespunde unei ordini liniare în raport cu care putem defini

- rangul unui aranjament,
- aranjamentul corespunzător unui rang dat, şi
- aranjamentul care urmează după aranjament dat.

De exemplu, permutările mulţimii ordonate $\{a, b, c\}$ cu $a < b < c$ pot fi enumerate în ordine lexicografică:

permutare	rang	permutare precedentă	permutare următoare
$\langle a, b, c\rangle$	0	–	$\langle a, c, b\rangle$
$\langle a, c, b\rangle$	1	$\langle a, b, c\rangle$	$\langle b, a, c\rangle$
$\langle b, a, c\rangle$	2	$\langle a, c, b\rangle$	$\langle b, c, a\rangle$
$\langle b, c, a\rangle$	3	$\langle b, a, c\rangle$	$\langle c, a, b\rangle$
$\langle c, a, b\rangle$	4	$\langle b, c, a\rangle$	$\langle c, b, a\rangle$
$\langle c, b, a\rangle$	5	$\langle c, a, b\rangle$	–

Vom presupune implicit că $S = \{a_1, a_2, \ldots, a_n\}$ este o mulţime total ordonată cu $a_1 < a_2 < \ldots < a_n$. **Rangul** unui element $x \in S$ este $rk(x, S) = i-1$ dacă $x = a_i$. De exemplu, dacă $S = \{1, 2, 3, 4\}$ atunci $rk(1, S) = 0$, $rk(2, S) = 1$, $rk(3, S) = 2$, $rk(4, S) = 3$.

Aranjamentele neordonate de r elemente din S au o reprezentare unică de forma $\{s_1, s_2, \ldots, s_r\}$ cu $s_1 \leq s_2 \leq \ldots \leq s_r$, numită **reprezentarea lexicografică** a aranjamentului respectiv.

În această secţiune vom presupune implicit că toate aranjamentele neordonate sunt reprezentate lexicografic.

1.2.1 Ordonarea lexicografică

Ordonarea elementelor lui S induce o ordine totală pe mulţimea şirurilor de elemente din S: $s_1 s_2 \ldots s_r <_{lex} s'_1 s'_2 \ldots s'_t$ dacă există $1 \leq i \leq \min(r, t)$ cu $s_1 s_2 \ldots s_{i-1} = s'_1 s'_s \ldots s'_{i-1}$ şi $s_i < s'_i$, sau $r < t$ şi $s_1 s_2 \ldots s_r = s'_1 s'_2 \ldots s'_r$.

Toate aranjamentele combinatoriale se pot ordona lexicografic:

- Dacă $\pi_1 = \langle s_1, s_2, \ldots, s_r\rangle$, $\pi_2 = \langle s'_1, s'_2, \ldots, s'_t\rangle$ sunt două aranjamente ordonate, atunci π_1 este lexicografic mai mic sau egal ca π_2 dacă $s_1 s_2 \ldots s_r \leq_{lex} s'_1 s'_2 \ldots s'_t$.

- Dacă $C_1 = \{s_1, s_2, \ldots, s_r\}$, $C_2 = \{s'_1, s'_2, \ldots, s'_t\}$ sunt reprezentările lexicografice a două aranjamente neordonate, atunci C_1 este lexicografic mai mic sau egal ca C_2 dacă $s_1 s_2 \ldots s_r \leq_{lex} s'_1 s'_2 \ldots s'_t$.

Exemplul 1. Dacă $S = \{a, b, c\}$, avem următoarele enumerări de aranjamente combinatoriale în ordine lexicografică:

- permutări: $\langle a, b, c\rangle, \langle a, c, b\rangle, \langle b, a, c\rangle, \langle b, c, a\rangle, \langle c, a, b\rangle, \langle c, b, a\rangle$,
  2-permutări: $\langle a, b\rangle, \langle a, c\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, a\rangle, \langle c, b\rangle$
- 2-permutări cu repetiţie: $\langle a, a\rangle$, $\langle a, b\rangle$, $\langle a, c\rangle$, $\langle b, a\rangle$, $\langle b, b\rangle$, $\langle b, c\rangle$, $\langle c, a\rangle$, $\langle c, b\rangle$, $\langle c, c\rangle$
- combinări: $\emptyset, \{a\}, \{a, b\}, \{a, b, c\}, \{a, c\}, \{b\}, \{b, c\}, \{c\}$
- 2-combinări: $\{a, b\}, \{a, c\}, \{b, c\}$
- 2-combinări cu repetiţie: $\{a, a\}$, $\{a, b\}$, $\{a, c\}$, $\{b, b\}$, $\{b, c\}$, $\{c, c\}$. □

1.2.2 Alte ordonări

Ordonarea binară a submulţimilor lui S este determinată de *reprezentarea binară* a submulţimilor:

> Dacă A este o submulţime a lui S, **reprezentarea binară** a lui A este şirul de n biţi $b_1 b_2 \ldots b_n$ cu $b_i = \begin{cases} 1 & \text{dacă } a_i \in A, \\ 0 & \text{dacă } a_i \notin A. \end{cases}$
>
> **Rangul binar** al lui A este numărul a cărui reprezentare în baza 2 este şirul binar $b_1 b_2 \ldots b_n$, iar enumerarea în ordine binară a submulţimilor lui S este cea în ordine crescătoare a rangului binar.

Exemplul 2. Enumerarea submulţimilor lui $\{a, b, c\}$ în ordine crescătoare a rangului binar, este

Rang binar	0	1	2	3	4	5	6	7
Repezentare binară	000	001	010	011	100	101	110	111
Submulţime	$\emptyset$,	$\{c\}$,	$\{b\}$,	$\{b, c\}$,	$\{a\}$,	$\{a, c\}$,	$\{a, b\}$,	$\{a, b, c\}$

□

O ordonare utilă a permutărilor lui S fost descoperită de Heap în 1963 [8]. În această ordonare, pe care o denumim **ordonare Heap**, orice permutare se obţine din cea precedentă interschimbând valorile a două elemente. Datorită costului foarte mic de generare a permutării următoare, rezultă unul din cei mai rapizi algoritmi de enumerare a tuturor permutărilor unei mulţimi.

Exemplul 3. Ordonarea Heap a permutărilor lui $\{a, b, c\}$ este

Rang Heap	0	1	2	3	4	5
Permutare	$\langle a, b, c\rangle$,	$\langle b, a, c\rangle$,	$\langle c, a, b\rangle$,	$\langle a, c, b\rangle$,	$\langle b, c, a\rangle$,	$\langle c, b, a\rangle$

1.2.3 Algoritmi de ranking, unranking şi enumerare

Pentru fiecare ordonare a unui tip de aranjament combinatorial, se pot defini următoarele operaţii elementare:

1. de ranking, care calculează rangul unui aranjament.
2. de unranking, care calculează aranjamentul care are un anumit rang. Aceasta este inversa operaţiei de ranking.
3. de calcul al aranjamentului care urmează după un aranjament.

În cele ce urmează vom scrie $\text{rkLex}_S(\pi)$ pentru rangul unui aranjament π în ordine lexicografică, şi $\text{nextLex}_S(\pi)$ pentru aranjamentul după π în ordine lexicografică.

Aranjamentele combinatoriale de un anumit tip se pot enumera lexicografic în două feluri:

1. Dacă N este numărul de aranjamente de tipul respectiv, atunci

   for $k = 0$ to $N - 1$ **do**
   calculează şi afişează aranjamentul π cu $\text{rkLex}_S(\pi) = k$

2. Dacă se cunoaşte primul aranjament π_{start} şi ultimul aranjament π_{end} de tipul respectiv în ordine lexicografică, atunci

   $\pi := \pi_{\text{start}}$
   afişează π
   while $\pi \neq \pi_{\text{end}}$ **do**
   $\pi := \text{nextLex}_S(\pi)$
   afişează π

De exemplu, algoritmii de enumerare a tuturor 3-permutărilor lui $\{1, 2, 3, 4\}$ se poate face cu algoritmul

for $k = 0$ to $P(4,3) - 1$ **do**
  calculează şi afişează 3-permutarea π cu $\text{rkLex}_S(\pi) = k$

sau cu algoritmul

$\pi := \langle 1, 2, 3\rangle$
afişează π
while $\pi \neq \langle 4, 3, 2\rangle$ **do**
  $\pi := \text{nextLex}_S(\pi)$
  afişează π

Permutări şi r-permutări ordonate lexicografic

Dacă $r = 0$, singura r-permutare a unei mulţimi S este $\langle\rangle$ cu $\text{rkLex}_S(\langle\rangle) = 0$. În continuare, vom presupune că $r > 0$. Enumerarea lexicografică a r-permutărilor lui S este o secvenţă de n blocuri $B_1, B_2, \ldots, B_n$ unde fiecare bloc B_i conţine toate $(r-1)$-permutările $\langle a_i, p_2, \ldots, p_r\rangle$ enumerate în ordine lexicografică. Acest lucru înseamnă că B_i se obţine adăugând a_i ca prim element la enumerarea lexicografică a $(r-1)$-permutărilor lui $S - \{a_i\}$. De exemplu, enumerarea lexicografică a 3-permutărilor lui $\{a, b, c, d\}$ este secvenţa de blocuri

$$\underbrace{\langle a,b,c\rangle, \langle a,b,d\rangle, \langle a,c,b\rangle, \langle a,c,d\rangle, \langle a,d,b\rangle, \langle a,d,c\rangle}_{B_1},$$
$$\underbrace{\langle b,a,c\rangle, \langle b,a,d\rangle, \langle b,c,a\rangle, \langle b,c,d\rangle, \langle b,d,a\rangle, \langle b,d,c\rangle}_{B_2},$$
$$\underbrace{\langle c,a,b\rangle, \langle c,a,d\rangle, \langle c,b,a\rangle, \langle c,b,d\rangle, \langle c,d,a\rangle, \langle c,d,b\rangle}_{B_3},$$
$$\underbrace{\langle d,a,b\rangle, \langle d,a,c\rangle, \langle d,b,a\rangle, \langle d,b,c\rangle, \langle d,c,a\rangle, \langle d,c,b\rangle}_{B_4}.$$

iar B_3 se obţine adăugând $a_3 = c$ ca prim element la enumerarea lexicografică a 2-permutărilor lui $\{a, b, d\}$: $\langle a,b\rangle, \langle a,d\rangle, \langle b,a\rangle, \langle b,d\rangle, \langle d,a\rangle, \langle d,b\rangle$.

Aşadar fiecare bloc B_i are $P(n-1, r-1)$ elemente şi

- r-permutarea $\pi = \langle p_1, p_2, \ldots, p_r\rangle$ cu $rk(p_1, S) = i$ are
$$\text{rkLex}_S(\pi) = i \cdot P(n-1, r-1) + \text{rkLex}_{S-\{p_1\}}(\langle p_2, \ldots, p_n\rangle) \quad (1.11)$$
- Dacă π este o r-permutare cu $\text{rkLex}_S(\pi) = k$ şi $i = \lfloor k/P(n-1, r-1)\rfloor$ atunci $\pi = \langle p_1, p_2, \ldots, p_r\rangle$ cu $rk(p_1, S) = i$ şi
$$\text{rkLex}_{S-\{p_1\}}(\langle p_2, \ldots, p_n\rangle) = k - i \cdot P(n-1, r-1) \quad (1.12)$$

Formula (1.11) ne dă un *algoritm recursiv de ranking* a permutărilor ordonate lexicografic, iar formula (1.12) ne dă un *algoritm recursiv de unranking* a permutărilor ordonate lexicografic. Cazul de bază pentru aceşti algoritmi recursivi este dat de formula

$$\text{rkLex}_S(\pi) = 0 \text{ dacă şi numai dacă } \pi = \langle a_1, \ldots, a_r \rangle. \tag{1.13}$$

Exemplul 4. Fie $S = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$. Ce rang are permutarea $\langle \mathtt{c}, \mathtt{b}, \mathtt{a}, \mathtt{d} \rangle$ în ordine lexicografică?

RĂSPUNS:

$$\begin{aligned}
\text{rkLex}_S(\langle \mathtt{c}, \mathtt{b}, \mathtt{a}, \mathtt{d} \rangle) &= rk(c, S) \cdot P(3,3) + \text{rkLex}_{\{\mathtt{a},\mathtt{b},\mathtt{d}\}}(\langle \mathtt{b}, \mathtt{a}, \mathtt{d} \rangle) \\
&= 2 \cdot 6 + rk(\mathtt{b}, \{\mathtt{a}, \mathtt{b}, \mathtt{d}\}) \cdot P(2,2) + \text{rkLex}_{\{\mathtt{a},\mathtt{d}\}}(\langle \mathtt{a}, \mathtt{d} \rangle) \\
&= 2 \cdot 6 + 1 \cdot 2 + 0 = 14.
\end{aligned}$$

□

Exemplul 5. Fie $S = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$. Care 3-permutare are rangul 20 în ordine lexicografică?

RĂSPUNS: Fie $\langle p_1, p_2, p_3 \rangle$ permutarea căutată. Ştim că

- $rk(p_1, S) = \lfloor 20/P(3,2) \rfloor = 3$, deci $p_1 = \mathtt{d}$ şi $\text{rkLex}_{\{\mathtt{a},\mathtt{b},\mathtt{c}\}}(\langle p_2, p_3 \rangle) = 20 - 3 \cdot 3! = 2$.
- $rk(p_2, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}) = \lfloor 2/P(2,1) \rfloor = 1$, deci $p_2 = \mathtt{b}$ şi $\text{rkLex}_{\{\mathtt{a},\mathtt{c}\}}(\langle p_3 \rangle) = 2 - 1 \cdot 2! = 0$. Însă $\text{rkLex}_{\{\mathtt{a},\mathtt{c}\}}(\langle p_3 \rangle) = 0$ implică $\langle p_3 \rangle = \langle \mathtt{a} \rangle$, deci $p_3 = \mathtt{a}$.

Rezultă că $\langle p_1, p_2, p_3 \rangle = \langle \mathtt{d}, \mathtt{b}, \mathtt{a} \rangle$. □

Deoarece permutările sunt n-permutări, aceşti algoritmi de ranking şi unranking funcţionează şi pentru permutări.

Metoda de generare a r-permutării care *urmează* după $\langle p_1, p_2, \ldots, p_r \rangle$ în ordine lexicografică este:

> Dacă $\langle p_1, p_2, \ldots, p_r \rangle = \langle a_n, a_{n-1}, \ldots, a_{n-r+1} \rangle$, nu există r-permutare următoare. Altfel, fie i cel mai mare indice din $\{1, 2, \ldots, r\}$ astfel încât $p_i < \max(S - \{p_1, \ldots, p_{i-1}\})$.
>
> Atunci $\text{nextLex}_S(\langle p_1, \ldots, p_r \rangle) = \langle p_1, \ldots, p_{i-1}, p'_i, p'_{i+1}, \ldots, p'_r \rangle$ unde
>
> - p'_i este elementul cu rang minim din $S - \{p_1, \ldots, p_{i-1}\}$ care satisface condiţia $p'_i > p_i$, şi
> - $p'_{i+1}, \ldots, p'_r$ sunt primele $r - i$ elemente ale lui $S - \{p_1, \ldots, p_{i-1}, p'_i\}$ enumerate în ordine crescătoare a rangului.

Exemplul 6. Ce 4-permutare a lui $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ urmează lexicografic după$\langle \mathtt{f}, \mathtt{c}, \mathtt{e}, \mathtt{d}\rangle$?

RĂSPUNS: Fie $S = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ şi $\langle p_1, p_2, p_3, p_4\rangle = \langle \mathtt{f}, \mathtt{c}, \mathtt{e}, \mathtt{d}\rangle$. Cel mai mare i pentru care $p_i < \max(S - \{p_1, \ldots, p_{i-1}\})$ este $i = 2$. Rezultă că $\text{nextLex}_S(\langle \mathtt{f}, \mathtt{c}, \mathtt{e}, \mathtt{d}\rangle) = \langle \mathtt{f}, p'_2, p'_3, p'_4\rangle$ unde

- p'_2 este elementul cu rang minim din $S - \{\mathtt{f}\}$ care satisface condiţia $p'_2 > \mathtt{c}$, adică $p'_2 = \mathtt{d}$, şi
- p'_3, p'_4 sunt primele 2 elemente ale lui $S - \{\mathtt{f}, \mathtt{d}\} = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{e}\}$ în ordine crescătoare a rangului, adică $p'_3 = \mathtt{a}$, $p'_4 = \mathtt{b}$.

Deci $\text{nextLex}_S(\langle \mathtt{f}, \mathtt{c}, \mathtt{e}, \mathtt{d}\rangle) = \langle \mathtt{f}, \mathtt{d}, \mathtt{a}, \mathtt{b}\rangle$. □

Pentru $r = n$, această metodă ne dă permutarea ce urmează după o permutare în ordine lexicografică. Metoda următoare de calcul al lui $\text{nextLex}_S(\pi)$ este optimizată pentru cazul în care $\pi = \langle p_1, p_2, \ldots, p_n\rangle$ este o permutare:

> Dacă $\langle p_1, p_2, \ldots, p_n\rangle = \langle a_n, a_{n-1}, \ldots, a_1\rangle$, nu există permutare următoare. Altfel, fie i cel mai mare indice din $\{1, 2, \ldots, n\}$ astfel încât $p_i < p_{i+1}$, şi j cel mai mare indice astfel încât $j > i$ şi $p_i < p_j$. În acest caz, $\text{nextLex}_S(\pi)$ se obţine din π în 2 paşi: mai întâi se interschimbă valorile elementelor p_i şi p_j, apoi se inversează ordinea elementelor $p_{i+1}, \ldots, p_n$.

Exemplul 7. Ce permutare urmează lexicografic după $\pi = \langle 1, 5, 3, 6, 4, 2\rangle$?

RĂSPUNS. În acest caz, $S = \{1, 2, 3, 4, 5, 6\}$, $i = 3$ şi $j = 5$. În primul pas, obţinem permutarea $\pi' = \langle 1, 5, 4, 6, 3, 2\rangle$, iar în al doilea pas obţinem $\text{nextLex}_S(\langle 1, 5, 3, 6, 4, 2\rangle) = \langle 1, 5, 4, 2, 3, 6\rangle$. □

Permutări enumerate în ordinea lui Heap

`[ -- TODO -- ]`

r-permutări cu repetiţie ordonate lexicografic

Fie $\pi = \langle p_1, p_2, \ldots, p_r\rangle$ o r-permutare cu repetiţie a lui S. Se observă că

$$\text{rkLex}_S(\langle p_1, p_2, \ldots, p_r\rangle) = \sum_{k=1}^{r} c_k \cdot n^{r-k} \tag{1.14}$$

unde $c_k = rk(p_k, S)$ pentru $1 \leq k \leq n$. Altfel spus, $c_1 c_2 \cdots c_r$ este reprezentarea rangului lexicografic al lui π în baza n, folosind r cifre.

Din această formulă deducem algoritmi iterativi de ranking, unranking şi enumerare pentru r-permutări cu repetiţie în ordine lexicografică:

1. $\text{rkLex}_S(\langle p_1, p_2, \ldots, p_r\rangle) = \sum_{k=1}^{r} rk_S(p_k, S) \cdot n^{r-k}$.

2. Dacă $\text{rkLex}_S(\langle p_1, p_2, \ldots, p_r\rangle) = k$ şi reprezentarea lui k în baza n este $c_1c_2 \ldots c_r$ atunci $p_i = a_{c_i}$ pentru $1 \leq i \leq r$.

3. $\text{nextLex}_S(\langle p_1, p_2, \ldots, p_r\rangle)$ nu există dacă $p_1 = p_2 = \ldots = p_r = a_n$. Altfel, $\text{nextLex}_S(\langle p_1, p_2, \ldots, p_r\rangle) = \langle p'_1, \ldots, p'_r\rangle$ se calculează astfel:

$$\text{Fie } j = \begin{cases} r & \text{dacă } p_r \neq a_n, \\ \min\{i \mid p_i = p_{i+1} = \ldots = p_r = a_n\} - 1 & \text{în caz contrar.} \end{cases}$$

$$\text{Atunci } p'_i = \begin{cases} p_i & \text{dacă } j < i, \\ a_\ell & \text{dacă } j = i \text{ şi } \ell = rk(p_i, S) + 2, \\ a_1 & \text{dacă } i < j \leq r. \end{cases}$$

Exemplul 8. Ce rang are 5-permutarea cu repetiţie $\langle 2, 1, 2, 2, 1\rangle$ a mulţimii $\{1, 2, 3\}$ în ordine lexicografică?

RĂSPUNS: $\text{rkLex}_{\{1,2,3\}}(\langle 2, 1, 2, 2, 1\rangle) = 1 \cdot 3^4 + 0 \cdot 3^3 + 1 \cdot 3^2 + 1 \cdot 3^1 + 0 \cdot 3^2 = 81 + 9 + 3 = 93$. □

Exemplul 9. Ce 4-permutare cu repetiţie a mulţimii $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ are rangul lexicografic 1111?

RĂSPUNS: $1111 = \mathtt{5} \cdot \mathtt{6}^\mathtt{3} + \mathtt{0} \cdot \mathtt{6}^\mathtt{2} + \mathtt{5} \cdot \mathtt{6} + \mathtt{1}$, deci reprezentarea lui 1111 în baza 6 este $\mathtt{5051}_\mathtt{6}$. Rezultă că 4-permutarea cu repetiţie cu acest rang este $\langle \mathtt{f}, \mathtt{a}, \mathtt{f}, \mathtt{b}\rangle$. □

Exemplul 10. Ce 3-permutare cu repetiţie a lui $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ urmează lexicografic după $\langle \mathtt{b}, \mathtt{a}, \mathtt{d}\rangle$?

RĂSPUNS: $\langle \mathtt{b}, \mathtt{c}, \mathtt{a}\rangle$. □

Submulţimi ordonate binar

De obicei, submulţimile lui S sunt enumerate în ordinea crescătoare a rangului binar. Formula de calcul a rangului binar al unei submulţimi A a lui S a fost prezentată în secţiunea 1.2.2. Reţinem că:

1. Rangul binar al lui $A \in 2^S$ este numărul cu reprezentarea binară $b_1b_2 \ldots b_n$ unde $b_i = \begin{cases} 1 & \text{dacă } a_i \in A, \\ 0 & \text{dacă } a_i \notin A \end{cases}$
adică $\sum_{i=1}^{n} b_i \cdot 2^{n-i}$.

2. Submulţimea A a lui S cu rangul binar k se determină astfel:

   - Mai întâi se calculează reprezentarea binară $b_1 b_2 \dots b_n$ a lui k,
   - Apoi se calculează $A = \{a_i \mid 1 \leq i \leq n$ şi $b_i = 1\}$.

Exemplul 11. Ce submulţime a lui $\{\mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}, \mathtt{5}, \mathtt{6}, \mathtt{7}\}$ are rangul binar 107?

RĂSPUNS: $107 = \mathtt{1101011}_2$, deci submulţimea căutată este $\{\mathtt{1}, \mathtt{2}, \mathtt{4}, \mathtt{6}, \mathtt{7}\}$. □

Exemplul 12. Care este rangul binar al submulţimii $\{\mathtt{a}, \mathtt{c}, \mathtt{e}\}$ a mulţimii $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$?

RĂSPUNS: $\mathtt{101010}_2 = 2^5 + 2^3 + 2 = 42$. □

Submulţimi ordonate lexicografic

Fie $\{p_1, \dots, p_u\}$ şi $\{p'_1, \dots, p'_v\}$ reprezentările lexicografice ale unor submulţimi ale lui S. Se observă că $\{p_1, \dots, p_u\}$ este lexicografic înaintea lui $\{p'_1, \dots, p'_v\}$ dacă şi numai dacă există $1 \leq i \leq \min(u, v)$ astfel încât

- ▶ $p_j = p'_j$ pentru toţi $1 \leq j < i$ şi
- ▶ $p_i < p'_i$ sau ($i = u < v$ şi $p_i = p'_i$).

În general, enumerarea lexicografică a submulţimilor lui S este o secvenţă de blocuri $B_0, B_1, B_2, \dots, B_n$ unde $B_0 = \emptyset$ şi fiecare alt bloc B_i este de forma

$$\{a_i\} \cup X_1, \dots, \{a_i\} \cup X_{n_i}$$

cu $n_i = 2^{n-i}$ şi $X_1, \dots, X_{n_i}$ o enumerare lexicografică a submulţimilor lui $\{a_{i+1}, \dots, a_n\}$. De exemplu, enumerarea lexicografică a submulţimulor lui $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ cu $\mathtt{a} < \mathtt{b} < \mathtt{c}$ este $\underbrace{\emptyset}_{B_0}, \underbrace{\{\mathtt{a}\}, \{\mathtt{a}, \mathtt{b}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \{\mathtt{a}, \mathtt{c}\}}_{B_1}, \underbrace{\{\mathtt{b}\}, \{\mathtt{b}, \mathtt{c}\}}_{B_2}, \underbrace{\{\mathtt{c}\}}_{B_3}$.

Deci, dacă $i > 0$ atunci (1) fiecare bloc B_i este precedat de $1 + \sum_{j=1}^{i-1} 2^{n-j} = 1 + 2^n - 2^{n-i+1}$ submulţimi, şi (2) A este în B_i dacă şi numai dacă $\min A = a_i$.

Rezultă că dacă $A \in 2^S$ atunci

$$\mathrm{rkLex}_S(A) = \begin{cases} 0 & \text{dacă } A = \emptyset, \\ 1 + 2^n - 2^{n-i+1} & \text{dacă } \min A = a_i \text{ şi} \\ \quad + \mathrm{rkLex}_{S - \{a_1, \dots, a_i\}}(A') & A' = A - \{a_i\} \end{cases} \tag{1.15}$$

şi

$$\mathrm{nextLex}_S(A) = \begin{cases} \{a_1\} & \text{dacă } A = \emptyset, \\ A \cup \{a_{i+1}\} & \text{dacă } \max A = a_i \neq a_n, \\ (X - \{a_j\}) \cup \{a_{j+1}\} & \text{dacă } A = X \cup \{a_i, \ldots, a_n\} \\ & \quad \text{şi } X = \emptyset \text{ sau} \\ & \qquad \max X = a_j < a_{i-1}, \\ \text{nu există} & \text{dacă } A = \{a_n\}. \end{cases} \tag{1.16}$$

Dacă $\mathrm{rkLex}_S(A) = k$ atunci

$$A = \begin{cases} \emptyset & \text{dacă } k = 0, \\ \{a_i\} \cup A' & \text{dacă } 2^{n-i} \in (\frac{2^n - k}{2}, 2^n - k] \text{ şi} \\ & \ \mathrm{rkLex}_{\{a_{i+1}+1, \ldots, a_n\}}(A') = k - 1 - 2^n + 2^{n-i+1}. \end{cases} \tag{1.17}$$

Acestea sunt formule recursive de calcul pentru algoritmii de ranking, unranking, şi calcul al submulţimii următoare, în ordine lexicografică.

Exemplul 14. Care sunt rangurile lexicografice ale submulţimilor $\{\mathtt{c}\}$ şi $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ ale lui $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$?

RĂSPUNS:

$$\begin{aligned} \mathrm{rkLex}_{\{\mathtt{a},\mathtt{b},\mathtt{c}\}}(\{\mathtt{c}\}) &= 1 + 2^3 - 2^{3-3+1} + \mathrm{rkLex}_{\{\mathtt{b},\mathtt{c}\}}(\emptyset) = 7 + 0 = 7, \\ \mathrm{rkLex}_{\{\mathtt{a},\mathtt{b},\mathtt{c}\}}(\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}) &= 1 + 2^3 - 2^3 + \mathrm{rkLex}_{\{\mathtt{b},\mathtt{c}\}}(\{\mathtt{b}, \mathtt{c}\}) \\ &= 1 + (1 + 2^2 - 2^2 + \mathrm{rkLex}_{\{\mathtt{c}\}}(\{\mathtt{c}\})) \\ &= 2 + (1 + 2^1 - 2^1 + \mathrm{rkLex}_{\emptyset}(\emptyset)) = 3. \end{aligned}$$

□

Exemplul 15. Ce submulţime a lui $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ are rangul lexicografic 12?

RĂSPUNS: Fie A submulţimea căutată. $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ are 4 elemente. Dacă $I = \left(\frac{2^4 - 12}{2}, 2^4 - 12\right] = (2, 4]$ atunci pentru $i = 2$ avem $2^{4-i} = 2^2 = 4 \in I$, deci $A = \{\mathtt{b}\} \cup A'$ cu $\mathrm{rkLex}_{\{\mathtt{c},\mathtt{d}\}}(A') = 12 - 1 - 2^4 + 2^3 = 3$.

$\{\mathtt{c}, \mathtt{d}\}$ are 2 elemente. Dacă $I = \left(\frac{2^2 - 3}{2}, 2^2 - 3\right] = (0.5, 1]$ atunci pentru $i = 2$ avem $2^{2-2} = 1 \in I$, deci $A' = \{\mathtt{d}\} \cup A''$ cu $\mathrm{rkLex}_{\{\mathtt{d}\}}(A'') = 3 - 1 - 2^2 + 2^{2-2+1} = 0$. Rezultă că $A'' = \emptyset$. În concluzie, $A = \{\mathtt{b}, \mathtt{d}\}$. □

Exemplul 16. Ce submulţime a lui $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ urmează după $\{\mathtt{a}, \mathtt{b}, \mathtt{e}, \mathtt{f}\}$ în ordine lexicografică?

RĂSPUNS: $A = \{\mathtt{a}, \mathtt{b}\} \cup \{\mathtt{e}, \mathtt{f}\}$ cu $\max\{\mathtt{a}, \mathtt{b}\} = \mathtt{b} < \mathtt{d}$, deci se aplică cazul 3 al formulei (1.16) şi se obţine $\mathrm{nextLex}_{\{\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d},\mathtt{e},\mathtt{f}\}}(\{\mathtt{a}, \mathtt{b}, \mathtt{e}, \mathtt{f}\}) = \{\mathtt{a}, \mathtt{c}\}$. □

r-combinări ordonate lexicografic

Fie $\{p_1, \dots, p_r\}$ şi $\{p'_1, \dots, p'_r\}$ reprezentările lexicografice a două r-combinări ale lui S. Se observă că $\{p_1, \dots, p_r\}$ este lexicografic înaintea lui $\{p'_1, \dots, p'_r\}$ dacă şi numai dacă există $1 \leq i \leq r$ astfel încât

$$p_j = p'_j \text{ pentru toţi } 1 \leq j < i \text{ şi } p_i < p'_i.$$

În general, enumerarea lexicografică a r-combinărilor lui S este o secvenţă de blocuri $B_1, B_2, \dots, B_{n-r+1}$ unde fiecare bloc B_i este de forma

$$\{a_i\} \cup X_1, \dots, \{a_i\} \cup X_{n_i}$$

cu $n_i = \binom{n-i}{r-1}$ şi $X_1, \dots, X_{n_i}$ enumerarea lexicografică a $(r-1)$-combinărilor lui $\{a_{i+1}, \dots, a_n\}$. De exemplu, enumerarea lexicografică a 3-combinărilor lui $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}\}$ este

$$\underbrace{\{\mathtt{a},\mathtt{b},\mathtt{c}\}, \{\mathtt{a},\mathtt{b},\mathtt{d}\}, \{\mathtt{a},\mathtt{b},\mathtt{e}\}, \{\mathtt{a},\mathtt{c},\mathtt{d}\}, \{\mathtt{a},\mathtt{c},\mathtt{e}\}, \{\mathtt{a},\mathtt{d},\mathtt{e}\}}_{B_1},$$
$$\underbrace{\{\mathtt{b},\mathtt{c},\mathtt{d}\}, \{\mathtt{b},\mathtt{c},\mathtt{e}\}, \{\mathtt{b},\mathtt{d},\mathtt{e}\}}_{B_2}, \underbrace{\{\mathtt{c},\mathtt{d},\mathtt{e}\}}_{B_3}$$

Deci (1) fiecare bloc B_i este precedat de $\sum_{j=1}^{i-1} \binom{n-j}{r-1} = \binom{n}{r} - \binom{n-i+1}{r}$ r-combinări, şi (2) C este în B_i dacă şi numai dacă $\min C = a_i$.

Rezultă că dacă $C = \{p_1, p_2, \dots, p_r\}$ este reprezentarea lexicografică a unei r-combinări a lui S atunci

1. Dacă $r = 1$ atunci $C = \{p_1\}$ şi $\mathrm{rkLex}_S(C) = rk(p_1, S)$ Altfel
$$\mathrm{rkLex}_S(C) = \binom{n}{r} - \binom{n - rk(p_1, S)}{r} + \mathrm{rkLex}_{S'}(\{p_2, \dots, p_r\}).$$
unde $S' = \{s \in S \mid p_1 < s\}$.
2. Dacă $\mathrm{rkLex}_S(C) = 0$ atunci $C = \{a_1, \dots, a_r\}$. Dacă $\mathrm{rkLex}_S(C) = k > 0$ atunci $k < \binom{n}{r}$ şi $C = \{a_i\} \cup C'$ unde
   - i este cel mai mic număr astfel încât
   $$\binom{n-i}{r} < \binom{n}{r} - k.$$
   - C' este $(r-1)$-combinarea lui $S' = \{s \in S \mid s > a_i\}$ cu
   $$\mathrm{rkLex}_{S'}(C') = k - \binom{n}{r} + \binom{n-i+1}{r}.$$

3. Dacă $i = \max\{j \mid p_j \neq a_j\}$ atunci

$$\text{nextLex}_S(C) = \begin{cases} \text{nu există} & \text{dacă } i = 1, \\ \{p_j \mid j < i\} \cup & \text{dacă } p_i = a_k. \\ \ \{a_{k+1}, \ldots, a_{k+r-i-1}\} & \end{cases}$$

Exemplul 17. Ce rang lexicografic are 2-combinarea $\{\texttt{2}, \texttt{5}\}$ a mulţimii $\{\texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}\}$? Ce rang lexicografic are 3-combinarea $\{\texttt{2}, \texttt{3}, \texttt{5}\}$ a mulţimii $\{\texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}\}$?

RĂSPUNS:

$$\begin{aligned} \text{rkLex}_{\{\texttt{1},\texttt{2},\texttt{3},\texttt{4},\texttt{5}\}}(\{\texttt{2}, \texttt{5}\}) &= \binom{5}{2} - \binom{5-1}{2} + \text{rkLex}_{\{\texttt{3},\texttt{4},\texttt{5}\}}(\{\texttt{5}\}) \\ &= 10 - 6 + rk(5, \{3, 4, 5\}) = 10 - 6 + 2 = \mathbf{6}. \\ \text{rkLex}_{\{\texttt{1},\texttt{2},\texttt{3},\texttt{4},\texttt{5}\}}(\{\texttt{2}, \texttt{3}, \texttt{5}\}) &= \binom{5}{3} - \binom{5-1}{3} + \text{rkLex}_{\{\texttt{3},\texttt{4},\texttt{5}\}}(\{\texttt{3}, \texttt{5}\}) \\ &= 10 - 4 + \binom{3}{2} - \binom{3-0}{2} + \text{rkLex}_{\{\texttt{4},\texttt{5}\}}(\{\texttt{5}\}) \\ &= 6 + 1 = \mathbf{7}. \end{aligned}$$

□

Exemplul 18. Care 3-combinare a lui $\{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e}\}$ are rangul lexicografic 7?

RĂSPUNS: Fie C 3-combinarea cu rangul lexicografic 7.

$$\binom{5}{3} - 7 = 10 - 7 = 3 \text{ şi } \binom{5-1}{3} = 3 \geq 3, \ \binom{5-2}{3} = 1 < 3,$$

deci cel mai mic i pentru care $\binom{5-i}{3} < 3$ este $i = 2$. Rezultă că $C = \{\texttt{b}\} \cup C'$ unde C' este 2-combinarea lui $\{\texttt{c}, \texttt{d}, \texttt{e}\}$ cu rangul lexicografic $7 - \binom{5}{3} + \binom{4}{3} = 7 - 10 + 4 = 1$.

$$\binom{3}{2} - 1 = 3 - 1 = 2 \text{ şi } \binom{3-1}{2} = 1 < 2,$$

deci cel mai mic i pentru care $\binom{3-i}{2} < 2$ este $i = 1$. Rezultă că $C' = \{\texttt{c}\} \cup C''$ unde C' este 1-combinarea lui $\{\texttt{d}, \texttt{e}\}$ cu rangul lexicografic $1 - \binom{3}{2} + \binom{3}{2} = 1$, deci $C'' = \{\texttt{e}\}$.

Deci $C = \{\mathtt{b}, \mathtt{c}, \mathtt{e}\}$. □

Exemplul 19. Ce 4-combinare a lui $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ urmează după $\{\mathtt{a}, \mathtt{d}, \mathtt{e}, \mathtt{f}\}$ în ordine lexicografică?

RĂSPUNS: $\{\mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}\}$. □

1.2.4 Concluzii

În această secţiune au fost prezentate metode de ranking, unranking, şi generare a aranjamentului următor pentru următoarele tipuri de aranjamente:

- aranjamente enumerate lexicografic: permutări, r-permutări, r-permutări cu repetiţie, submulţimi, şi r-combinări
- aranjamente enumerate binar: submulţimi
- aranjamente enumerate în ordinea Heap: permutări.

1.2.5 Exerciţii

1. Implementaţi în unul din limbajele Python, C++ sau Java un program care să citească de la consolă următoarele date:
   - reprezentarea lexicografică a unei mulţimi S de elemente,
   - un identificator pentru un anumit tip de aranjament combinatorial: `P` pentru permutare; `RP` pentru r-permutare; `RR` pentru r-permutare cu repetiţie; sau `S` pentru submulţime; sau `RC` pentru r-combinare cu repetiţie,
   - valoarea lui r pentru tipurile de aranjament `RP`, `RR`, `RC`,
   - un număr $k \in \mathbb{N}$

   şi returnează aranjamentul cu rangul lexicografic k.
2. Implementaţi în unul din limbajele Python, C++ sau Java un program care să citească de la consolă următoarele date:
   - reprezentarea lexicografică a unei mulţimi S de elemente,
   - un identificator pentru un anumit tip de aranjament combinatorial: `P` pentru permutare; `RP` pentru r-permutare; `RR` pentru r-permutare cu repetiţie; sau `S` pentru submulţime; sau `RC` pentru r-combinare cu repetiţie,

- valoarea lui r pentru tipurile de aranjament $\mathtt{RP}, \mathtt{RR}, \mathtt{RC}$,
- un aranjament de tipul respectiv

şi apoi să returneze (1) rangul lexicografic al aranjamentului respectiv şi (2) aranjamentul care urmează în ordine lexicografică după aranjamentul citit.

3. Fie $S = \{1, 2, 3, 4, 5, 6\}$.
   (a) Câte permutări are S? Câte 4-permutări are S? Câte 4-permutări cu repetiţie are S?
   (b) Câte submulţimi are S? Câte 4-combinări are S?
   (c) Ce rang lexicografic are permutarea $\langle 4, 1, 3, 2, 6, 5\rangle$?
   (d) Ce permutare a lui S are rangul lexicografic 93?
   (e) Ce rang lexicografic are 3-permutarea $\langle 3, 5, 2\rangle$ a lui S?
   (f) Ce rang lexicografic are 3-permutarea cu repetiţie $\langle 3, 5, 2\rangle$ a lui S?
   (g) Care 4-permutare cu repetiţie a lui S are rangul lexicografic 800?
   (h) Enumeraţi în ordine lexicografică primele cinci 4-combinări ale lui S.
   (i) Determinaţi 3-combinările lui S cu rang lexicografic 9 şi 10.

4. Determinaţi şi implementaţi în C++ sau Java o metodă de calcul al permutării care precede lexicografic o permutare a unei mulţimi total ordonate S. Exemplu de interacţiune cu programul:

```
Introduceti elementele multimii S: a b c d
Introduceti o permutare a lui S: c b a d
Permutarea precedenta este: c a d b
```

   unde textul albastru este cel introdus de utilizator, iar cel negru este cel afişat de program.

5. Determinaţi metode de ranking, unranking şi calcul al aranjamentului următor în ordine lexicografică pentru r-combinările cu repetiţie ale unei mulţimi total ordonate $S = \{a_1, a_2, \ldots, a_n\}$.

6. Determinaţi 4-combinarea cu repetiţie a mulţimii $\{\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}\}$ cu rangul lexicografic 19. Ce rang lexicografic are 4-combinarea cu repetiţie $\{\mathtt{A}, \mathtt{C}, \mathtt{D}, \mathtt{D}\}$?