Curs 3

Permutări cu repetiție. Combinări. Algoritmi de ordonare și generare

Octombrie 2015

Cuprins

- Algoritmi de ordonare și generare pentru permutări cu repetiție
- Reprezentarea binară a submulțimilor
  - ▷ Algoritmi de ordonare și generare
- Generarea rapidă a combinărilor
  - ▷ Coduri Gray; propertăți
- Submulțimi ordonate lexicografic
- Generarea *k*-combinărilor

Permutări cu repetiție

r-permutările cu repetiție ale unui alfabet $A = \{a_1, \ldots, a_n\}$ sunt secvențele ordonate de forma

$$\langle x_1, \ldots, x_r \rangle$$

unde $x_1, \ldots, x_r \in A$.

- Același simbol poate să apară de mai multe ori în o r-permutare cu repetiție
- Conform regulii produsului, există n^r r-permutări cu repetiție

r-permutările cu repetiție pot fi ordonate lexicografic, la fel ca și r-permutările obișnuite.

Exemplu ($A = \{a_1, a_2\}$ cu $a_1 < a_2$, $r = 3$)

r-permutare cu repetiție a lui A	rang
$\langle a_1, a_1, a_1 \rangle$	0
$\langle a_1, a_1, a_2 \rangle$	1
$\langle a_1, a_2, a_1 \rangle$	2
$\langle a_1, a_2, a_2 \rangle$	3
$\langle a_2, a_1, a_1 \rangle$	4
$\langle a_2, a_1, a_2 \rangle$	5
$\langle a_2, a_2, a_1 \rangle$	6
$\langle a_2, a_2, a_2 \rangle$	7

Definiție (Ordonare lexicografică)

$\langle x_1, \ldots, x_r \rangle < \langle y_1, \ldots, y_r \rangle$ dacă există $k \in \{1, \ldots, n\}$ astfel încât $x_k < y_k$ și $x_i = y_i$ pentru toți $1 \leq i < k$.

Fie $A = \{a_1, a_2, \ldots, a_n\}$ cu $a_1 < a_2 < \ldots < a_n$.

- Dacă definim $index(a_i) := i - 1$ pentru $1 \leq i \leq n$ și înlocuim a_i cu $index(a_i)$ în enumerarea lexicografică a r-permutărilor, avem

r-permutare cu repetiție a lui A	rang
$\langle a_1, \ldots, a_1, a_1, a_1 \rangle \leftrightarrow \langle 0, \ldots, 0, 0, 0 \rangle$	0
$\vdots$	$\vdots$
$\langle a_1, \ldots, a_1, a_1, a_n \rangle \leftrightarrow \langle 0, \ldots, 0, 0, n-1 \rangle$	$n-1$
$\langle a_1, \ldots, a_1, a_2, a_1 \rangle \leftrightarrow \langle 0, \ldots, 0, 1, 0 \rangle$	n
$\vdots$	$\vdots$
$\langle a_1, \ldots, a_1, a_2, a_n \rangle \leftrightarrow \langle 0, \ldots, 0, 1, n-1 \rangle$	$2\,n-1$
$\vdots$	$\vdots$

OBSERVAȚIE: r-permutarea cu repetiție a indecșilor este reprezentarea în baza n a rangului.

1. Să se definească un algoritm care calculează rangul unei r-permutări cu repetiție $\langle x_1, \ldots, x_r \rangle$ a lui $A = \{1, \ldots, n\}$ în raport cu ordinea lexicografică.
2. Să se definească un algoritm care calculează r-permutarea cu repetiție a lui $A = \{1, \ldots, n\}$ care are rangul k în raport cu ordinea lexicografică.
3. Să se definească un algoritm care calculează r-permutarea cu repetiție care urmează imediat după r-permutarea cu repetiție $\langle x_1, \ldots, x_r \rangle$ a lui A.

O r-combinare a unei mulțimi A cu n elemente este o submulțime cu r elemente a lui A.

Există o correspondență bijectivă între șirurile de n-biți și submulțimile mulțimii $A = \{a_1, a_2, \ldots, a_n\}$:

$$B \subseteq A \mapsto b_{n-1}b_{n-2}\ldots b_0 \quad \text{unde } b_i = \begin{cases} 1 & \text{dacă } a_{n-i} \in B \\ 0 & \text{în caz contrar.} \end{cases}$$

șir de n-biți $b_0b_1\ldots b_{n-1} \mapsto$ submulțimea $\{a_{n-i} \mid b_i = 1\}$ lui A

Exemplu

$A = \{a_1, a_2, a_3, a_4, a_5\}$ unde $a_1 = a, a_2 = b, a_3 = c, a_4 = d, a_5 = e$.

$$\begin{array}{rrr}
\emptyset \leftrightarrow 00000 & \{a,b\} \leftrightarrow 00011 & \{c,d,e\} \leftrightarrow 11100 \\
\{a\} \leftrightarrow 00001 & \{a,c\} \leftrightarrow 00101 & \{b,c,d,e\} \leftrightarrow 11110 \\
\{b\} \leftrightarrow 00010 & \{a,d\} \leftrightarrow 01001 & \{a,b,d,e\} \leftrightarrow 11011 \\
\{c\} \leftrightarrow 00100 & \{a,e\} \leftrightarrow 10001 & \{a,c,d,e\} \leftrightarrow 11101 \\
\{d\} \leftrightarrow 01000 & \{b,c\} \leftrightarrow 00110 & \{a,b,c,d\} \leftrightarrow 01111 \\
\{e\} \leftrightarrow 10000 & \ldots & \{a,b,c,d,e\} \leftrightarrow 11111
\end{array}$$

```
BitString(B: submultime a lui A,
          A:  multime ordonata {a_1, ..., a_n})
int bit_string[0 .. n − 1]
for i:=0 to n − 1 do
    if a_i ∈ B then
       bit_string[n − i] := 1
    else
       bit_string[n − i] := 0
return bit_string
```

Calculului combinării corespunzătoare unui șir de n-biți

```
Combination(b[0..n-1]: sir de biti,
            A: multime ordonata {a_1,...,a_n})
B:=∅
for i:=0 to n-1 do
    if b[i] = 1 then
       adauga a_{n-i} la B
return B
```

Ordonarea submulțimilor folosind codificări cu șiruri binare

Există o corespondență bijectivă între codificările cu șiruri binare de n-biți și numerele cuprinse între 0 și $2^n - 1$:

- șir de $n-$biți $b[0\,..\,n-1] \mapsto$ numărul $\displaystyle\sum_{i=0}^{n-1} b[i] \cdot 2^i$
- număr $0 \leq r < 2^n \mapsto$ șirul de n-biți $b[0\,..\,n-1]$ unde

$$b[i] := \left\lfloor \frac{c_i}{2^i} \right\rfloor \text{ unde } c_i \text{ este restul împărțirii lui } r \text{ cu } 2^{i+1}.$$

Definiție

Rangul unei submulțimi B a mulțimii ordonate A cu n elemente este

$$\text{Rank}(B, A) := \sum_{i=0}^{n-1} b[i] \cdot 2^i$$

unde $b[0\,..\,n-1]$ este codificarea cu șir de n-biți a lui B ca submulțime a lui A.

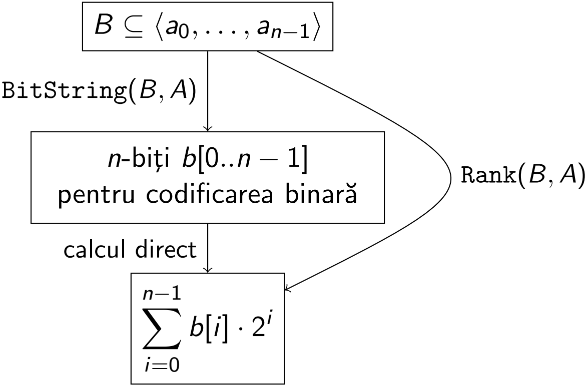
Ordonarea submulțimilor unei mulțimi prin intermediul șirurilor de biți

Exemplu ($A = \{a_0, a_1, a_2\}$)

submulțime	codificare binară $b_2 b_1 b_0$	rang
$\emptyset$	000	0
$\{a_0\}$	001	1
$\{a_1\}$	010	2
$\{a_0, a_1\}$	011	3
$\{a_2\}$	100	4
$\{a_0, a_2\}$	101	5
$\{a_1, a_2\}$	110	6
$\{a_0, a_1, a_2\}$	111	7

OBSERVAȚIE. Acest mod de enumerare a submulțimilor unei mulțimi de numește ordonare canonică, iar șirul $b_2 b_1 b_0$ se numește cod canonic (sau binar).

Ordonarea submulțimilor cu ajutorul șirurilor de biți (2)



Enumerarea submulțimilor prin intermediul reprezentării binare

Se dă o mulțime ordonată $A = \{a_0, a_1, \ldots, a_{n-1}\}$, și $0 \leq r < 2^n$

Să se determine submulțimea B a lui A cu rangul r

$$
\begin{tikzcd}
\boxed{0 \leq r < 2^n} \arrow[d] \arrow[dd, bend left=60, "\texttt{Unrank}(A, r)"] \\
\boxed{\begin{array}{c} n\text{-bit string encoding } b[0\,..\,n-1] \\ b[i] := \lfloor c/2^i \rfloor \text{ unde } c := r \mod 2^{i+1} \\ \text{pentru } 0 \leq i < n \end{array}} \arrow[d, "\texttt{Combination}(b, A)"'] \\
\boxed{B := \{a_i \mid b[n-i-1] = 1\}}
\end{tikzcd}
$$

Enumerarea submulțimilor prin intermediul reprezentării binare

Problema 1:

Se dă o submulțime B a mulțimii ordonate A.

Să se determine submulțimea lui A care urmează după B în ordonarea prin intermediul reprezentării binare.

SUGESTIE: Se poate defini `NextBinaryRankSubset(A,B)` folosind algoritmii anteriori de ordonare și enumerare.

Enumerarea submulțimilor prin intermediul reprezentării binare

Problema 1:

Se dă o submulțime B a mulțimii ordonate A.

Să se determine submulțimea lui A care urmează după B în ordonarea prin intermediul reprezentării binare.

SUGESTIE: Se poate defini `NextBinaryRankSubset(A,B)` folosind algoritmii anteriori de ordonare și enumerare.

Problema 2: Să se genereze lista tuturor submulțimilor lui A în ordinea crescătoare a rangului calculat prin intermediul codificării binare.

SUGESTIE: Se poate folosi funcția $\mathtt{Unrank}(A, r)$ definită anterior.

- Frank Grey a descoperit în 1953 o metodă de enumerare a tuturor submulţimilor unei mulţimi într-o ordine în care submulţimile consecutive diferă prin inserarea sau ştergerea unui singur element.
- Această schemă de enumerare se numeşte cod Grey reflectat standard.

Exemplu

Folosind metoda lui Grey, submulţimile of $\{a, b, c\}$ sunt enumerate în ordinea următoare:

$$\{\}, \{c\}, \{b, c\}, \{b\}, \{a, b\}, \{a, b, c\}, \{a, c\}, \{a\}$$

Codificările ca şiruri binare $b_0 b_1 b_2$ ale acestor submulţimi sunt:

$$000, 100, 110, 010, 011, 111, 101, 001$$

Se dorește enumerarea submulțimilor mulțimii $A = \{a_0, a_1, \ldots, a_{n-1}\}$ într-o ordine cu modificări minime între submulțimi consecutive. Fie G_n lista respectivă.

Procedăm recursiv:

1. Se calculează lista G_{n-1} a submulțimilor lui $B = \{a_1, \ldots, a_{n-1}\}$ într-o ordine Grey cu modificări minime.
2. Fie G'_{n-1} lista obțintă prin adăugarea lui a_0 la fiecare element al unei copii inversate a listei G_{n-1}.
3. G_n este rezultatul concatenării listei G_{n-1} cu G'_{n-1}.

Propertăți ale codurilor Grey reflectate

Se presupune că B este submulțime a mulțimii ordonate A cu n elemente.
Dacă

- m este rangul lui B in în ordinea enumerării lui Grey, și $m = \sum_{i=0}^{n-1} b_i \cdot 2^i$
- Codificarea ca șir de n biți a lui B este $c_0 c_1 \dots c_{n-1}$

atunci

- $c_i = (b_i + b_{i+1}) \mod 2$ for all $0 \leq i < n$, unde $b_n = 0$.
- Reciproc, se poate demonstra că

$$b_i = (c_i + c_{i+1} + \dots + c_{n-1}) \mod 2 \quad \text{pentru toți } 0 \leq i < n.$$

Exemplu ($A = \{a, b, c\}$ cu $a < b < c$)

submulțime B	rang Grey m	$b_0 b_1 b_2$ astfel încât $m = \sum_{i=0}^{2} b_{2-i} 2^i$	șir de biți al lui B $c_0 c_1 c_2$	rang al lui B
$\{\}$	0	000	000	0
$\{c\}$	1	100	100	4
$\{b, c\}$	2	010	110	6
$\{b\}$	3	110	010	2
$\{a, b\}$	4	001	011	3
$\{a, b, c\}$	5	101	111	7
$\{a, c\}$	6	011	101	5
$\{a\}$	7	111	001	1

Se observă că $c_i = (b_i + b_{i+1}) \mod 2$ pentru toți $0 \leq i < 3$, unde $b_3 = 0$.

Exerciții

1. Folosiți ecuațiile de pe slide-ul precedent ca să implementați metodele de ordonare `RankGrey(B,A)` și de enumerare `UnrankGrey(A,r)` pentru enumerarea submulțimilor bazată pe coduri Grey.
2. Să se definescă metoda `NextGreyRankSubset(A,B)` care calculează submulțimea lui A care urmează imediat după submulțimea B în enumerarea submulțimilor bazată pe coduri Grey.

Se dă o mulțime ordonată A cu n elemente și $0 \leq k \leq n$.

Să se genereze toate k-combinările lui A.

Se dă o mulțime ordonată A cu n elemente și $0 \leq k \leq n$.

Să se genereze toate k-combinările lui A.

Metoda 1 (naivă și ineficientă): generare și testare

1. Se generează toate cele 2^n submulțimi ale lui A
2. Se elimină submulțimile generate care nu au k elemente.

Se dă o mulțime ordonată A cu n elemente și $0 \leq k \leq n$.
Să se genereze toate k-combinările lui A.

Metoda 1 (naivă și ineficientă): generare și testare

1. Se generează toate cele 2^n submulțimi ale lui A
2. Se elimină submulțimile generate care nu au k elemente.

Metoda 2 (recursie simplă): Dacă $A = \{a\} \cup B$ unde $a \notin B$ este cel mai mic element al lui A atunci

1. Generează lista L_1 a tuturor $(k-1)$-combinărilor lui B, și fie L_2 lista tuturor k-combinărilor lui B.
2. Fie L_3 lista ce se obține adăugând a la toate elementele lui L_1.
3. Returnează rezultatul concatenării listelor L_2 și L_3.

Ordonarea lexicografică a k-combinărilor
Enunțul problemei. Observații preliminare (1)

Se presupune că $A = \{1, 2, \ldots, n\}$ și $X = \{x_1, x_2, \ldots, x_k\} \subseteq A$ astfel încât $x_1 < x_2 < \ldots < x_k$.

Î: Care este rangul lui X în enumerarea lexicografică a k-combinărilor lui A?

k-combinările care apar înaintea lui X în ordine lexicografică sunt de 2 feluri:

1. Cele care conțin un element mai mic decât x_1.
2. Cele al căror element minim este x_1, dar restul elementelor este o $(k-1)$-combinare mai mică decât $\{x_2, x_3, \ldots, x_k\}$.

Se presupune că $A = \{1, 2, \ldots, n\}$ și $X = \{x_1, x_2, \ldots, x_k\} \subseteq A$ astfel încât $x_1 < x_2 < \ldots < x_k$.

Î: Care este rangul lui X în enumerarea lexicografică a k-combinărilor lui A?

k-combinările care apar înaintea lui X în ordine lexicografică sunt de 2 feluri:

1. Cele care conțin un element mai mic decât x_1.
2. Cele al căror element minim este x_1, dar restul elementelor este o $(k-1)$-combinare mai mică decât $\{x_2, x_3, \ldots, x_k\}$.

$\Rightarrow$ rangul lui X în enumerarea lexicografică a k-combinărilor lui A este $N_1 + N_2$ unde

- N_1 este numărul k-combinărilor de primul fel
- N_2 este numărul k-combinărilor de al doilea fel

IPOTEZĂ: $A = \{1, 2, \ldots, n\}$.

Cum putem calcula N_1?

IPOTEZĂ: $A = \{1, 2, \ldots, n\}$.

Cum putem calcula N_1?

- Numărul k-combinărilor lui A care au pe i cel mai mic element este

IPOTEZĂ: $A = \{1, 2, \ldots, n\}$.

Cum putem calcula N_1?

- Numărul k-combinărilor lui A care au pe i cel mai mic element este $\binom{n-i}{k-1}$

IPOTEZĂ: $A = \{1, 2, \ldots, n\}$.

Cum putem calcula N_1?

- Numărul k-combinărilor lui A care au pe i cel mai mic element este $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (regula sumei)

IPOTEZĂ: $A = \{1, 2, \ldots, n\}$.

Cum putem calcula N_1?

- Numărul k-combinărilor lui A care au pe i cel mai mic element este $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (regula sumei)
  Știm că $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ (vezi curs 1)

IPOTEZĂ: $A = \{1, 2, \ldots, n\}$.

Cum putem calcula N_1?

- Numărul k-combinărilor lui A care au pe i cel mai mic element este $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (regula sumei)
  Știm că $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ (vezi curs 1)
  $\Rightarrow N_1 = \sum_{i=1}^{x_1-1} \left(\binom{n-i+1}{k} - \binom{n-i}{k} \right) = \binom{n}{k} - \binom{n-x_1+1}{k}$

Cum putem calcula N_2?

IPOTEZĂ: $A = \{1, 2, \ldots, n\}$.

Cum putem calcula N_1?

- Numărul k-combinărilor lui A care au pe i cel mai mic element este $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (regula sumei)
  Știm că $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ (vezi curs 1)
  $\Rightarrow N_1 = \sum_{i=1}^{x_1-1} \left(\binom{n-i+1}{k} - \binom{n-i}{k} \right) = \binom{n}{k} - \binom{n-x_1+1}{k}$

Cum putem calcula N_2?

- N_2 este rangul lui $\{x_2, \ldots, x_k\}$ în enumerarea lexicografică a $(k-1)$-combinărilor lui $\{x_1 + 1, x_1 + 2, \ldots, n-1, n\}$

IPOTEZĂ: $A = \{1, 2, \ldots, n\}$.

Cum putem calcula N_1?

- Numărul k-combinărilor lui A care au pe i cel mai mic element este $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (regula sumei)
  Știm că $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ (vezi curs 1)
  $\Rightarrow N_1 = \sum_{i=1}^{x_1-1} \left(\binom{n-i+1}{k} - \binom{n-i}{k} \right) = \binom{n}{k} - \binom{n-x_1+1}{k}$

Cum putem calcula N_2?

- N_2 este rangul lui $\{x_2, \ldots, x_k\}$ în enumerarea lexicografică a $(k-1)$-combinărilor lui $\{x_1+1, x_1+2, \ldots, n-1, n\}$
- $\Rightarrow N_2$ se poate calcula recursiv.

Din observațiile anterioare rezultă următoarea implementare recursivă a operației de calcul al rangului:

- $\texttt{RankKSubset}(\{x_1, \ldots, x_k\}, \{\ell, \ldots, n\})$ calculează rangul în ordine lexicografică a k-combinării $\{x_1, \ldots, x_k\}$ a mulțimii ordonate $\{\ell, \ell + 1, \ldots, n - 1, n\}$. Se presupune că $x_1 < x_2 < \ldots < x_k$.

RankKSubset($\{x_1, \ldots, x_k\}$, $\{\ell, \ell + 1, \ldots, n\}$)
    if ($n = k$ **or** k=0)
        return 0,
        $p := x_1 - \ell + 1$
        if ($k = 1$)
            return $p - 1$
        else
            return $\binom{n}{k} - \binom{n-p+1}{k} + \texttt{RankKSubset}(\{x_2, \ldots, x_k\}, \{x_1 + 1, \ldots, n\})$

Ipoteze:

- $A = \{1, 2, \ldots, n\}$ și $X = \{x_1, x_2, \ldots, x_k\}$ cu $x_1 < x_2 < \ldots < x_k$ este submulțimea lui A cu rangul m în enumerarea lexicografică a tuturor k-combinărilor lui A.
[Reținem că $0 \leq m < \binom{n}{k}$.]

Î: Care sunt valorile lui $x_1, x_2, \ldots, x_k$?

1. Numărul total al k-combinărilor lui A care conțin un element $< x_1$ este

$$\sum_{i=1}^{x_1-1} \binom{n-i}{k-1} = \binom{n}{k} - \binom{n-x_1+1}{k} \leq m. \qquad (1)$$

unde $\binom{n-i}{k-1}$ este numărul k-combinărilor în care cel mai mic element este $i \in \{1, \ldots, x_1 - 1\}$. Acest număr este $\leq m$ fiindcă toate aceste k-combinări sunt lexicografic mai mici decât X, care are rangul m.

1. Numărul total al k-combinărilor lui A care conțin un element $< x_1$ este

$$\sum_{i=1}^{x_1-1} \binom{n-i}{k-1} = \binom{n}{k} - \binom{n-x_1+1}{k} \leq m. \tag{1}$$

unde $\binom{n-i}{k-1}$ este numărul k-combinărilor în care cel mai mic element este $i \in \{1, \ldots, x_1 - 1\}$. Acest număr este $\leq m$ fiindcă toate aceste k-combinări sunt lexicografic mai mici decât X, care are rangul m.

2. Numărul total al k-combinărilor lui A care conțin un element $\leq x_1$ este

$$\sum_{i=1}^{x_1} \binom{n-i}{k-1} = \binom{n}{k} - \binom{n-x_1}{k} > m. \tag{2}$$

unde $\binom{n-i}{k-1}$ este numărul k-combinărilor în care cel mai mic element este $i \in \{1, \ldots, x_1\}$. Acest număr este $> m$ deoarece sunt $m+1$ întregi i între 0 și rangul lui X (care este m), și toate k-combinările cu un astfel de rang i conțin un element $\leq x_1$.

1. Numărul total al k-combinărilor lui A care conțin un element $< x_1$ este

$$\sum_{i=1}^{x_1-1} \binom{n-i}{k-1} = \binom{n}{k} - \binom{n-x_1+1}{k} \leq m. \qquad (1)$$

   unde $\binom{n-i}{k-1}$ este numărul k-combinărilor în care cel mai mic element este $i \in \{1, \ldots, x_1 - 1\}$. Acest număr este $\leq m$ fiindcă toate aceste k-combinări sunt lexicografic mai mici decât X, care are rangul m.

2. Numărul total al k-combinărilor lui A care conțin un element $\leq x_1$ este

$$\sum_{i=1}^{x_1} \binom{n-i}{k-1} = \binom{n}{k} - \binom{n-x_1}{k} > m. \qquad (2)$$

   unde $\binom{n-i}{k-1}$ este numărul k-combinărilor în care cel mai mic element este $i \in \{1, \ldots, x_1\}$. Acest număr este $> m$ deoarece sunt $m + 1$ întregi i între 0 și rangul lui X (care este m), și toate k-combinările cu un astfel de rang i conțin un element $\leq x_1$.

$\Rightarrow$ putem folosi (1) și (2) ca să aflăm x_1: $\binom{n}{k} - \binom{n-x_1+1}{k} \leq m < \binom{n}{k} - \binom{n-x_1}{k}$

1. Numărul total al k-combinărilor lui A care conțin un element $< x_1$ este

$$\sum_{i=1}^{x_1-1} \binom{n-i}{k-1} = \binom{n}{k} - \binom{n-x_1+1}{k} \leq m. \quad (1)$$

   unde $\binom{n-i}{k-1}$ este numărul k-combinărilor în care cel mai mic element este $i \in \{1, \ldots, x_1 - 1\}$. Acest număr este $\leq m$ fiindcă toate aceste k-combinări sunt lexicografic mai mici decât X, care are rangul m.

2. Numărul total al k-combinărilor lui A care conțin un element $\leq x_1$ este

$$\sum_{i=1}^{x_1} \binom{n-i}{k-1} = \binom{n}{k} - \binom{n-x_1}{k} > m. \quad (2)$$

   unde $\binom{n-i}{k-1}$ este numărul k-combinărilor în care cel mai mic element este $i \in \{1, \ldots, x_1\}$. Acest număr este $> m$ deoarece sunt $m+1$ întregi i între 0 și rangul lui X (care este m), și toate k-combinările cu un astfel de rang i conțin un element $\leq x_1$.

$\Rightarrow$ putem folosi (1) și (2) ca să aflăm x_1: $\binom{n}{k} - \binom{n-x_1+1}{k} \leq m < \binom{n}{k} - \binom{n-x_1}{k}$

Celelalte elemente $x_2, \ldots, x_k$ se pot determina recursiv.

Enumerarea lexicografică a k-combinărilor

`UnrankKSubset`$(m, k, \{a_1, \ldots, a_n\})$ produce k-combinarea $\{x_1, \ldots, x_k\}$ cu rangul m a lui $\{a_1, \ldots, a_n\}$ în ordine lexicografică. Se presupune că $x_1 < \ldots < x_k$ și $a_1 < \ldots < a_n$.

`UnrankKSubset`$(m, k, \{a_1, \ldots, a_n\})$
if $(k = 1)$
    return a_{k+1}
else if $(m = 0)$
    return $\{a_1, \ldots, a_m\}$
else
    $u := \binom{n}{k}$
    $i := 1$
    while $\binom{i}{k} < u - m$
        i++
    $x1$:=$n - (i - 1)$
    return $\{a_{n-i+1}\} \cup$ `UnrankKSubset`$(m - u + \binom{n-x1+1}{k}, k - 1, \{a_{n-i+2}, \ldots, a_n\})$

Bibliografie

- S. Pemmaraju, S. Skiena. *Combinatorics and Graph Theory with Mathematica*. Section 2.3: Combinations. Cambridge University Press. 2003.