Lecture 3

Permutations with repetition. Combinations. Enumeration, ranking and unranking algorithms

October 2014

Cuprins

- Enumeration, ranking and unranking algorithms for permutations with repetition
- Binary represention of subsets
  - Ranking and unranking algorithms
- Fast generation of all subsets
  - Gray codes; properties
- Lexicographically ordered combinations (or subsets)
- r-combinations: ranking and unranking algorithms

Permutations with repetition

The *r-permutations with repetition* of an alphabet $A = \{a_1, \ldots, a_n\}$ are the ordered sequences of symbols of the form

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

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

- The same symbol of A can occur many times
- By the rule of product, there are n^r r-permutations with repetition

The r-permutations with repetition can be ordered lexicographically:

- $\langle x_1, \ldots, x_r \rangle < \langle y_1, \ldots, y_r \rangle$ if there exists $k \in \{1, \ldots, n\}$ such that $x_k < y_k$ and $x_i = y_i$ for all $1 \leq i < k$.

Example ($A = \{a_1, a_2\}$ with $a_1 < a_2$, and $r = 3$)

r-permutation with repetition of A	lexicographic rank
$\langle a_1, a_1, a_1 \rangle$	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

Let $A = \{a_1, a_2, \ldots, a_n\}$ with $a_1 < a_2 < \ldots < a_n$.

- If we define $index(a_i) := i - 1$ for $1 \leq i \leq n$, and replace a_i with $index(a_i)$ in the lexicographic enumeration of the r-permutations, we get

r-permutation with repetition	encoding as number in base n	lexicogaphic rank
$\langle a_1, \ldots, a_1, a_1, a_1 \rangle$	$\langle 0, \ldots, 0, 0, 0 \rangle$	0
$\vdots$	$\vdots$	$\vdots$
$\langle a_1, \ldots, a_1, a_1, a_n \rangle$	$\langle 0, \ldots, 0, 0, n-1 \rangle$	$n-1$
$\langle a_1, \ldots, a_1, a_2, a_1 \rangle$	$\langle 0, \ldots, 0, 1, 0 \rangle$	n
$\vdots$	$\vdots$	$\vdots$
$\langle a_1, \ldots, a_1, a_2, a_n \rangle$	$\langle 0, \ldots, 0, 1, n-1 \rangle$	$2\,n-1$
$\vdots$	$\vdots$	$\vdots$

REMARK: The r-permutation with repetition of the indexes is the representation in base n of its lexicographic rank.

1. Define an algorithm which computes the rank of the r-permutation with repetition $\langle x_1, \ldots, x_r \rangle$ of $A = \{1, \ldots, n\}$ with respect to the lexicographic order.
2. Define an algorithm which computes r-permutation with repetition $\langle x_1, \ldots, x_r \rangle$ with rank k of $A = \{1, \ldots, n\}$ with respect to the lexicographic order.
3. Define an algorithm which computes the r-permutation with repetition immediately after the r-permutation with repetition $\langle x_1, \ldots, x_r \rangle$ of A, in lexicographic order.

An r-combination of a set $A = \{a_1, a_2, \ldots, a_n\}$ is a subset with r elements of A.

There is a bijective correspondence between the set of n-bit strings and the set of subsets of A:

$$B \subseteq A \mapsto b_{n-1} b_{n-2} \ldots b_0 \quad \text{where } b_i = \begin{cases} 1 & \text{if } a_{n-i} \in B \\ 0 & \text{otherwise.} \end{cases}$$

n-bit string $b_0 b_1 \ldots b_{n-1} \mapsto$ subset $\{a_{n-i} \mid b_i = 1\}$ of A

Example ($A = \{a, b, c, d, e\}$ with $a > b > c > d > e$.)

subset	n-bit string encoding *edcba*	canonic rank
$\emptyset$	00000	0
$\{a\}$	00001	1
$\{b\}$	00010	2
$\{a, b\}$	00011	3
$\vdots$	$\vdots$	$\vdots$

The n-bit string encoding of a subset

```
BitString(B: subset of A,
          A:  ordered set {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
```

The subset of an n-bit string encoding

```
Combination(b[0..n-1]: bit string,
            A: ordered set {a_1,...,a_n})
B:=∅
for i:=0 to n-1 do
    if b[i] = 1 then
       add a_{n-i} to B
return B
```

The ordering of combinations via bit string encodings

There is a bijective correspondence between the n-bit string encodings and the numbers from 0 to $2^n - 1$:

- n-bit-string $b[0\,..\,n-1] \mapsto$ number $\displaystyle\sum_{i=0}^{n-1} b[i]\cdot 2^i \in \{0,1,\ldots,2^n-1\}$
- number $0 \leq r < 2^n \mapsto$ n-bit-string $b[0\,..\,n-1]$ where

$$b[i] := \left\lfloor \frac{c_i}{2^i} \right\rfloor$$ where c_i is the remainder of dividing r with 2^{i+1}.

Definition

The canonic rank of a a subset B of an ordered set A with n elements is

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

where $b[0\,..\,n-1]$ is the n-bit-string encoding of B as subset of A.

The ordering of combinations via bit string encodings (2)

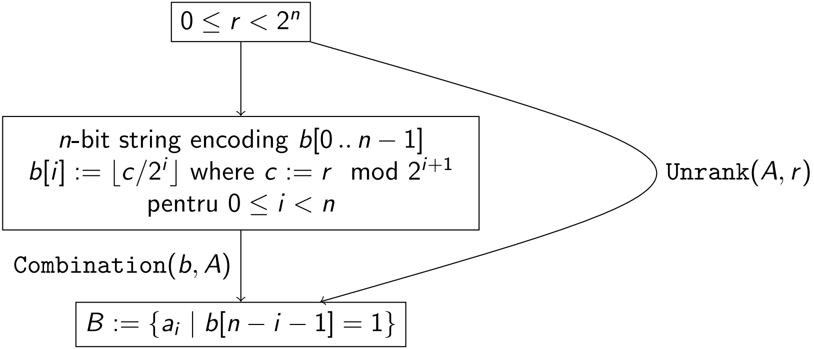
REMARK. This way of enumerating the subsets of a set is called canonic ordering, and the n-bit string $b_{n-1} \ldots, b_1 b_0$ is called canonic (or binary) code.

$$
\begin{tikzcd}
\boxed{B \subseteq \langle a_0, \ldots, a_{n-1} \rangle} \arrow[d, "\texttt{BitString}(B, A)"'] \arrow[dd, bend left=60, "\texttt{CanonicRank}(B, A)"] \\
\boxed{\begin{array}{c} n\text{-bits } b[0..n-1] \\ \text{for the binary encoding of } B \end{array}} \arrow[d, "\text{direct computation}"'] \\
\boxed{\displaystyle\sum_{i=0}^{n-1} b[i] \cdot 2^i}
\end{tikzcd}
$$

The ordering of combinations via bit string encodings (3)

Given an ordered set $A = \{a_0, a_1, \ldots, a_{n-1}\}$, and $0 \leq r < 2^n$

Find the subset B of A with rank r



- Frank Grey discovered in 1953 a method to enumerate subsets in an order so that adjacent subsets differ by the insertion or deletion of only one element.
- His enumeration scheme is called standard reflected Grey code.

Example

With Grey's method, the subsets of $\{a, b, c\}$ are enumerated in the following order:

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

The 3-bit-string encodings of these subsets are

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

We want to enumerate the subsets of $A = \{a_1, \ldots, a_n\}$ in minimum change order G_n. (G_n is the list of those subsets)

We want to enumerate the subsets of $A = \{a_1, \ldots, a_n\}$ in minimum change order G_n. (G_n is the list of those subsets)

We proceed recursively:

1. Compute the list G_{n-1} of subsets of $B = \{a_2, \ldots, a_n\}$ in the minimum change order of Gray.
2. Let G'_{n-1} be the list of subsets obtained by adding a_1 to every element of a reversed copy fo G_{n-1}.
3. G_n is the concatenation of G_{n-1} with 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 $c_0 c_1 \ldots c_{n-1}$

atunci

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

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

Example ($A = \{a, b, c\}$ 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 generez 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 $A = \{1, 2, \dots, n\}$ și $X = \{x_1, x_2, \dots, x_k\} \subseteq A$ astfel încât $x_1 < x_2 < \dots < x_k$.

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 $\{x_2, x_3, \dots, x_k\}$.

$\Rightarrow$ 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 $x_1 < x_2 < \ldots < x_k$.

$\texttt{RankKSubset}(\{x_1, \ldots, x_k\},\ \{\ell, \ell+1, \ldots, n\})$
  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

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

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.

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

$\Rightarrow$ putem folosi (**??**) și (**??**) 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
$$\sum_{i=1}^{x_1-1}\binom{n-i}{k-1}=\binom{n}{k}-\binom{n-x_1+1}{k}\leq m. \qquad (1)$$
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 (**??**) și (**??**) 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`$(m - u + \binom{n-x1+1}{k}, k - 1, \{a_{n-i+2}, \ldots, a_n\})$

References

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