Lecture 3

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

October 2014

Outline

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

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

Definition

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

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

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

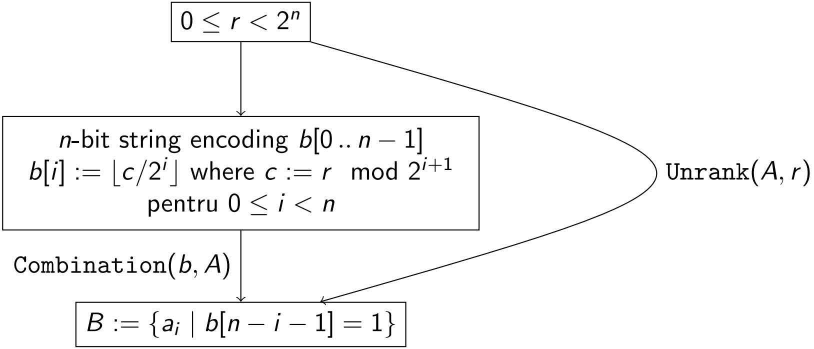
REMARK. This way of enumerating the subsets of a set is called canonic ordering, and the n-bit string $b_{n-1} \ldots, b_1 b_0$ is called canonic (or binary) code.

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

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}.

Properties of Grey's codes reflected

Assume that B is a subset of the ordered set A with n elements. If

- m is the rank of B in the order of the Grey's enumeration and $m = \sum_{i=0}^{n-1} b_i \cdot 2^i$
- The codification as a n bit string of B is $c_0 c_1 \dots c_{n-1}$

then

- $c_i = (b_i + b_{i+1}) \mod 2 \quad \text{for all } 0 \leq i < n, \text{ where } b_n = 0.$
- On the other hand, one can prove that

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

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

subset B	Grey rank m	$b_0 b_1 b_2$ such that $m = \sum_{i=0}^{2} b_{2-i} 2^i$	bit string of B $c_0 c_1 c_2$	rank of 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

Notice that $c_i = (b_i + b_{i+1}) \mod 2$ for all $0 \leq i < 3$, where $b_3 = 0$.

Exercises

1. Use the ecuations in the previous slide to implement the ordering method `RankGrey(B,A)` and the enumeration method `UnrankGrey(A,r)` for enumerationg the subsets based on Grey's codes.
2. Define the method `NextGreyRankSubset(A,B)` which computes the subset of A which is the immediately next one after the subset B in the enumeration of subsets based on Grey's codes.

Given an ordered set A with n elements and $0 \leq k \leq n$.

Generate all the k-combinations of A.

Given an ordered set A with n elements and $0 \leq k \leq n$.

Generate all the k-combinations of A.

Method 1 (naive and inefficient): generate and test

1. Generate all the 2^n subsets of A
2. Eliminate the generated subsets which do not have k elements.

Given an ordered set A with n elements and $0 \leq k \leq n$.

Generate all the k-combinations of A.

Method 1 (naive and inefficient): generate and test

1. Generate all the 2^n subsets of A
2. Eliminate the generated subsets which do not have k elements.

Method 2 (simple recursion): If $A = \{a\} \cup B$ where $a \notin B$ is the smallest element of A then

1. Generate the list L_1 of all $(k-1)$-combinations of B, and let L_2 be the list of all k-combinations of B.
2. Let L_3 be the list obtained by adding a to all the elements of L_1.
3. Return the result of the concatenation of L_2 with L_3.

Assume $A = \{1, 2, \ldots, n\}$ and $X = \{x_1, x_2, \ldots, x_k\} \subseteq A$ such that $x_1 < x_2 < \ldots < x_k$.

Q: Which is the rank of X in the lexicographic enumeration of the k-combinations of A?

The k-combinations which occur before X in lexicographic order are of 2 kinds:

1. The ones which contain an element smaller than x_1.
2. The ones which contain the minimum element x_1, but the rest of the elements is a $(k-1)$-combination smaller than $\{x_2, x_3, \ldots, x_k\}$.

Assume $A = \{1, 2, \ldots, n\}$ and $X = \{x_1, x_2, \ldots, x_k\} \subseteq A$ such that $x_1 < x_2 < \ldots < x_k$.

Q: Which is the rank of X in the lexicographic enumeration of the k-combinations of A?

The k-combinations which occur before X in lexicographic order are of 2 kinds:

1. The ones which contain an element smaller than x_1.
2. The ones which contain the minimum element x_1, but the rest of the elements is a $(k-1)$-combination smaller than $\{x_2, x_3, \ldots, x_k\}$.

$\Rightarrow$ the rank of X in the lexicographic enumeration of the k-combinations of A is $N_1 + N_2$ where

- N_1 is the number of k-combinations of the first kind
- N_2 is the number of the k-combinations of the second kind

HYPOTHESIS: $A = \{1, 2, \ldots, n\}$.

How can we compute N_1?

HYPOTHESIS: $A = \{1, 2, \ldots, n\}$.

How can we compute N_1?

- The number of k-combinations of A which contain i as the smallest element is

HYPOTHESIS: $A = \{1, 2, \ldots, n\}$.

How can we compute N_1?

- The number of k-combinations of A which contain i as the smallest element is $\binom{n-i}{k-1}$

HYPOTHESIS: $A = \{1, 2, \ldots, n\}$.

How can we compute N_1?

- The number of k-combinations of A which contain i as the smallest element is $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (the sum rule)

HYPOTHESIS: $A = \{1, 2, \ldots, n\}$.

How can we compute N_1?

- The number of k-combinations of A which contain i as the smallest element is $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (the sum rule)
  We know that $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ (see lecture 1)

HYPOTHESIS: $A = \{1, 2, \ldots, n\}$.

How can we compute N_1?

- The number of k-combinations of A which contain i as the smallest element is $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (the sum rule)
  We know that $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ (see lecture 1)
  $\Rightarrow N_1 = \sum_{i=1}^{x_1-1} \left(\binom{n-i+1}{k} - \binom{n-i}{k} \right) = \binom{n}{k} - \binom{n-x_1+1}{k}$

How can we compute N_2?

HYPOTHESIS: $A = \{1, 2, \ldots, n\}$.

How can we compute N_1?

- The number of k-combinations of A which contain i as the smallest element is $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (the sum rule)
  We know that $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ (see lecture 1)
  $\Rightarrow N_1 = \sum_{i=1}^{x_1-1} \left(\binom{n-i+1}{k} - \binom{n-i}{k} \right) = \binom{n}{k} - \binom{n-x_1+1}{k}$

How can we compute N_2?

- N_2 is the rank of $\{x_2, \ldots, x_k\}$ in the lexicographic enumeration of the $(k-1)$-combinations of $\{x_1+1, x_1+2, \ldots, n-1, n\}$

HYPOTHESIS: $A = \{1, 2, \ldots, n\}$.

How can we compute N_1?

- The number of k-combinations of A which contain i as the smallest element is $\binom{n-i}{k-1} \Rightarrow N_1 = \sum_{i=1}^{x_1-1} \binom{n-i}{k-1}$ (the sum rule)
  We know that $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$ (see lecture 1)
  $\Rightarrow N_1 = \sum_{i=1}^{x_1-1} \left(\binom{n-i+1}{k} - \binom{n-i}{k} \right) = \binom{n}{k} - \binom{n-x_1+1}{k}$

How can we compute N_2?

- N_2 is the rank of $\{x_2, \ldots, x_k\}$ in the lexicographic enumeration of the $(k-1)$-combinations of $\{x_1 + 1, x_1 + 2, \ldots, n-1, n\}$
- $\Rightarrow N_2$ can be computed recursively.

The lexicographic ordering of the k-combinations

From the previous remarks results the following recursive implementation for computing the rank:

- $\texttt{RankKSubset}(\{x_1, \ldots, x_k\}, \{\ell, \ldots, n\})$ computes the rank in lexicographic order of the k-combination $\{x_1, \ldots, x_k\}$ of the ordered set $\{\ell, \ell+1, \ldots, n-1, n\}$. Assume that $x_1 < x_2 < \ldots < x_k$.

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

Hypothesis:

- $A = \{1, 2, \ldots, n\}$ and $X = \{x_1, x_2, \ldots, x_k\}$ with $x_1 < x_2 < \ldots < x_k$ is the subset of A with rank m in the lexicographic enumeration of all k-combinations of A.
  [Keep in mind that $0 \leq m < \binom{n}{k}$.]

Q̂: Which are the values $x_1, x_2, \ldots, x_k$?

1. The total number of k-combinations of A which contain the element $< x_1$ is

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

where $\binom{n-i}{k-1}$ is the number of k-combinations in which the smallest element is $i \in \{1, \ldots, x_1 - 1\}$. This number is $\leq m$ because all these k-combinations are lexicographic smaller than X, which has the rank m.

1. The total number of k-combinations of A which contain the element $< x_1$ is
$$\sum_{i=1}^{x_1-1}\binom{n-i}{k-1}=\binom{n}{k}-\binom{n-x_1+1}{k}\leq m. \tag{1}$$
where $\binom{n-i}{k-1}$ is the number of k-combinations in which the smallest element is $i\in\{1,\ldots,x_1-1\}$. This number is $\leq m$ because all these k-combinations are lexicographic smaller than X, which has the rank m.
2. The total number of k-combinations of A which contain an element $\leq x_1$ is
$$\sum_{i=1}^{x_1}\binom{n-i}{k-1}=\binom{n}{k}-\binom{n-x_1}{k}>m. \tag{2}$$
where $\binom{n-i}{k-1}$ is the number of k-combinations in which the smallest element is $i\in\{1,\ldots,x_1\}$. This number is $> m$ because there are $m+1$ integers i between 0 and the rank of X (which is m), and all the k-combinations with such a rank i contain one element $\leq x_1$.

1. The total number of k-combinations of A which contain the element $< x_1$ is

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

   where $\binom{n-i}{k-1}$ is the number of k-combinations in which the smallest element is $i \in \{1, \ldots, x_1 - 1\}$. This number is $\leq m$ because all these k-combinations are lexicographic smaller than X, which has the rank m.

2. The total number of k-combinations of A which contain an element $\leq x_1$ is

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

   where $\binom{n-i}{k-1}$ is the number of k-combinations in which the smallest element is $i \in \{1, \ldots, x_1\}$. This number is $> m$ because there are $m+1$ integers i between 0 and the rank of X (which is m), and all the k-combinations with such a rank i contain one element $\leq x_1$.

$\Rightarrow$ one can use (1) and (2) to find x_1: $\binom{n}{k} - \binom{n-x_1+1}{k} \leq m < \binom{n}{k} - \binom{n-x_1}{k}$

1. The total number of k-combinations of A which contain the element $< x_1$ is

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

where $\binom{n-i}{k-1}$ is the number of k-combinations in which the smallest element is $i \in \{1, \ldots, x_1 - 1\}$. This number is $\leq m$ because all these k-combinations are lexicographic smaller than X, which has the rank m.

2. The total number of k-combinations of A which contain an element $\leq x_1$ is

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

where $\binom{n-i}{k-1}$ is the number of k-combinations in which the smallest element is $i \in \{1, \ldots, x_1\}$. This number is $> m$ because there are $m+1$ integers i between 0 and the rank of X (which is m), and all the k-combinations with such a rank i contain one element $\leq x_1$.

$\Rightarrow$ one can use (1) and (2) to find x_1: $\binom{n}{k} - \binom{n-x_1+1}{k} \leq m < \binom{n}{k} - \binom{n-x_1}{k}$

The other elements $x_2, \ldots, x_k$ can be computed recursively.

`UnrankKSubset`$(m, k, \{a_1, \ldots, a_n\})$ produces the k-combination $\{x_1, \ldots, x_k\}$ with rank m of $\{a_1, \ldots, a_n\}$ in lexicographic order. Assume that $x_1 < \ldots < x_k$ and $a_1 < \ldots < a_n$.

`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`$\left(m - u + \binom{n - x1 + 1}{k}, k - 1, \{a_{n-i+2}, \ldots, a_n\}\right)$

References

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