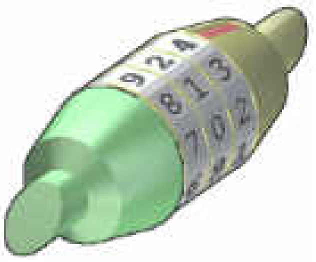
C & A

Chap. II
Permutation and Combination

Yuchun Ma(马昱春)
myc@tsinghua.edu.cn

Generating Permutations



- Can we start from the simple thing first?
  - The permutation of {1}

    1

  - The permutation of {1 2} 2 1 2

    1 2

    2 1

  - The permutation of {1 2 3} 3 1 3 2 3

    3 2 3 1 3

    1 2 3 3 2 1

    1 3 2 2 3 1

    3 1 2 2 1 3

Induction

High complexity!

Waste of memory!

Generate the permutations of {1,2,…n} based on the permutations of {1,2,…,n-1}

Lexicographic Order

123
132
213

[Eg]For alphabet {1,2,3}, smaller digits are in the left, so the permutations in lexicographic order are:

123,132,213,231,312,321.

A permutation could be regarded as a string, the string could have **prefix and suffix**。

The so called *next permutation of this one* means that there's no more *other permutations* among this and the next.

This means that this one and next one should have a **common prefix** which is as long as possible, and the changes are limited to **suffixes** as short as possible.

Generating Permutations

单选题 1分

What is the next permutation for sequence “46739521” with lexicographical order ____?

A 46791235

B 46751238

C 46751239

D 46759321

单选题 1分

设置

What is the next permutation for sequence "35418762" with lexicographical order?

A 35428761

B 35421678

C 35421678

D 35421679

提交

Lexicographic Order

- **[Eg]**Calculate the next permutation of 839647521
- 1 Find the first decrease point from right to left: 4

839647521

- 2. Exchange Find the smallest number larger than 4 in the suffix

839647521

- 3. Overturn the suffix 839651247
- The next permutation is: 839651247

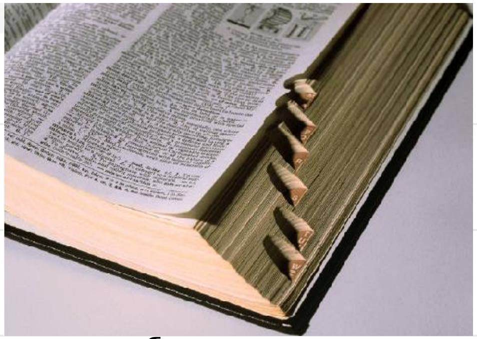
Thinking question: How many permutations are before “839647521” in lexicographical order?

作答

Thinking question: How many permutations are before “839647521” in lexicographical order?

$7 \times 8! + 2 \times 7! + 6 \times 6! + 4 \times 5! + 2 \times 4! + 3 \times 3! + 2 \times 2! + 1 \times 1!$
$= 297191$

Generating r-combinations



- Lexicographic order (dictionary order)
  - "abz" precedes "acb" in the dictionary
  - 129 precedes 132 in the lexicographic order
- Let A and B be two r-combinations of the set {1, 2, …, n}.
  - *A precedes B* in the lexicographic order provided the smallest integer which is in their union $A \cup B$ but not in their intersection $A \cap B$ is in A.
  - 5-combinations of {1, 2 ,…,8},A = {2,3,4,7,8}, B= {2, 3 ,5 ,6 ,7}.
    - Then the smallest element which is in one but not both of the sets is 4.
    - Hence A precedes B in the lexicographic order.

A theorem

- Consider 5-combinations of {1,2,…,9}. What 5-combination immediately follows 12389?
  - 12456
- Let $a_1a_2...a_r \neq$ (n-r+1)(n-r+2)…n be an r-combination of {1, 2, …, n}.
- Let k be the largest integer such that $a_k < n$ and a_k+1 is different from each of $a_1, a_2, \ldots a_r$.
- Then the r-combination which is the immediate successor of $a_1a_2...a_r$ in the lexicographic ordering is
- $a_1 \ldots a_{k-1}(a_k+1)(a_k+2)\ldots(a_k+r-k+1)$.

单选题 1分

设置

Consider 5-combinations of {1,2,…,9}. What 5-combination immediately before 34579?

A 34569

B 34578

C 34568

D 34597

提交

Algorithm for generating the r-combinations

Begin with the r-combination $a_1a_2...a_r$ =12…r. while $a_1a_2...a_r$ $\neq (n\text{-}r+1)(n\text{-}r+2)...n$, do

1) determine the largest integer k such that a_k+1 ≤ n and a_k+1 is not one of a_1, a_2, ...a_r.

2) replace $a_1a_2...a_r$ *with the r-combination*

$$a_1a_2...a_{k-1}(a_k+1)(a_k+2)\ldots(a_k+r\text{-}k+1).$$

Generate the 4-combinations of *S* = {1,2,3,4,5,6}.

1234;	1245;	1345;	1456;	2356;
1235;	1246;	1346;	2345;	2456;
1236;	1256;	1356;	2346;	3456.

Generating Permutations

- Can we start from the simple thing first?
  - The permutation of {1}

    1

  - The permutation of {1 2} 2 1 2

    1 2

    2 1

  - The permutation of {1 2 3} 3 1 3 2 3

    1 2 3   3 2 1   3 2 3 1 3

    1 3 2   2 3 1

    3 1 2   2 1 3

Induction

High complexity!

Waste of memory!

Generate the permutations of {1,2,…n} based on the permutations of {1,2,…,n-1}

Generating Permutations

– The permutation of {1 2 3 4}

1 2 3↔4	3 1 2↔4	2 3 1↔4
1 2↔4 3	3 1↔4 2	2 3↔4 1
1↔4 2 3	3 4↔1 2	2↔4 3 1
4 1 2 ↔ 3	4 3 1 ↔ 2	4 2 3 ↔ 1
4↔1 3 2	4↔3 2 1	4↔2 1 3
1 4↔3 2	3 4↔2 1	2 4↔1 3
1 3 4↔2	3 2 4↔1	2 1 4↔3
1 ↔ 3 2 4	3 ↔ 2 1 4	2 1 3 4

- Each permutation other than the first one is obtained from the preceding one by **switching two adjacent numbers**.
  - Move the largest number 4 between two ends
  - When the end is met, the largest number 4 will be fixed for a step and the switching of the second largest number 3 makes 4 movable again.

Mobile Integer

- Given an integer *k* we assign a direction to it by writing an arrow above it pointing to the left or to the right. Consider a permutation of {1,2,…,*n*} in which each of the integers is given a direction.
  - The integer *k* is called *mobile* if its arrow points to a smaller integer adjacent to it.

    $\overrightarrow{2}\ \overrightarrow{6}\ \overrightarrow{3}\ \overrightarrow{1}\ \overrightarrow{5}\ \overrightarrow{4}$ only 3, 5, and 6 are mobile.
  - Integer 1 can never be mobile since there is no integer smaller than 1.
  - The integer *n* is always mobile, except in two cases:

    i) *n* is the first integer and its arrow points to the left.

    ii) *n* is the last integer and its arrow points to the right.

$\overleftarrow{\cancel{4}}\ \overleftarrow{1}\ \overleftarrow{2}\ \overleftarrow{3}$ $\qquad\qquad$ $\overrightarrow{3}\ \overleftarrow{2}\ \overleftarrow{1}\ \overrightarrow{\cancel{4}}$

Generating Permutations

– The permutation of {1 2 3 4}

1 2 3 ↔ 4
1 2 ↔ 4 3
1 ↔ 4 2 3
4 1 2 ↔ 3
4 ↔ 1 3 2
1 4 ↔ 3 2
1 3 4 ↔ 2
1 ↔ 3 2 4

Step 0: All mobile
Step 1: Choose the largest mobile integer 4
Step 2: Switch 4 and 3 as the arrow points to.
…
…
Step i: 4 is fixed, the largest mobile integer is 3. After the switch between 3 and 2, switch the direction of the arrow above 4(4>3)

- Each permutation other than the first one is obtained from the preceding one by **switching two adjacent numbers**.
  - Move the largest number 4 between two ends
  - When the end is met, the largest number 4 will be fixed for a step and the switching of the second largest number 3 makes 4 movable again.

Steinhaus–Johnson–Trotter algorithm

Begin with $\overleftarrow{1}\ \overleftarrow{2} \cdots \overleftarrow{n}$.

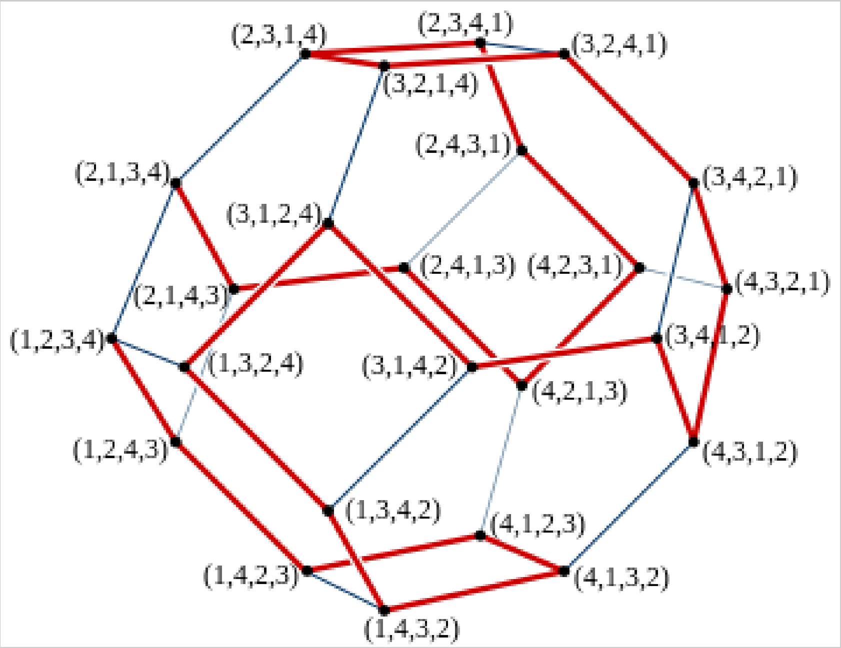
While there exists a mobile integer, do

1) find the largest mobile integer *m*.
2) switch *m* and the adjacent integer its arrow points to.
3) switch the direction of all integers *p* with *p*>*m*.

$\overleftarrow{1}\ \overleftarrow{2}\ \overleftarrow{3}\ \overleftarrow{4}$	$\overleftarrow{1}\ \overleftarrow{3}\ \overrightarrow{4}\ \overleftarrow{2}$	$\overrightarrow{4}\ \overrightarrow{3}\ \overleftarrow{2}\ \overleftarrow{1}$	$\overleftarrow{2}\ \overleftarrow{4}\ \overrightarrow{3}\ \overleftarrow{1}$
$\overleftarrow{1}\ \overleftarrow{2}\ \overleftarrow{4}\ \overleftarrow{3}$	$\overleftarrow{1}\ \overleftarrow{3}\ \overleftarrow{2}\ \overrightarrow{4}$	$\overrightarrow{3}\ \overrightarrow{4}\ \overleftarrow{2}\ \overleftarrow{1}$	$\overleftarrow{4}\ \overleftarrow{2}\ \overrightarrow{3}\ \overleftarrow{1}$
$\overleftarrow{1}\ \overleftarrow{4}\ \overleftarrow{2}\ \overleftarrow{3}$	$\overleftarrow{3}\ \overleftarrow{1}\ \overleftarrow{2}\ \overleftarrow{4}$	$\overrightarrow{3}\ \overleftarrow{2}\ \overrightarrow{4}\ \overleftarrow{1}$	$\overrightarrow{4}\ \overleftarrow{2}\ \overleftarrow{1}\ \overrightarrow{3}$
$\overleftarrow{4}\ \overleftarrow{1}\ \overleftarrow{2}\ \overleftarrow{3}$	$\overleftarrow{3}\ \overleftarrow{1}\ \overleftarrow{4}\ \overleftarrow{2}$	$\overrightarrow{3}\ \overleftarrow{2}\ \overleftarrow{1}\ \overrightarrow{4}$	$\overleftarrow{2}\ \overrightarrow{4}\ \overleftarrow{1}\ \overrightarrow{3}$
$\overrightarrow{4}\ \overleftarrow{1}\ \overleftarrow{3}\ \overleftarrow{2}$	$\overleftarrow{3}\ \overleftarrow{4}\ \overleftarrow{1}\ \overleftarrow{2}$	$\overleftarrow{2}\ \overrightarrow{3}\ \overleftarrow{1}\ \overleftarrow{4}$	$\overleftarrow{2}\ \overleftarrow{1}\ \overrightarrow{4}\ \overrightarrow{3}$
$\overleftarrow{1}\ \overrightarrow{4}\ \overleftarrow{3}\ \overleftarrow{2}$	$\overleftarrow{4}\ \overleftarrow{3}\ \overleftarrow{1}\ \overleftarrow{2}$	$\overleftarrow{2}\ \overrightarrow{3}\ \overleftarrow{4}\ \overleftarrow{1}$	$\overleftarrow{2}\ \overleftarrow{1}\ \overrightarrow{3}\ \overrightarrow{4}$

SJT Algorithm

- **Steinhaus–Johnson–Trotter algorithm**
  - Hugo Steinhaus[1958], H. F. Trotter[1962], S. M. Johnson[1963]
  - Each permutation i generates differs fr permutation by swa elements of the sec
  - Equivalently, this a a Hamiltonian path



The Hamiltonian path in the Cayley graph of the symmetric group generated by the Steinhaus–Johnson–Trotter algorithm

Inversions in Permutations

- Let $i_1 i_2 .... i_n$ be a permutation of the set {1,2, …,n}. The pair (i_k, i_l) is called an inversion if $k<l$ and $i_k>i_l$.
- An inversion in a permutation corresponds to a pair of numbers which are out of their natural order.
- The only permutation of {1,2, …,n} with no inversions is 1 2 … n.
- Example: the inversions in (3 2 1) are (3,2),(3,1),(2,1)

The number of inversions

- Let a_j denote the number of inversions. It equals the number of integers which precede j in the permutation but are greater than j; it measures how much j is out of order.
- The sequence of numbers $a_1, a_2, \ldots, a_n$ is called the inversion sequence of the permutation $i_1 i_2 \ldots. i_n$.
- The number $a_1 + a_2 + \ldots + a_n$ measures the disorder of a permutation.

Examples of inversions

- Consider the permutation 31524. It has four inversions, namely (3, 1), (3, 2), (5, 2), (5, 4).
- Inversion records the number of integers which precede *j* in the permutation but are greater than *j*
- The inversion sequence of the permutation 31524
  - 1: 3 is larger than 1 and before 1 **1**
  - 2: 3 and 5 are larger than 2 and before 2 **2**
  - 3: no number is before 3 **0**
  - 4: 5 is larger than 4 and before 4 **1**
  - 5: no number is larger than 5 **0**

A theorem

Let $b_1, b_2, \ldots, b_n$ be a sequence of integers satisfying

$$0 \le b_1 \le n-1, 0 \le b_2 \le n-2, \cdots, 0 \le b_{n-1} \le 1, b_n = 0.$$

Then there exists a unique permutation of {1, 2, …, n} whose inversion sequence is $b_1, b_2, \ldots, b_n$.

With the inversion sequence as 1, 2, 0, 1, 0, how to construct the permutation?

3 1 5 2 4

- 5: no number is larger than 5
- 4: 1 larger number before
- 3: no larger number is before 3
- 2: 2 larger numbers before
- 1: 1 larger number before

Algorithm I

- n: write down n.
- n-1: consider b_{n-1}. We are given that $0 \le b_{n-1} \le 1$

  If b_{n-1} =0, then n-1 must be placed before n. If b_{n-1} =1, then n-1 must be placed after n.
- n-2: consider b_{n-2}. We are given that $0 \le b_{n-2} \le 2$

  If b_{n-2} = 0, then n-2 must be placed before the two numbers from step n-1.

  If b_{n-2} =1, then n-2 must be placed between the two numbers from step n-1.

  If b_{n-2} =2, then n-2 must be placed after the two numbers from step n-1…….

……

- 1: We must place 1 after the b_{1st} number in the sequence constructed in step n-1.

Comments on Algorithm I

- The algorithm can determine the unique permutation of {1, 2, …, n} whose inversion sequence is $b_1, b_2, \ldots, b_n$.
- The disadvantage of this algorithm is that the location of each integer in the permutation are not known until the very end; only the relative positions of the integers remain fixed throughout the algorithm.
- The inversion sequence of the permutation 31524 is 1, 2, 0, 1, 0.
  - 1: 3 is larger than 1 and before 1 **1** 1 larger number is before "1"
  - 2: 3 and 5 are larger than 2 and before 2 **2** 2 larger number is before "2"
  - 3: no number is before 3 **0** 0 larger number is before "3"
  - 4: 5 is larger than 4 and before 4 **1**
  - 5: no number is larger than 5 **0**

Example

- Determine the permutation of {1, 2, 3, 4, 5, 6 ,7 ,8} whose inversion sequence is 5, 3, 4 ,0, 2, 1, 1, 0.
- b_1=5: put 1 in the 6th empty location
- b_2=3: put 2 in the 4th empty location
- b_3=4: put 3 in the 5th empty location
- b_4=0: put 4 in the 1st empty location
- b_5=2: put 5 in the 3rd empty location
- b_6=1: put 6 in the 2nd empty location
- b_7=1: put 7 in the 2nd empty location
- b_8=0: put 8 in the 1st empty location

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

Algorithm II: Inversion Method

- Begin with n empty locations which label 1, 2, …, n from left to right
- 1: put 1 in location number b_1 +1
- 2: put 2 in the (b_2 + 1)st empty location
- ……
- k: (general step) counting from the left we put k in the (b_k +1)st empty location.
- n: put n in the one remaining empty location.

Comments on Algorithm II

- The algorithm can determine the unique permutation of {1, 2, …, n} whose inversion sequence is $b_1, b_2, \ldots, b_n$.
- The advantage of this algorithm is that the location of each integer in the permutation can be determined.
  - Algorithm I: insert the integers one by one
  - Algorithm II: find the locations of the integers one by one.

Switches and inversions

- Bring the permutation 361245 to 123456 by successive switches of adjacent numbers.
- The inversion sequence is 220110. Hence there will be 6 times of switch.

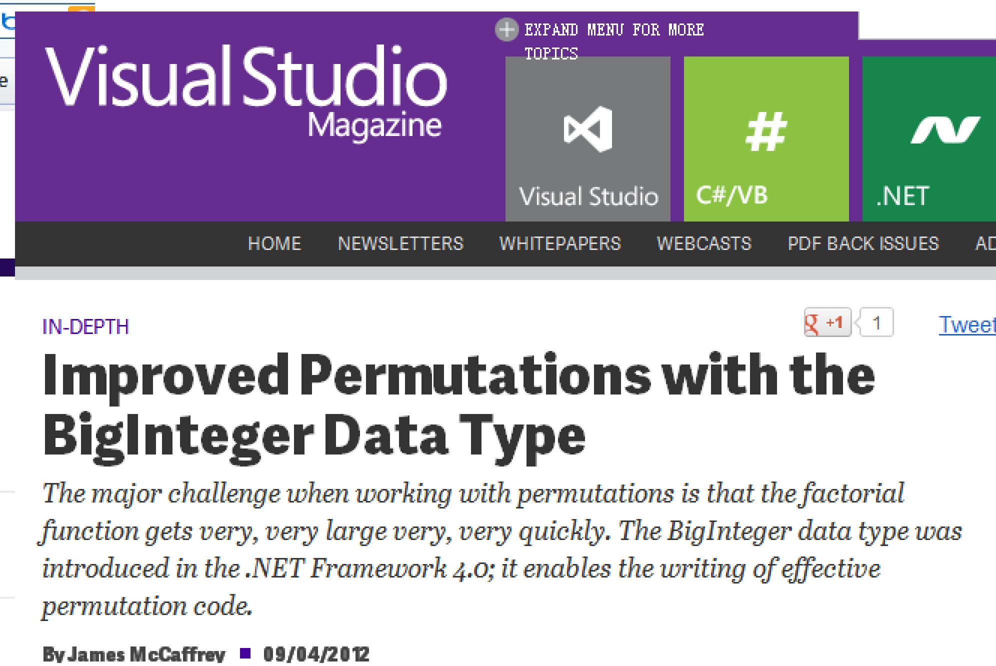
3	6	1	2	4	5
3	1	6	2	4	5
1	3	6	2	4	5
1	3	2	6	4	5
1	2	3	6	4	5
1	2	3	4	6	5
1	2	3	4	5	6

Industrial Support

—Java

```java
for(List<String> list : Permutation.of(Arrays.asList("a", "b", "c"))) {
    System.out.println(list);
}
```

```java
for(List<Integer> list : Combination.of(Arrays.asList(1, 2, 3, 4, 5), 3)) {
    System.out.println(list);
}
```



ELSEVIER

Finite Fields and Their Applications

Volume 17, Issue 1, January 2011, Pages 51–67

Ann. Comb. 14 (2010) 3–16
DOI 10.1007/s00026-010-0042-9
Published online February 16, 2010

Annals of Combinatorics

On constructing permutations of finite fields ☆

Amir Akbary[a], Dragos Ghioca[b], Qiang Wang[c]

a Dep

b Dep

c Sch

Permutations Generated by Stacks and Deques

Michael Albert[1], Mike Atkinson[1], and Steve Linton[2]

[1]Department of Computer Science, University of Otago, PO Box 56, Dunedin 9054, New Zealand

Menemui Matematik (Discovering Mathematics)
Vol. 32, No. 2: 51– 56 (2010)

A Two-level Algorithm for Generating Multiset Permutations

Tadao Takaoka
Department of Computer Science, University of Canterbury
Christchurch, New Zealand

y.ac.nz

New Recursive Circular Algorith

Sharmila Karim[1], Zurni On
Khairil Iskandar Othman[4], a

[123] *College of Art and S*

[1] *1*

rmutations in O(1) time for eac
ı memory requirement. There a

ISSN:2231-0711
www.ijcset.net

Hoang Chi Thanh et al

IJCSET |July 2012| Vol 2, Issue 7,1310-1315

From Permutations to Iterative Permutations

Hoang Chi Thanh [#1], Nguyen Thi Thuy Loan [*2], Nguyen Duy Ham [^3]

[#] *Department of Computer Science, VNU University of Science, Hanoi, Vietnam*
[1] thanhhc@vnu.vn
[*] *Department of Computer Science, College of Broadcasting II, HoChiMinh City, Vietnam*
[2] nguyenthithuyloan@vov.org.vn
[^] *Department of Computer Science, University of People's Security, HoChiMinh City, Vietnam*
[3] duyhaman@yahoo.com

Common Permutation generators

In C++ standard library, next_permutation, prev_permutation, could generate permutations in lexicographic order.

#include <algorithm>

bool next_permutation(iterator start, iterator end);

bool prev_permutation(iterator start, iterator end);

The next_permutation() function attempts to transform the given range of elements [start,end) into the next lexicographically greater permutation of elements. If it succeeds, it returns true, otherwise, it returns false.

http://www.slyar.com/blog/stl_next_permutation.html

Thinking Question

- Find the 2020th permutation with nine numbers of 1-9 in lexicographic order？
- We will post the GOOD HW as post session, if you do not want your HW to be posted, please inform me before Monday (Sep.28) noon.

- OJ Tasks
  - Lexicographic Order and SJT
  - Office Hour
    - Sep.28 after class TA：Kai Su

Thank you!