Generating Permutations and Combinations

Section 6.6

Generating Permutations

Suppose that a salesperson must visit six cities. In which order should these cities be visited to minimize total travel time?

Problem: List the permutations of any set of *n* elements.

How?

- ✓ **Any set with *n* elements can be placed in one-to-one correspondence with the set {1, 2, …, *n*}**
- ✓ **Generate the permutation of the *n* smallest positive integers, and then replace these integers with the corresponding elements.**

Introduction: lexicographic (or dictionary) ordering for permutation

What is lexicographic ordering for Permutations

The permutation $a_1a_2 \ldots a_n$ precedes the permutation of $b_1b_2 \ldots b_n$, if for some k, with $1 \leq k \leq n$, $a_1 = b_1$, $a_2 = b_2$, …, $a_{k-1} = b_{k-1}$, and $a_k < b_k$.

For example:

123465 precedes 124635

Algorithm for Generating Permutations

Algorithm of producing the $n!$ permutations of the integers 1, 2, …, n

- **Begin with the smallest permutation in lexicographic order, namely 1, 2, 3,4,…, n .**
- **Produce the next largest permutation.**
- **Continue until all $n!$ permutations have been found.**

Generating the next largest Permutations

Given permutation $a_1a_2\ldots a_n$, find the next largest permutation in increasing order:

(1) Find the integers

$$a_j, a_{j+1} \text{ with } a_j < a_{j+1} \text{ and } a_{j+1} > a_{j+2} > \ldots > a_n$$

(2) Put in the *j*th position the least integer among

$a_{j+1}, a_{j+2}, \ldots, a_n$ **that is greater than** a_j

(3) List in increasing order the rest of the integers

$$a_j, a_{j+1}, \ldots, a_n$$

Question: This algorithm produce the next largest Permutation in lexicographic order?

- What is the next largest permutation in lexicographic order after 124653?

The next largest permutation of 124653 in lexicographic order is 125346

- Generate the permutation of the integers 1, 2, 3 in lexicographic order.

123 → 132 → 213 → 231 → 312 → 321

Question: The algorithm can produce all Permutations in lexicographic order?

Generating Combinations

Problem 1:

Generate all combinations of the elements of a finite set .

How?

- ✓ **A combination is just a subset. ⇒ We need to list all subsets of the finite set.**
- ✓ **Use bit strings of length *n* to represent a subset of a set with *n* elements. ⇒ We need to list all bit strings of length *n*.**
- ✓ **The 2^n bit strings can be listed in order of their increasing size as integers in their binary expansions.**

Algorithm of Producing All Bit Strings of length n

- **Start with the bit string 000…00,with *n* zeros.**
- **Then, successively find the next largest expansion until the bit string 111…11 is obtained.**

The method to find the next largest binary expansion:

Locate the first position from the right that is not a 1, then changing all the 1s to the right of this position to 0s and making this first 0 a 1.

For example:

1000110011 → 1000110100

Problem 2:

Generate all *r*-combinations of the set {1, 2, …, *n*}

The algorithm for generating the *r*-combination of the set {1, 2, …, *n*}

(1) $S_1 = \{1, 2, \ldots, r\}$

(2) If $S_i = \{a_1, a_2, \ldots, a_r\}, 1 \le i \le C_n^r - 1$ **has found, then the next combination can be obtained using the following rules.**

First, locate the last element a_i **in the sequence such that** $a_i \ne n - r + i$ **. Then replace** a_i **with** $a_i + 1$ **and** a_j **with** $a_i + j - i + 1$, for $j = i+1, i+2, \ldots, r$.

- S_i ={2,3,5,6,9,10} is given. Find S_{i+1} .

$$S_{i+1} = \{2,3,5,7,8,9\}$$

- List all the 2-combination of {1,2,3,4,5}?

$\{1, 2\} \longrightarrow \{1, 3\} \longrightarrow \{1, 4\} \longrightarrow$

$\{1, 5\} \longrightarrow \{2, 3\} \longrightarrow \{2, 4\} \longrightarrow$

$\{2, 5\} \longrightarrow \{3, 4\} \longrightarrow \{3, 5\} \longrightarrow$

$\{4, 5\}$