Chapter 4

Generating Permutations and Combinations

In this chapter we explore some features of permutations and combinations that are not directly related to counting. We discuss some ordering schemes for them and algorithms for carrying out these schemes. In case of combinations, we use the subset terminology as discussed in Section 2.3. We also introduce the idea of a relation on a set and discuss two important instances, those of partial order and equivalence relation.

4.1 Generating Permutations

The set $\{1, 2, \ldots, n\}$ consisting of the first n positive integers has $n!$ permutations, which, even if n is only moderately large, is quite enormous. For instance, $15!$ is more than $1,000,000,000,000$. A useful and readily computable approximation to $n!$ is given by *Stirling's formula*,

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n,$$

where $\pi = 3.141\ldots$, and $e = 2.718\ldots$ is the base of the natural logarithm. As n grows without bound, the ratio of $n!$ to $\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ approaches 1. A proof of this can be found in many texts on advanced calculus and in an article by Feller.[1]

Permutations are of importance in many different circumstances, both theoretical and applied. For sorting techniques in computer science they correspond to the unsorted input data. We consider in this section a simple but elegant algorithm for generating all the permutations of $\{1, 2, \ldots, n\}$.

[1]W. Feller, A Direct Proof of Stirling's Formula, *Amer. Math. Monthly*, 74 (1967), 1223–1225.

Because of the large number of permutations of a set of n elements, for such an algorithm to be effective on a computer the individual steps must be simple to perform. The result of the algorithm should be a list containing each of the permutations of $\{1, 2, \ldots, n\}$ exactly once. The algorithm to be described has these features. It was independently discovered by Johnson[2] and Trotter[3] and was described by Gardner in a popular article.[4] The algorithm is based on the following observation:

> If the integer n is deleted from a permutation of $\{1, 2, \ldots, n\}$, the result is a permutation of $\{1, 2, \ldots, n-1\}$.

The same permutation of $\{1, 2, \ldots, n-1\}$ can result from different permutations of $\{1, 2, \ldots, n\}$. For instance, if $n = 5$ and we delete 5 from the permutation $3, 4, 1, 5, 2$, the result is $3, 4, 1, 2$. However $3, 4, 1, 2$ also results when 5 is deleted from $3, 5, 4, 1, 2$. Indeed there are exactly 5 permutations of $\{1, 2, 3, 4, 5\}$ which yield $3, 4, 1, 2$ upon the deletion of 5, namely,

$$\begin{array}{ccccc} \mathbf{5} & 3 & 4 & 1 & 2 \\ 3 & \mathbf{5} & 4 & 1 & 2 \\ 3 & 4 & \mathbf{5} & 1 & 2 \\ 3 & 4 & 1 & \mathbf{5} & 2 \\ 3 & 4 & 1 & 2 & \mathbf{5}, \end{array}$$

which we can also write as

$$\begin{array}{ccccc} 3 & 4 & 1 & 2 & \mathbf{5} \\ 3 & 4 & 1 & \mathbf{5} & 2 \\ 3 & 4 & \mathbf{5} & 1 & 2 \\ 3 & \mathbf{5} & 4 & 1 & 2 \\ \mathbf{5} & 3 & 4 & 1 & 2. \end{array}$$

More generally, each permutation of $\{1, 2, \ldots, n-1\}$ results from exactly n permutations of $\{1, 2, \ldots, n\}$ upon the deletion of n. Looked at from the opposite viewpoint, given a permutation of $\{1, 2, \ldots, n-1\}$, there are exactly n ways to insert n into this permutation to obtain a permutation of $\{1, 2, \ldots, n\}$. Thus, given a list of the $(n-1)!$ permutations of $\{1, 2, \ldots, n-1\}$, we can obtain a list of the $n!$ permutations of $\{1, 2, \ldots, n\}$ by systematically inserting n into each permutation of $\{1, 2, \ldots n-1\}$ in all possible ways. We now give an inductive description of such an algorithm; it generates the permutations of $\{1, 2, \ldots, n\}$ from the permutations of $\{1, 2, \ldots, n-1\}$. Thus, starting with the unique permutation 1 of $\{1\}$, we build up the permutations of $\{1, 2\}$, then the permutations of $\{1, 2, 3\}$, and so on until finally we obtain the permutations of $\{1, 2, \ldots, n\}$.

---

[2] S. M. Johnson, Generation of Permutations by Adjacent Transpositions, *Mathematics of Computation*, 17 (1963), 282–285.

[3] H. F. Trotter, Algorithm 115, *Communications of the Association for Computing Machinery*, 5 (1962), 434–435.

[4] M. Gardner, Mathematical Games, *Scientific American*, November (1974), 122–125.

$n = 2$: To generate the permutations of $\{1, 2\}$, write the unique permutation of $\{1\}$ twice and "interlace" the 2:

$$\begin{array}{ccc}
 & 1 & \mathbf{2} \\
\mathbf{2} & 1 &
\end{array}$$

The second permutation is obtained from the first by switching the two numbers.

$n = 3$: To generate the permutations of $\{1, 2, 3\}$, write each of the permutations of $\{1, 2\}$ three times in the order generated above, and interlace the 3 with them as shown:

$$\begin{array}{ccccc}
 & 1 & & 2 & \mathbf{3} \\
 & 1 & \mathbf{3} & 2 & \\
\mathbf{3} & 1 & & 2 & \\
\mathbf{3} & 2 & & 1 & \\
 & 2 & \mathbf{3} & 1 & \\
 & 2 & & 1 & \mathbf{3}
\end{array}$$

It is seen that each permutation other than the first is obtained from the preceding one by switching two adjacent numbers. When the 3 is fixed, as it is from the third to the fourth permutation in the sequence of generation, the switch comes from a corresponding switch for $n = 2$. We note that by switching 1 and 2 in the last permutation generated, we obtain the first one, namely, 123.

$n = 4$: To generate the permutations of $\{1, 2, 3, 4\}$, write each of the permutations of $1, 2, 3$ four times in the order generated above, and interlace the 4 with them.

Again we observe that each permutation is obtained from the preceding one by switching two adjacent numbers. When the 4 is fixed, as it is between the 4th and 5th, the 8th and 9th, the 12th and 13th, the 16th and 17th, and the 20th and 21st permutations in the sequence of generation, the switch comes from a corresponding switch for $n = 3$. Also, by switching 1 and 2 in the last permutation generated, we obtain the first permutation 1234.

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

It should now be clear how to proceed for any n. It readily follows by induction on n, using our earlier remarks, that the algorithm generates all permutations of $\{1, 2, \ldots, n\}$ exactly once. Moreover, each permutation other than the first is obtained from the preceding one by switching two adjacent numbers. The first permutation generated is $12 \cdots n$. This is so for $n = 1$ and follows by induction, since, in the algorithm, n is first put on the extreme right. Provided that $n \geq 2$, the last permutation generated is always $213 \cdots n$. This observation can be verified by induction on n as follows: If $n = 2$, the last permutation generated is 21. Now suppose that $n \geq 3$ and that $213 \cdots (n-1)$ is the last permutation generated for $\{1, 2, \ldots, n-1\}$. There are $(n-1)!$, an even number, of permutations of $\{1, 2, \ldots n-1\}$, and it follows that, in applying the algorithm, the integer n ends on the extreme right. Hence, $213 \cdots n$ is the last permutation generated. Since the last permutation is $213 \cdots n$, by switching 1 and 2 in the last permutation the first permutation results. Thus the algorithm is cyclical in nature.

To generate the permutations of $\{1, 2, \ldots, n\}$ in the manner just described, we must first generate the permutations of $\{1, 2, \ldots, n-1\}$. To generate the permutations of

$\{1, 2, \ldots, n-1\}$, we must first generate the permutations of $\{1, 2, \ldots, n-2\}$, and so on. We would like to be able to generate the permutations one at a time, using only the current permutation in order to generate the next one. We next show how it is possible to generate in this way the permutations of $\{1, 2, \ldots, n\}$ in the same order as above. Thus, rather than having to retain a list of all the permutations, we can simply overwrite the current permutation with the one that follows it. To do this, we need to determine which two adjacent integers are to be switched as the permutations appear on the list. The particular description we give is taken from Even.[5]

Given an integer k, we assign a *direction* to it by writing an arrow above it pointing to the left or to the right: $\overleftarrow{k}$ or $\overrightarrow{k}$. Consider a permutation of $\{1, 2, \ldots, n\}$ 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. For example, in

$$\overrightarrow{2}\ \overrightarrow{6}\ \overrightarrow{3}\ \overleftarrow{1}\ \overrightarrow{5}\ \overrightarrow{4}$$

only $3, 5$, and 6 are mobile. It follows that the integer 1 can never be mobile since there is no integer in $\{1, 2, \ldots, n\}$ smaller than 1. The integer n is mobile, except in two cases:

(1) n is the first integer and its arrow points to the left: $\overleftarrow{n} \cdots$,

(2) n is the last integer and its arrow points to the right: $\cdots \overrightarrow{n}$.

This is because n, being the largest integer in the set $\{1, 2, \ldots, n\}$, is mobile whenever its arrow points to an integer. We can now describe the algorithm for generating the permutations of $\{1, 2, \ldots, n\}$ directly.

Algorithm for generating the permutations of $\{1, 2, \ldots, n\}$

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

While there exists a mobile integer, do the following:

(1) Find the largest mobile integer m.

(2) Switch m and the adjacent integer to which its arrow points.

(3) Switch the direction of all the arrows above integers p with $p > m$.

We illustrate the algorithm for $n = 4$. The results are displayed in two columns, with the first column giving the first 12 permutations:

[5] S. Even, *Algorithmic Combinatorics*, Macmillan, New York (1973).

$$
\begin{array}{cccccccccc}
\overleftarrow{1} & \overleftarrow{2} & \overleftarrow{3} & \overleftarrow{4} & \qquad & \overrightarrow{4} & \overrightarrow{3} & \overleftarrow{2} & \overleftarrow{1} \\
\overleftarrow{1} & \overleftarrow{2} & \overleftarrow{4} & \overleftarrow{3} & & \overrightarrow{3} & \overrightarrow{4} & \overleftarrow{2} & \overleftarrow{1} \\
\overleftarrow{1} & \overleftarrow{4} & \overleftarrow{2} & \overleftarrow{3} & & \overrightarrow{3} & \overleftarrow{2} & \overrightarrow{4} & \overleftarrow{1} \\
\overleftarrow{4} & \overleftarrow{1} & \overleftarrow{2} & \overleftarrow{3} & & \overrightarrow{3} & \overleftarrow{2} & \overleftarrow{1} & \overrightarrow{4} \\
\overrightarrow{4} & \overleftarrow{1} & \overleftarrow{3} & \overleftarrow{2} & & \overleftarrow{2} & \overrightarrow{3} & \overleftarrow{1} & \overleftarrow{4} \\
\overleftarrow{1} & \overrightarrow{4} & \overleftarrow{3} & \overleftarrow{2} & & \overleftarrow{2} & \overrightarrow{3} & \overleftarrow{4} & \overleftarrow{1} \\
\overleftarrow{1} & \overleftarrow{3} & \overrightarrow{4} & \overleftarrow{2} & & \overleftarrow{2} & \overleftarrow{4} & \overrightarrow{3} & \overleftarrow{1} \\
\overleftarrow{1} & \overleftarrow{3} & \overleftarrow{2} & \overrightarrow{4} & & \overleftarrow{4} & \overleftarrow{2} & \overrightarrow{3} & \overleftarrow{1} \\
\overleftarrow{3} & \overleftarrow{1} & \overleftarrow{2} & \overleftarrow{4} & & \overrightarrow{4} & \overleftarrow{2} & \overleftarrow{1} & \overrightarrow{3} \\
\overleftarrow{3} & \overleftarrow{1} & \overleftarrow{4} & \overleftarrow{2} & & \overleftarrow{2} & \overrightarrow{4} & \overleftarrow{1} & \overrightarrow{3} \\
\overleftarrow{3} & \overleftarrow{4} & \overleftarrow{1} & \overleftarrow{2} & & \overleftarrow{2} & \overleftarrow{1} & \overrightarrow{4} & \overrightarrow{3} \\
\overleftarrow{4} & \overleftarrow{3} & \overleftarrow{1} & \overleftarrow{2} & & \overleftarrow{2} & \overleftarrow{1} & \overrightarrow{3} & \overrightarrow{4}
\end{array}
$$

Since no integer is mobile in $\overleftarrow{2}\ \overleftarrow{1}\ \overrightarrow{3}\ \overrightarrow{4}$, the algorithm stops.

That this algorithm generates the permutations of $\{1, 2, \ldots, n\}$, and in the same order as our previous method, follows by induction on n. We don't give a formal proof, and we only illustrate the inductive step from $n = 3$ to $n = 4$. We begin with $\overleftarrow{1}\ \overleftarrow{2}\ \overleftarrow{3}\ \overleftarrow{4}$, with 4 the largest mobile integer. The integer 4 remains mobile until it reaches the extreme left. At that point 4 has been inserted in all possible ways in the permutation 123 of $\{1, 2, 3\}$. Now 4 is no longer mobile. The largest mobile integer is 3, which is the same as the largest mobile integer in $\overleftarrow{1}\ \overleftarrow{2}\ \overleftarrow{3}$. Then 3 and 2 switch places and 4 changes direction. The switch is the same switch that would have occurred in $\overleftarrow{1}\ \overleftarrow{2}\ \overleftarrow{3}$. The result is now $\overrightarrow{4}\ \overleftarrow{1}\ \overleftarrow{3}\ \overleftarrow{2}$; now 4 is mobile again and remains mobile until it reaches the extreme right. Again a switch takes place, which is the same switch that would have occurred in $\overleftarrow{1}\ \overleftarrow{3}\ \overleftarrow{2}$. The algorithm continues like this, and 4 is interlaced in all possible ways with each permutation of $\{1, 2, 3\}$.

It is possible to determine, for a given permutation of $\{1, 2, \ldots, n\}$, at which step the permutation occurs in the preceding algorithm. Conversely, it is possible to determine which permutation occurs at a given step. For a clear analysis of this, we refer to the book by Even.[6]

Given a positive integer n, we have described an algorithm to generate *all* the $n!$ permutations of $\{1, 2, \ldots, n\}$. To conclude this section, we say a few brief words about generating a *random permutation* $i_1 i_2 \ldots i_n$ of $\{1, 2, \ldots, n\}$; that is, we want to generate one permutation of $\{1, 2, \ldots, n\}$ in such a way that each of the $n!$ permutations has an equal chance, namely $1/n!$, of being generated. Let $A = \{1, 2, \ldots, n\}$. One obvious

[6] Op. cit.

way to do this is to choose an integer at random from A (so each of the integers in A has a probability of $1/n$ of being chosen) and call this integer i_1. Then remove i_1 from A and choose an integer at random from the remaining $n-1$ elements (so now each integer left in A has a probability of $1/(n-1)$ of being chosen) and call this integer i_2. Continue this process of choosing an integer in A at random and removing it. When A becomes empty, we have a permutation $i_1 i_2 \dots i_n$ of $1, 2, \dots, n$ whose probability of being chosen is

$$\frac{1}{n} \cdot \frac{1}{n-1} \cdot \frac{1}{n-2} \cdots \frac{1}{2} \cdot \frac{1}{1} = \frac{1}{n!},$$

and hence a random permutation.[7] Another possible way, known as the *Knuth shuffle*, for generating a random permutation is as follows: Start with the identity permutation $12 \dots n$ and, sequentially, for each $k = 1, 2, \dots, n-1$, randomly choose one of the positions $k, k+1, \dots, n$ and switch the integers in position k and the chosen position.[8]

4.2 Inversions in Permutations

In this section we discuss a method of describing a permutation by means of its inversions discovered by Hall.[9] The notion of an inversion is an old one, and it plays an important role in the theory of determinants of matrices.

Let $i_1 i_2 \dots i_n$ be a permutation of the set $\{1, 2, \dots, n\}$. The pair (i_k, i_l) is called an *inversion* if $k < l$ and $i_k > i_l$. Thus, an inversion in a permutation corresponds to a pair of numbers that are out of their natural order. For example, the permutation 31524 has four inversions, namely $(3,1), (3,2), (5,2), (5,4)$. The only permutation of $\{1, 2, \dots, n\}$ with no inversions is $12 \dots n$. For a permutation $i_1 i_2 \dots i_n$, we let a_j denote the number of inversions whose second component is j. In other words,

> a_j equals the number of integers that precede j in the permutation but are greater than j; it measures how much j is out of order.

The sequence of numbers

$$a_1, a_2, \dots, a_n$$

is called the *inversion sequence* of the permutation $i_1 i_2 \dots i_n$. The number $a_1 + a_2 + \cdots + a_n$ measures the *disorder* of a permutation.

---

[7] Those with more knowledge of probability than given in this book will have recognized that we have cheated a little here by multiplying the individual probabilities. We can justify this as follows: In choosing the first k integers, there are $n(n-1)\cdots(n-k+1)$ possible outcomes with, each outcome having the same chance of being chosen, and so a 1 in $n(n-1)\cdots(n-k+1)$ chance, as any other. When $k = n$ we get $1/n!$.

[8] Note that we allow k as one of the possible positions and when k is chosen as the position, no switch actually occurs. If we didn't allow k, we could never end up with the identity permutation and hence we would not have a random generation scheme.

[9] M. Hall, Jr., *Proceedings Symposium in Pure Mathematics*, American Mathematical Society, Providence, 6 (1963), 203.

Example. The inversion sequence of the permutation 31524 is

$$1, 2, 0, 1, 0.$$

□

The inversion sequence $a_1, a_2, \ldots, a_n$ of the permutation $i_1 i_2 \ldots i_n$ satisfies the conditions

$$0 \leq a_1 \leq n-1,\ 0 \leq a_2 \leq n-2,\ \ldots,\ 0 \leq a_{n-1} \leq 1,\ a_n = 0.$$

This is so because for each $k = 1, 2, \ldots, n$, there are $n-k$ integers in the set $\{1, 2, \ldots, n\}$ which are greater than k. Using the multiplication principle, we see that the number of sequences of integers $b_1, b_2, \ldots, b_n$, with

$$0 \leq b_1 \leq n-1,\ 0 \leq b_2 \leq n-2,\ \ldots,\ 0 \leq b_{n-1} \leq 1,\ b_n = 0, \tag{4.1}$$

equals $n \times (n-1) \times \cdots \times 2 \times 1 = n!$.

Thus, there are as many permutations of $\{1, 2, \ldots, n\}$ as there are possible inversion sequences. This suggests (but does not yet prove!) that different permutations of $\{1, 2, \ldots, n\}$ have different inversion sequences. If we can show that each sequence of integers $b_1, b_2, \ldots, b_n$ satisfying (4.1) is the inversion sequence of a permutation of $\{1, 2, \ldots, n\}$, then it follows (from the pigeonhole principle) that different permutations have different inversion sequences.

Theorem 4.2.1 *Let $b_1, b_2, \ldots, b_n$ be a sequence of integers satisfying*

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

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

Proof. We describe two methods for uniquely constructing a permutation whose inversion sequence is $b_1, b_2, \ldots, b_n$.

Algorithm I

Construction of a permutation from its inversion sequence

n: Write down n.

$n-1$: Consider b_{n-1}. We are given that $0 \leq b_{n-1} \leq 1$. 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.

-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 $n-1$.

$\vdots$

$-k$: (*general step*) Consider b_{n-k}. We are given that $0 \le b_{n-k} \le k$. In steps n through $n-k+1$, the k numbers $n, n-1, \ldots, n-k+1$ have already been placed in the required order. If $b_{n-k} = 0$, then $n-k$ *must* be placed before all the numbers from step $n-k+1$. If $b_{n-k} = 1$, then $n-k$ *must* be placed between the first two numbers. . . . If $b_{n-k} = k$, then $n-k$ *must* be placed after all the numbers.

$\vdots$

1: We *must* place 1 after the b_1st number in the sequence constructed in step 2.

Steps $n, n-1, n-2, \ldots, 1$, when carried out, determine the unique permutation of $\{1, 2, \ldots, n\}$ 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 is not known until the very end; only the relative positions of the integers remain fixed throughout the algorithm.

In the second algorithm,[10] the positions of the integers $1, 2, \ldots, n$ in the permutation are determined.

Algorithm II

Construction of a permutation from its inversion sequence

We begin with n empty locations, which we label $1, 2, \ldots, n$ from left to right.

1 : Since there are to be b_1 integers that precede 1 in the permutation, we must put 1 in location number $b_1 + 1$.

2 : Since there are to be b_2 integers that precede 2 and are larger than 2 in the permutation, and since these integers have not yet been inserted, we must leave exactly b_2 empty locations for them. Thus, counting from the left, we put 2 in the $(b_2 + 1)$st empty location.

$\vdots$

[10]This algorithm was brought to my attention by J. Csima.

k : (*general step*) Since there are to be b_k integers that precede k in the permutation, and since these integers have not yet been inserted, we must leave exactly b_k empty locations for them. We observe that the number of empty locations at the beginning of this step is $n-(k-1)=n-k+1$. Counting from the left, we put k in the (b_k+1)st such empty location. Since $b_k \leq n-k$, we have $b_k+1 \leq n-k+1$ and so such an empty location can be determined.

$\vdots$

n : We put n in the one remaining empty location.

Carrying out the steps $1, 2, \ldots, n$ in the order described, we obtain the unique permutation of $\{1, 2, \ldots, n\}$ whose inversion sequence is $b_1, b_2, \ldots, b_n$. □

Example. Determine the permutation of $\{1, 2, 3, 4, 5, 6, 7, 8\}$ whose inversion sequence is $5, 3, 4, 0, 2, 1, 1, 0$.

The steps in the two algorithms in the proof of Theorem 4.2.1, when carried out for the given inversion sequence, yield the following results:

Algorithm I

8 :	8
7 :	87
6 :	867
5 :	8657
4 :	48657
3 :	486537
2 :	4862537
1 :	48625137

Thus, the permutation is 48625137.

Algorithm II

1 :						1		
2 :				2		1		
3 :				2		1	3	
4 :	4			2		1	3	
5 :	4			2	5	1	3	
6 :	4		6	2	5	1	3	
7 :	4		6	2	5	1	3	7
8 :	4	8	6	2	5	1	3	7
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

Again, the permutation is 48625137. □

It follows from Theorem 4.2.1 that the correspondence which associates the inversion sequence to each permutation is a one-to-one correspondence between the permutations of $\{1, 2, \ldots, n\}$ and the sequences of integers $b_1, b_2, \ldots, b_n$ satisfying

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

Thus, a permutation is uniquely specified by specifying its inversion sequence. Think of it as a code for the permutation. In the proof of Theorem 4.2.1, we have given two methods to break this code.

There is a subtle distinction worth making between a permutation and its inversion sequence. In choosing a permutation of $\{1, 2, \ldots, n\}$, we have to make n choices, one for each term of the permutation. We choose the first term, in any one of n ways, then the second term, in any one of $n-1$ ways, but notice that while the *number* of choices for the second term is always $n-1$, the actual possible *choices* for the second term depend on what was chosen for the first term (we cannot choose whatever has already been chosen). A similar situation occurs for the choice of the kth term. We have $n-(k-1)$ choices for the kth term, but the actual choices depend on what has already been chosen for the first $k-1$ terms.

The preceding description can be contrasted with choosing an inversion sequence $b_1, b_2, \ldots, b_n$ for a permutation of $\{1, 2, \ldots, n\}$. For b_1, we can choose any of the n integers $0, 1, \ldots, n-1$. For b_2, we can choose any of the $n-1$ integers $0, 1, \ldots, n-2$, and *it does not matter what our choice for* b_1 *is.* In general, for b_k, we can choose any of the $n-(k-1)$ integers $0, 1, \ldots, n-k$, and *it does not matter what our choices for* $b_1, b_2, \ldots, b_{k-1}$ *are.* Thus, the inversion sequence replaces dependent choices by independent choices.

It is customary to call a permutation $i_1 i_2 \ldots i_n$ of $\{1, 2, \ldots, n\}$ *even* or *odd* according to whether its number of inversions is even or odd. The *sign* of the permutation is then defined to be $+1$ or -1 according to whether it is even or odd. The sign of a permutation is important in the theory of determinants of matrices, where the determinant of an $n \times n$ matrix

$$A = [a_{ij}] \qquad (i, j = 1, 2, \ldots, n)$$

is defined to be

$$\det(A) = \sum \epsilon(i_1 i_2 \ldots i_n) a_{1i_1} a_{2i_2} \cdots a_{ni_n},$$

the summation extending over all permutations $i_1 i_2 \ldots i_n$ of the set $\{1, 2, \ldots, n\}$, and $\epsilon(i_1 i_2 \ldots i_n)$ is equal to the sign of $i_1 i_2 \ldots i_n$.[11]

If the permutation $i_1 i_2 \ldots i_n$ has inversion sequence $b_1, b_2, \ldots, b_n$ and $k = b_1 + b_2 + \cdots + b_n$ is the number of inversions, then $i_1 i_2 \ldots i_n$ can be brought to $12 \ldots n$ by k

[11]Thinking of an $n \times n$ matrix as an n-by-n chessboard in which the squares are occupied by numbers, the terms in the summation for the formula for the determinant correspond to the $n!$ ways to place n nonattacking rooks on the board.

successive switches of adjacent numbers. We first switch 1 successively with the b_1 numbers to its left. We then switch 2 successively with the b_2 numbers to its left which are greater than 2, and so on. In this way, we arrive at $12\ldots n$ after $b_1 + b_2 + \cdots + b_n$ switches.

Example. Bring the permutation 361245 to 123456 by successive switches of adjacent numbers.

The inversion sequence is 220110. The results of successive switches are as follows:

$$\begin{array}{cccccc}
3 & 6 & 1 & 2 & 4 & 5 \\
3 & 1 & 6 & 2 & 4 & 5 \\
1 & 3 & 6 & 2 & 4 & 5 \\
1 & 3 & 2 & 6 & 4 & 5 \\
1 & 2 & 3 & 6 & 4 & 5 \\
1 & 2 & 3 & 4 & 6 & 5 \\
1 & 2 & 3 & 4 & 5 & 6
\end{array}$$

□

This procedure is one instance of a sorting procedure common in computer science. The elements of a permutation $i_1 i_2 \ldots i_n$ correspond to the unsorted data. For more efficient sorting techniques and their analysis, consult Knuth.[12]

4.3 Generating Combinations

Let S be a set of n elements. For reasons that will be clear shortly, we take the set S in the form

$$S = \{x_{n-1}, \ldots, x_1, x_0\}.$$

We now seek an algorithm that generates all of the 2^n combinations of S, thus, all 2^n subsets of S. This means that we want a systematic procedure that lists all the subsets of S. The resulting list should contain all the subsets of S (and only subsets of S) with no duplications. Thus, according to Theorem 2.3.4, there will be 2^n subsets on the list.

Given a subset A of S, then each element x either belongs or does not belong to A. If we use 1 to denote that an element belongs and 0 to denote that an element does not belong, then we can identify the 2^n subsets of S with the 2^n n-tuples

$$(a_{n-1}, \ldots, a_1, a_0) = a_{n-1} \cdots a_1 a_0$$

[12]D. E. Knuth, *Sorting and Searching.* Volume 3 of *The Art of Computer Programming*, 2nd edition, Addison-Wesley, Reading, MA (1998).

of 0s and 1s.[13] We let the ith term a_i of the n-tuple correspond to the element x_i for each $i = 0, 1, \ldots, n-1$. For example, when $n = 3$, the $2^3 = 8$ subsets and their corresponding 3-tuples are given as follows:

	a_2	a_1	a_0
$\emptyset$	0	0	0
$\{x_0\}$	0	0	1
$\{x_1\}$	0	1	0
$\{x_1, x_0\}$	0	1	1
$\{x_2\}$	1	0	0
$\{x_2, x_0\}$	1	0	1
$\{x_2, x_1\}$	1	1	0
$\{x_2, x_1, x_0\}$	1	1	1

Example. Let $S = \{x_6, x_5, x_4, x_3, x_2, x_1, x_0\}$. The 7-tuple corresponding to the subset $\{x_5, x_4, x_2, x_0\}$ is 0110101. The subset corresponding to the 7-tuple 1010001 is $\{x_6, x_4, x_0\}$. □

Because of this identification of subsets of a set of n elements with n-tuples of 0s and 1s, to generate the subsets of a set of n elements, it suffices to describe a systematic procedure for writing in a list the 2^n n-tuples of 0s and 1s. Now, each such n-tuple can be regarded as a base 2 numeral.[14] For example, 10011 is the binary numeral for the integer 19 since

$$19 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0.$$

In general, given an integer m from 0 up to $2^n - 1$, it can be expressed in the form

$$m = a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \cdots + a_1 \times 2^1 + a_0 \times 2^0,$$

where each a_i is 0 or 1. Its binary numeral is

$$a_{n-1}a_{n-2} \cdots a_1 a_0.$$

Conversely, since

$$2^{n-1} + 2^{n-2} + \cdots + 2^1 + 2^0 = 2^n - 1,$$

every expression of the preceding form has value equal to an integer between 0 and $2^n - 1$. The n-tuples of 0s and 1s are thus in one-to-one correspondence with the integers $0, 1, \ldots, 2^n - 1$. Note that, in writing the binary numeral for an integer between 0 and $2^n - 1$, our convention is to use exactly n digits and thus to include, if necessary, some initial 0s that are not normally included.

[13] In the language of Section 3.3, we identify the subsets with the n-permutations of the multiset $\{n \cdot 0, n \cdot 1\}$.

[14] See also Section 1.7.

Example. Let $n = 7$. The number 29 is between 0 and $2^7 - 1 = 127$ and can be expressed as

$$29 = 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

Thus, 29 has a binary numeral of seven digits given by 0011101 and corresponds to the subset $\{x_4, x_3, x_2, x_0\}$ of the set

$$S = \{x_6, x_5, x_4, x_3, x_2, x_1, x_0\}.$$

□

How do we generate the 2^n subsets of $S = \{x_{n-1}, \ldots, x_1, x_0\}$? Equivalently, how do we generate the 2^n n-tuples of 0s and 1s? The answer is now simple. We write down the numbers from 0 to $2^n - 1$ in increasing order by size, *but in binary form, adding* 1 *each time, using base* 2 *arithmetic.* This is how the 3-tuples of 0s and 1s were generated earlier.

Example. Generate the 4-tuples of 0s and 1s.

Number	Binary Numeral
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

□

Example. If we use the base 2 arithmetic scheme just described, what is the subset of $\{x_6, x_5, x_4, x_3, x_2, x_1, x_0\}$ immediately following the subset $\{x_6, x_4, x_2, x_1, x_0\}$?

The subset $\{x_6, x_4, x_2, x_1, x_0\}$ corresponds to the binary numeral 1010111. Using base 2 arithmetic, we see that the next subset corresponds to

$$\begin{array}{rccccccc} & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ + & & & & & & & 1 \\ \hline & 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{array}$$

and thus is the subset $\{x_6, x_4, x_3\}$. Since

$$1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 87,$$

the subset $\{x_6, x_4, x_2, x_1, x_0\}$ is the 87th on the list. The subset that is 88th on the list is $\{x_6, x_4, x_3\}$. Note that the places on the list of all subsets are numbered beginning with 0 and ending with $2^n - 1$. The subset occupying the 0th place is always the empty set. When we say, for instance, the 5th subset on the list, we mean the subset on the list corresponding to the number 5, and not the subset corresponding to the number 4. Five subsets precede the 5th subset on the list. If this is not yet clear, the next example should clarify our convention. □

Example. Which subset of $S = \{x_6, x_5, x_4, x_3, x_2, x_1, x_0\}$ is 108th on the list?

We first find the base 2 numeral for 108:

$$108 = 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0.$$

Hence, the base 2 numeral for 108 is

$$1101100.$$

Thus, the subset is $\{x_6, x_5, x_3, x_2\}$. Which subset immediately precedes this one? We simply subtract in base 2:

$$\begin{array}{rccccccc} & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ - & & & & & & & 1 \\ \hline & 1 & 1 & 0 & 1 & 0 & 1 & 1. \end{array}$$

This corresponds to the subset $\{x_6, x_5, x_3, x_1, x_0\}$. □

We now describe in compact form our algorithm for generating the subsets of a set of n elements. The description is in terms of n-tuples of 0s and 1s. The *rule of succession* given in the algorithm is a consequence of addition using base 2 arithmetic.

Base 2 Algorithm for Generating the Subsets of

$$\{x_{n-1}, \ldots, x_1, x_0\}$$

Begin with $a_{n-1} \cdots a_1 a_0 = 0 \cdots 00$.

While $a_{n-1} \cdots a_1 a_0 \neq 1 \cdots 11$, do the following:

(1) Find the smallest integer j (between $n-1$ and 0) such that $a_j = 0$.

(2) Replace a_j with 1 and replace each of $a_{j-1}, \ldots, a_0$ (which, by our choice of j, all equal 1) with 0.

The algorithm comes to an end when $a_{n-1} \cdots a_1 a_0 = 1 \cdots 11$, which is the last binary n-tuple on the resulting list.

The ordering of the n-tuples of 0s and 1s produced by the base 2 generation scheme is called the *lexicographic ordering of n-tuples*. In this ordering, an n-tuple $a_{n-1} \cdots a_1 a_0$ occurs earlier on the list than another n-tuple $b_{n-1} \cdots b_1 b_0$ provided that, starting at the left, the first position in which they disagree, say position j, we have $a_j = 0$ and $b_j = 1$. (Why? Because this is equivalent to saying that the number whose base 2 numeral is given by $a_{n-1} \cdots a_1 a_0$ is smaller than the number whose base 2 numeral is given by $b_{n-1} \cdots b_1 b_0$.) Thinking of the n-tuples as "words" of length n in an alphabet of two "letters," 0 and 1, in which 0 is the first letter of the alphabet and 1 is the second letter, the lexicographic ordering is the order in which these words would occur in a dictionary.

Viewing the n-tuples as subsets of the set $\{x_{n-1}, \ldots, x_1, x_0\}$, we see that for each j with $n-1 > j$, all the subsets of $\{x_j, \ldots, x_1, x_0\}$ precede those subsets which contain at least one of the elements $x_{n-1}, \ldots, x_{j+1}$. For this reason, the lexicographic ordering on n-tuples of 0s and 1s, when viewed as an ordering of the subsets of $\{x_{n-1}, \ldots, x_1, x_0\}$, is sometimes called the *squashed ordering of subsets*. In the squashed ordering we list all the subsets of the current elements before introducing a new element. The squashed ordering of the subsets of $\{x_3 = 4, x_2 = 3, x_1 = 2, x_0 = 1\}$ is given below and corresponds to our earlier (lexicographic) listing of the binary 4-tuples. Notice how, in this ordering, all the subsets that do not contain 4 come before those that do. Of the subsets that do not contain 4, those that do not contain 3 come before those that do. Of the subsets that contain neither 4 nor 3, those that do not contain 2 come before those that do.

$$\begin{array}{r}
\emptyset \\
1 \\
2 \\
1,2 \\
3 \\
1,3 \\
2,3 \\
1,2,3 \\
4 \\
1,4 \\
2,4 \\
1,2,4 \\
3,4 \\
1,3,4 \\
2,3,4 \\
1,2,3,4.
\end{array}$$

Subsets of $\{1, 2, 3, 4\}$ *in the squashed ordering.*

Notice how, in this ordering, all the subsets that do not contain 4 come before those that do. Of the subsets that do not contain 4, those that do not contain 3 come before those that do. Of the subsets that contain neither 4 nor 3, those that do not contain 2 come before those that do.

The immediate successor of a subset in the squashed ordering of subsets (equivalently, the immediate successor of an n-tuple in the lexicographic ordering of n-tuples) may differ greatly from the subset itself. The subset $A = \{x_6, x_4, x_3\}$ (equivalently, the 7-tuple 1011000) which follows the subset $B = \{x_6, x_4, x_2, x_1, x_0\}$ (equivalently, the 7-tuple 1010111) differs from B in four instances, since A contains x_3 (and B doesn't) while B contains x_2, x_1, and x_0 (and A doesn't). This suggests consideration of the following question: *Is it possible to generate the subsets of a set of* n *elements in a different order so that the immediate successor of a subset differs from it as little as possible?* Here *as little as possible means* that the immediate successor of a subset is obtained by either including a new element or deleting an old element, but not both; in short, one in or one out. Such a generation scheme can be important for many reasons, not the least of which is that there would be a smaller chance of error in generating all the subsets.

Example. Let $S = \{x_{n-1}, \ldots, x_1, x_0\}$, and consider the following lists of the subsets of S and the corresponding n-tuples for $n = 1, 2, 3$.

$n = 1$

$\emptyset$	0
$\{x_0\}$	1

$n = 2$

$\emptyset$	0 0
$\{x_0\}$	0 1
$\{x_1, x_0\}$	1 1
$\{x_1\}$	1 0

$n = 3$

$\emptyset$	0 0 0
$\{x_0\}$	0 0 1
$\{x_1, x_0\}$	0 1 1
$\{x_1\}$	0 1 0
$\{x_2, x_1\}$	1 1 0
$\{x_2, x_1, x_0\}$	1 1 1
$\{x_2, x_0\}$	1 0 1
$\{x_2\}$	1 0 0

In each list, the transition from one subset to the next is obtained by inserting a new element or removing an element already present, but not both. In terms of n-tuples of 0s and 1s, we change a 0 to a 1 or a 1 to a 0, but not both. □

We now make a further identification, this time a geometric one. We regard an n-tuple of 0s and 1s as the coordinates of a point in n-dimensional space. Thus, for

$n = 1$, the identification is with points on a line; for $n = 2$, it is with points in 2-space or a plane; for $n = 3$, it is with points in three-dimensional space.

Figure 4.1

Example. Let $n = 1$. The 1-tuples of 0s and 1s correspond to the endpoints or corners of a unit line segment, as shown in Figure 4.1. □

Example. Let $n = 2$. The 2-tuples of 0s and 1s correspond to the corners of a unit square, as shown in Figure 4.2. □

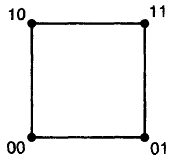

Figure 4.2

Example. Let $n = 3$. The 3-tuples of 0s and 1s correspond to the corners of a unit cube, as shown in Figure 4.3. □

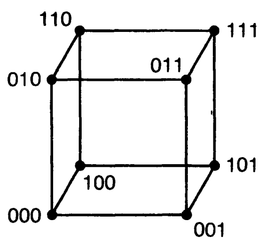

Figure 4.3

Notice that in all three examples there is an edge between two corners precisely when their coordinates differ in only one place. This is precisely the feature we are looking for in generating the n-tuples of 0s and 1s.

We can generalize to any n. The *unit n-cube* (a 1-cube is a line segment, a 2-cube is a square, a 3-cube is an ordinary cube) has 2^n corners whose coordinates are the 2^n n-tuples of 0s and 1s. There is an edge of the n-cube joining two corners precisely when the coordinates of the corners differ in only one place. An algorithm for generating the n-tuples of 0s and 1s which has the property that the successor of an n-tuple differs from it in only one place corresponds to a walk along the edges of an n-cube that visits every corner exactly once. Any such walk (or the resulting list of n-tuples) is called a *Gray code of order n*.[15] If it is possible to traverse one more edge to get from the terminal corner to the initial corner of the walk, then the Gray code is called *cyclic*. The lists for $n = 1, 2$, and 3 in the examples are cyclic Gray codes. They have an additional property that makes them quite special, and we now investigate it.

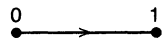

Figure 4.4

Let us begin with the unit 1-cube and the Gray code, which starts at 0 and ends at 1, as shown in Figure 4.4. We build a unit 2-cube by taking two copies of the 1-cube and joining corresponding corners. We attach a 0 to the coordinates of one copy and a 1 to the coordinates of the other: We obtain a cyclic Gray code for the 2-cube by first following the Gray code on one copy of the 1-cube, crossing over to the other copy, and then following the Gray code for the 1-cube *in the reverse direction*, as shown on the left in Figure 4.5.

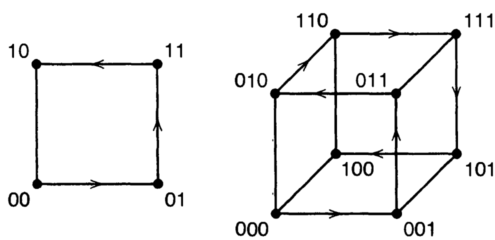

Figure 4.5

We build a unit 3-cube in a similar way from the unit 2-cube. We take two copies of the 2-cube and join corresponding corners. We attach a 0 to the coordinates of one copy and a 1 to the coordinates of the other. We obtain a cyclic Gray code for the 3-cube by first following the Gray code on one copy of the 2-cube, crossing over to the other copy, and then following the Gray code for the 2-cube in the reverse direction, as shown on the right in Figure 4.5.

We may continue in this manner to construct inductively a Gray code of order n for any integer $n \geq 1$. The Gray code constructed in this way is called the *reflected Gray code*. The n-cube is a convenient visual device and, as we shall see, need not be introduced in order to obtain the reflected Gray code of order n. The reflected Gray

[15]In 1878, the French engineer Émile Baudot demonstrated the use of a Gray code in a telegraph. It was the Bell Labs researcher Frank Gray who first patented these codes in 1953.

code for $n = 4$ is as follows:

$$\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0
\end{array}$$

The general inductive definition of the reflected Gray code of order n is the following:

(1) The reflected Gray code of order 1 is $\begin{array}{c} 0 \\ 1 \end{array}$.

(2) Suppose $n > 1$ and the reflected Gray code of order $n-1$ has been constructed. To construct the reflected Gray code of order n, we first list the $(n-1)$-tuples of 0s and 1s in the order given by the reflected Gray code of order $n-1$, and attach a 0 at the beginning (i.e. on the left) of each $(n-1)$-tuple. We then list the $(n-1)$-tuples in the order which is the reverse of that given by the reflected Gray code of order $n-1$, and attach a 1 at the beginning.

It follows from this inductive definition that the reflected Gray code of order n begins with the n-tuple $00\cdots0$ and ends with the n-tuple $10\cdots0$. It is therefore cyclic, since $00\cdots0$ and $10\cdots0$ differ in only one place.

Since the reflected Gray codes have been defined inductively, to construct the reflected Gray code of order n, we first construct the reflected Gray code of order $n-1$. So, for instance, to construct the reflected Gray code of order 6, we first construct the reflected Gray code of order 5. To do that we must first construct the reflected Gray code of order 4, and so on. Therefore, to construct the reflected Gray code of order 6, using the inductive definition, we must construct sequentially the reflected Gray codes of orders 1, 2, 3, 4, and 5. We now describe an algorithm that enables us to construct the reflected Gray code of order n directly. To do this we need a *rule of succession*, which tells us which place to change (from a 0 to a 1 or a 1 to a 0) in going from one n-tuple to the next in the reflected Gray code. This rule of succession is provided in the next algorithm.

If $a_{n-1}a_{n-2}\cdots a_0$ is an n-tuple of 0s and 1s, then

$$\sigma(a_{n-1}a_{n-2}\cdots a_0) = a_{n-1} + a_{n-2} + \cdots + a_0$$

is the number of its 1s (and thus equals the size of the subset to which it corresponds).

Algorithm for generating the n-tuples of 0s and 1s in the reflected Gray code order

Begin with the n-tuple $a_{n-1}a_{n-2}\cdots a_0 = 00\cdots 0$.

While the n-tuple $a_{n-1}a_{n-2}\cdots a_0 \neq 10\cdots 0$, do the following:

(1) Compute $\sigma(a_{n-1}a_{n-2}\cdots a_0) = a_{n-1} + a_{n-2} + \cdots + a_0$.

(2) If $\sigma(a_{n-1}a_{n-2}\cdots a_0)$ is even, change a_0 (from 0 to 1 or 1 to 0).

(3) Else, determine j such that $a_j = 1$ and $a_i = 0$ for all i with $j > i$ (i.e., the first 1 from the right), and then change a_{j+1} (from 0 to 1 or 1 to 0).

We note that if, in step (3), we have $a_{n-1}a_{n-2}\cdots a_0 \neq 10\cdots 0$, then $j \leq n-2$, so that $j+1 \leq n-1$ and a_{j+1} is defined. We also note that in step (3) we may have $j = 0$, that is, $a_0 = 1$; in this case there is no i with $i < j$, and we change a_1 as instructed in step (3).

You may wish to check that this algorithm does give the Gray code of order 4 as already presented.

Theorem 4.3.1 *The preceding algorithm for generating the n-tuples of* 0*s and* 1*s produces the reflected Gray code of order n for each positive integer n.*

Proof. We prove the theorem by induction on n. It is clear that the algorithm applied to $n = 1$ produces the reflected Gray code of order 1. Let $n > 1$, and assume that the algorithm applied to $n-1$ produces the reflected Gray of order $n-1$. The first 2^{n-1} n-tuples of the reflected Gray code of order n consist of the $(n-1)$-tuples of the reflected Gray code of order $n-1$ with a 0 attached at the beginning of each $(n-1)$-tuple. Since the $(n-1)$-tuple $10\cdots 0$ occurs last in the reflected Gray code of order $n-1$, it follows that the rule of succession applied to the first $(2^{n-1}-1)$ n-tuples of the reflected Gray code of order n has the same effect as applying the rule of succession to all but the last $(n-1)$-tuple of the reflected Gray code of order $n-1$ and then attaching a 0. Hence it is a consequence of the inductive hypothesis that the rule of succession produces the first half of the reflected Gray code of order n. The

2^{n-1}st n-tuple of the reflected Gray code of order n is $010\cdots0$. Since $\sigma(010\cdots0)=1$, an odd number, the rule of succession applied to $010\cdots0$ gives $110\cdots0$, which is the $(2^{n-1}+1)$st n-tuple of the reflected Gray code of order n.

Consider now two consecutive n-tuples in the second half of the reflected Gray code of order n:

$$\begin{array}{l} 1\ a_{n-2}\cdots a_0 \\ 1\ b_{n-2}\cdots b_0. \end{array}$$

Then $a_{n-2}\cdots a_0$ immediately follows $b_{n-2}\cdots b_0$ in the reflected Gray code of order $n-1$:

$$\begin{array}{l} b_{n-2}\cdots b_0 \\ a_{n-2}\cdots a_0. \end{array}$$

Now $\sigma(a_{n-2}\cdots a_0)$ and $\sigma(b_{n-2}\cdots b_0)$ are of opposite parity. One is even and the other is odd. Also, $\sigma(1a_{n-2}\cdots a_0)$ and $\sigma(a_{n-2}\cdots a_0)$ are of opposite parity, and so are $\sigma(1b_{n-2}\cdots b_0)$ and $\sigma(b_{n-2}\cdots b_0)$. Suppose that $\sigma(b_{n-2}\cdots b_0)$ is even. Then $\sigma(a_{n-2}\cdots a_0)$ is odd and $\sigma(1a_{n-2}\cdots a_0)$ is even. Using the induction assumption, we see that $a_{n-2}\cdots a_0$ is obtained from $b_{n-2}\cdots b_0$ by changing b_0. The rule of succession applied to $1a_{n-2}\cdots a_0$ instructs us to change a_0, and this gives $1b_{n-2}\cdots b_0$ as desired. Now suppose that $\sigma(b_{n-2}\cdots b_0)$ is odd. Then $\sigma(a_{n-2}\cdots a_0)$ is even and $\sigma(1a_{n-2}\cdots a_0)$ is odd. The rule of succession applied to $1a_{n-2}\cdots a_0$ has the opposite effect from the rule of succession applied to $b_{n-2}\cdots b_0$. Hence, it also follows by the induction assumption that the rule of succession applied to $1a_{n-2}\cdots a_0$ gives $1b_{n-2}\cdots b_0$, as desired. Therefore, the theorem holds by induction. □

Example. Determine the 8-tuples that are successors of 10100110, 00011111, and 01010100 in the reflected Gray code of order 8.

Because $\sigma(10100110)=4$ is an even number, 10100111 follows 10100110. Because $\sigma(00011111)=5$is an odd number, then in step (3) of the algorithm $j=0$ so that 00011101 follows 00011111. Since $\sigma(01010100)=3$, 01011100 follows 01010100. □

We have described two linear orderings of the 2^n binary n-tuples: the lexicographic order obtained, starting with $00\cdots0$, by using base 2 arithmetic; and the reflected Gray code order, which also starts with $00\cdots0$. The lexicographic order corresponds to the integers from 0 to 2^n-1 in base 2, and we can think of the reflected Gray code order as listing the binary n-tuples in a specified order from 0 to 2^n-1. Let $a_{n-1}\cdots a_1a_0$ be a binary n-tuple. We can say explicitly in what place this binary n-tuple occurs on the list in Gray code order. For $i=0,1,\ldots,n-1$, let

$$b_i=\begin{cases} 0 & \text{if } a_{n-1}+\cdots+a_i \text{ is even, and} \\ 1 & \text{if } a_{n-1}+\cdots+a_i \text{ is odd.} \end{cases}$$

Then $a_{n-1} \cdots a_1 a_0$ is in the same place on the Gray code order list as $b_{n-1} \cdots b_1 b_0$ is on the lexicographic order list. Put another way, $a_{n-1} \cdots a_1 a_0$ is in place

$$k = b_{n-1} \times 2^{n-1} + \cdots + b_1 \times 2 + b_0 \times 2^0$$

on the Gray code order list. We leave this verification as an exercise.

4.4 Generating r-Subsets

In Section 4.3, we described two orderings for the subsets of a set of n elements and corresponding algorithms based on a rule of succession for generating the subsets. We now consider only the subsets of a fixed size r and seek a method to generate these subsets. One way to do this is to generate *all* subsets and then go through the list and select those that contain exactly r elements. This is obviously a very inefficient approach.

Example. In Section 4.3, we listed all the 4-subsets of $\{1,2,3,4\}$ in the squashed ordering. Selecting the 2-subsets from among them, we get the squashed ordering of the 2-subsets of $\{1,2,3,4\}$:

$$\begin{array}{c} 1,2 \\ 1,3 \\ 2,3 \\ 1,4 \\ 2,4 \\ 3,4. \end{array}$$

□

In this section, we develop an algorithm for a lexicographic ordering of the r-subsets of a set of n elements, where r is a fixed integer with $1 \leq r \leq n$. We now take our set to be the set

$$S = \{1, 2, \ldots, n\}$$

consisting of the first n positive integers. This gives us a natural order,

$$1 < 2 < \cdots < n,$$

on the elements of S. Let A and B be two r-subsets of the set $\{1, 2, \ldots, n\}$. Then we say that *A precedes B in the lexicographic order* provided that the smallest integer which is in their union $A \cup B$, but not in their intersection $A \cap B$ (that is, in one but not both of the sets), is in A.

Example. Let 5-subsets A and B of $\{1,2,3,4,5,6,7,8\}$ be given by

$$A = \{2,3,4,7,8\}, \qquad B = \{2,3,5,6,7\}.$$

The smallest element that is in one, but not both, of the sets is 4 (4 is in A). Hence A precedes B in the lexicographic order. □

How is this a lexicographic order in the sense used in the preceding section and in the sense used in a dictionary? We think of the elements of S as the letters of an alphabet, where 1 is the first letter of the alphabet, 2 is the second letter, and so on. We want to think of the r-subsets as "words" of length r over the alphabet S and then impose a dictionary-type order on the words. But the letters in a word form an ordered sequence (e.g., *part* is not the same word as *trap*), and for subsets, as we have learned, order doesn't matter. Since order doesn't matter in a subset, let us agree that, whenever we write a subset of $\{1, 2, \ldots, n\}$, we write the integers in it from smallest to largest. Thus, we agree that an r-subset of $S = \{1, 2, \ldots, n\}$ is to be written in the form

$$a_1, a_2, \ldots, a_r, \text{ where } 1 \leq a_1 < a_2 < \cdots < a_r \leq n.$$

Let us also agree, for convenience, to write this r-subset as

$$a_1 a_2 \cdots a_r$$

without commas; that is, as a word of length r. We now have established a convention for writing subsets that allows us to regard a subset as a word. But note that not all words are allowed. The only words that will be in our dictionary are those that have r letters from our alphabet $1, 2, \ldots, n$ *and* for which the letters are in strictly increasing order (in particular, there are no repeated letters in our words).

Example. We return to our previous example and now, with our established conventions, write $A = 23478$ and $B = 23567$. We see that A and B agree in their first two letters and disagree in their third letter. Since $4 < 5$ (4 comes earlier in our alphabet than 5), A precedes B in the lexicographic order. □

Example. We consider the lexicographic order of the 5-subsets of $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ The first 5-subset is 12345; the last 5-subset is 56789. What 5-subset immediately follows 12389 (in our dictionary)? Among the 5-subsets that begin with 123, 12389 is the last. Among the 5-subsets that begin with 12 and don't have a 3 in the third position, 12456 is the first. Thus, 12456 immediately follows 12389. □

We generalize this example and determine, for all but the last word in our dictionary, the word that immediately follows it.

Theorem 4.4.1 *Let $a_1 a_2 \cdots a_r$ be an r-subset of $\{1, 2, \ldots, n\}$. The first r-subset in the lexicographic ordering is $12 \cdots r$. The last r-subset in the lexicographic ordering is $(n-r+1)(n-r+2) \cdots n$. Assume that $a_1 a_2 \cdots a_r \neq (n-r+1)(n-r+2) \cdots n$. Let k be the largest integer such that $a_k < n$ and $a_k + 1$ is different from each of $a_{k+1}, \ldots, a_r$. Then the r-subset that is the immediate successor of $a_1 a_2 \cdots a_r$ in the lexicographic ordering is*

$$a_1 \cdots a_{k-1}(a_k + 1)(a_k + 2) \cdots (a_k + r - k + 1).$$

Proof. It follows from the definition of the lexicographic order that $12\cdots r$ is the first and $(n-r+1)(n-r+2)\cdots n$ is the last r-subset in the lexicographic ordering. Now let $a_1a_2\cdots a_r$ be any r-subset other than the last, and determine k as indicated in the theorem. Then

$$a_1a_2\cdots a_r = a_1\cdots a_{k-1}a_k(n-r+k+1)(n-r+k+2)\cdots(n),$$

where

$$a_k+1 < n-r+k+1.$$

Thus $a_1a_2\cdots a_r$ is the last r-subset that begins with $a_1\cdots a_{k-1}a_k$. The r-subset

$$a_1\cdots a_{k-1}(a_k+1)(a_k+2)\cdots(a_k+r-k+1)$$

is the first r-subset that begins $a_1\cdots a_{k-1}a_k+1$ and hence is the immediate successor of $a_1a_2\cdots a_r$. □

From Theorem 4.4.1, we conclude that the next algorithm generates the r-subsets of $\{1,2,\ldots,n\}$ in lexicographic order.

Algorithm for generating the r-subsets of $\{1,2,\ldots,n\}$ in lexicographic order

Begin with the r-subset $a_1a_2\cdots a_r = 12\cdots r$.
While $a_1a_2\cdots a_r \neq (n-r+1)(n-r+2)\cdots n$, do the following:

(1) Determine the largest integer k such that $a_k+1 \leq n$ and a_k+1 is not one of $a_1,a_2,\ldots,a_r$.

(2) Replace $a_1a_2\cdots a_r$ with the r-subset

$$a_1\cdots a_{k-1}(a_k+1)(a_k+2)\cdots(a_k+r-k+1).$$

Example. We apply the algorithm to generate the 4-subsets of $S=\{1,2,3,4,5,6\}$ and obtain the following (using three columns):

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

□

Combining the algorithm for generating permutations of a set with that for generating r-subsets of an n-element set, we obtain an algorithm for generating r-permutations of an n-element set.

Example. Generate the 3-permutations of $\{1, 2, 3, 4\}$. We first generate the 3-subsets in lexicographic order: 123, 124, 134, 234. For each 3-subset, we then generate all of its permutations:

123	124	134	234
132	142	143	243
312	412	413	423
321	421	431	432
231	241	341	342
312	214	314	324.

□

We conclude by determining the position of each r-subset in the lexicographic order of the r-subsets of $\{1, 2, \ldots, n\}$.

Theorem 4.4.2 *The r-subset $a_1 a_2 \cdots a_r$ of $\{1, 2, \ldots, n\}$ occurs in place number*

$$\binom{n}{r} - \binom{n-a_1}{r} - \binom{n-a_2}{r-1} - \cdots - \binom{n-a_{r-1}}{2} - \binom{n-a_r}{1}$$

in the lexicographic order of the r-subsets of $\{1, 2, \ldots, n\}$.

Proof. We first count the number of r-subsets that come *after* $a_1 a_2 \cdots a_r$:

(1) There are $\binom{n-a_1}{r}$ r-subsets whose first element is greater than a_1 that come after $a_1 a_2 \cdots a_r$.

(2) There are $\binom{n-a_2}{r-1}$ r-subsets whose first element is a_1 but whose second element is greater than a_2 that come after $a_1 a_2 \cdots a_r$.

$\vdots$

$-$ 1) There are $\binom{n-a_{r-1}}{2}$ r-subsets that begin $a_1 \cdots a_{r-2}$ but whose $(r-1)$st element is greater than a_{r-1} that come after $a_1 a_2 \cdots a_r$.

(r) There are $\binom{n-a_r}{1}$ r-subsets that begin $a_1 \cdots a_{r-1}$ but whose rth element is greater than a_r that come after $a_1 a_2 \cdots a_r$.

Subtracting the number of r-subsets that come after $a_1 a_2 \cdots a_r$ from the total number $\binom{n}{r}$ of r-subsets, we find that the place of $a_1 a_2 \cdots a_r$ is as given in the theorem. □

Example. In which place is the subset 1258 among the 4-subsets of $\{1, 2, 3, 4, 5, 6, 7, 8\}$ in lexicographic order?

We apply Theorem 4.4.2 and find that 1258 is in place

$$\binom{8}{4} - \binom{7}{4} - \binom{6}{3} - \binom{3}{2} - \binom{0}{1} = 12.$$

□

4.5 Partial Orders and Equivalence Relations

In this chapter we have defined various "natural" orders on the sets of permutations, subsets, and r-rubsets of a finite set, namely, the orders determined by the generating schemes. These orders are "total orders" in the sense that there is a first object, a second object, a third object, ... , a last object. There is a more general notion of order, called partial order, which is extremely important and useful in mathematics. Perhaps the two partial orders which are not total orders that are most familiar are those defined by containment of one set in another and divisibility of one integer by another. These are *partial* orders in the sense that, given any two sets, neither need be a subset of the other, and given any two integers, neither need be divisible by the other.

To give a precise definition of a partial order, it is important to know what is meant in mathematics by a *relation*. Let X be a set. A relation on X is a subset R of the set $X \times X$ of ordered pairs of elements of X. We write $a\,R\,b$ (a is related to b), provided that the ordered pair (a, b) belongs to R; we also write $a \not\!R\, b$ whenever (a, b) is not in R (a is not related to b).

Example. Let $X = \{1,2,3,4,5,6\}$. Write $a \mid b$ to mean that a is a divisor of b (equivalently, b is divisible by a). This defines a partial order on X and we have, for example, $2 \mid 6$ and $3 \not| 5$.

Now consider the collection $\mathcal{P}(X)$ of all subsets of X. For A and B in $\mathcal{P}(X)$, we write as usual $A \subseteq B$, read A is *contained in* B, provided that every element of A is also an element of B. This defines a relation on $\mathcal{P}(X)$ and we have, in particular, that, $\{1\} \subseteq \{1,3\}$ and $\{1,2\} \not\subseteq \{2,3\}$. □

The following are special properties that a relation R on a set X may have:

1. R is *reflexive*, provided that $x\,R\,x$ for all x in X.

2. R is *irreflexive*, provided that $x \not\!R\, x$ for all x in X.

3. R is *symmetric*, provided that, for all x and y in X, whenever we have $x\,R\,y$ we also have $y\,R\,x$.

4. R is *antisymmetric*, provided that, for all x and y in X with $x \neq y$, whenever we have $x\,R\,y$, we also have $y \not\!R\, x$. Equivalently, for all x and y in X, $x\,R\,y$ and $y\,R\,x$ together imply that $x = y$.

5. R is *transitive*, provided that, for all x, y, z in X, whenever we have $x\,R\,y$ and $y\,R\,z$, we also have $x\,R\,z$.

Example. The relations of subset, $\subseteq$, and divisibility, $\mid$, as used in the previous example are reflexive and transitive. The relation of subset is also antisymmetric, as is that of divisibility provided we consider only positive integers.

The relation of *proper subset*, $\subset$, defined by $A \subset B$, provided that every element of A is also an element of B and $A \neq B$, is irreflexive, antisymmetric, and transitive. The relation of *less than or equal*, $\leq$, on a set of numbers, is reflexive, antisymmetric, and transitive, while the relation of *less than*, $<$, is irreflexive, antisymmetric, and transitive. □

A *partial order* on a set X is a reflexive, antisymmetric, and transitive relation R. A *strict partial order* on a set X is an irreflexive, antisymmetric, and transitive relation. Thus, $\subseteq$, $\leq$, and $|$ are partial orders, while $\subset$ and $<$ are strict partial orders.[16] If a relation R is a partial order, we generally use the usual inequality symbol "$\leq$" instead of R;[17] the relation $<$ defined by $a < b$ if and only if $a \leq b$ and $a \neq b$ is then a strict partial order. (Conversely, starting from a strict partial order $<$ on X, the relation $\leq$ defined by $a \leq b$ if and only if $a < b$ or $a = b$ is a partial order.)

A set X on which a partial order $\leq$ is defined is usually called a *partially ordered set* (or more simply, a *poset*) and denoted by $(X, \leq)$.

If R is a relation on a set X, then for x and y in X, x and y are *comparable*, provided that either $x\,R\,y$ or $y\,R\,x$; x and y are *incomparable* otherwise.[18] A partial order R on a set X is a *total order*, provided that every pair of elements of X is comparable. The standard relation $\leq$ on a set of numbers is a total order.[19]

If X is a finite set and we list the elements of X in some linear order $a_1, a_2, \ldots, a_n$ (a permutation of X), then by defining $a_i \leq a_j$ provided that $i \leq j$ (that is, provided that a_i comes before a_j in the permutation), it can be checked that we obtain a total order on X. We now show that every total order on X arises in this way.

Theorem 4.5.1 *Let X be a finite set with n elements. Then there is a one-to-one correspondence between the total orders on X and the permutations of X. In particular, the number of different total orders on X is $n!$.*

Proof. We show by induction on n that each total order $\leq$ on X corresponds to a permutation $a_1, a_2, \ldots, a_n$ of X with $a_1 < a_2 < \cdots < a_n$. If $n = 1$, this is trivial. Let $n > 1$. We first show that there is a *minimal element* of X; that is, an element a_1 such that $b \leq a_1$ implies that $b = a_1$ (equivalently, there is no element x with $x < a_1$). Let a be any element of X. If a is not a minimal element, then there is an element b such that $b < a$. If b is not a minimal element, there is an element c such that $c < b$ so that $c < b < a$. Continuing like this and using the fact that X is a finite set,

[16] The relation *is divisible by but does not equal* is also a strict partial order.

[17] It is important, then, to be aware that $a \leq b$ does not mean that a and b are numbers with a no bigger than b. The symbol "$\leq$" now becomes an abstract symbol for a partial order.

[18] Think of the phrase "x and y are incomparable" as an abstract version of the common phrase "one cannot compare apples and oranges," and so apples and oranges are incomparable.

[19] This is one reason why we should be careful to distinguish between the abstract symbol "$\leq$" for a partial order and the standard relation "$\leq$" on numbers; the latter is a total order where any two numbers a and b are comparable (either $a \leq b$ or $b \leq a$), but this property does not hold for a general partial order.

eventually we locate a minimal element a_1. Suppose there is an element $x \neq a_1$ of X such that $a_1 \not< x$. Since we have a total order, we must have $x < a_1$, contradicting the minimality of a_1. Hence, $a_1 < x$ for all x in X different from a_1. Applying induction to the set of $n-1$ elements of X different from a_1, we conclude that these elements can be ordered $a_2, a_3, \ldots, a_n$ with $a_2 < a_3 < \cdots < a_n$. Hence, $a_1, a_2, a_3, \ldots, a_n$ is a permutation of the elements of X with $a_1 < a_2 < a_3 < \cdots < a_n$. □

As a consequence of Theorem 4.5.1, a finite totally ordered set is often denoted as $a_1 < a_2 < \cdots < a_n$, or simply as a permutation $a_1, a_2, \ldots, a_n$.

A partially ordered set can be represented geometrically. To illustrate this, we need to define the cover relation of a partially ordered set $(X, \leq)$. Let a and b be in X. Then a is *covered by* b (also expressed as b *covers* a), denoted $a <_c b$, provided that $a < b$ and no element x can be squeezed between a and b; that is, there does not exist an element x such that both $a < x$ and $x < b$ hold. If X is a finite set, then, by transitivity, the partial order $\leq$ is uniquely determined by its cover relation. Thus, the cover relation is an efficient way to describe a partial order. It follows from Theorem 4.5.1 that, if $(X, \leq)$ is a totally ordered set, then the elements of X can be listed as $x_1, x_2, \ldots, x_n$ such that $x_1 <_c x_2 <_c \cdots <_c x_n$. It is for this reason that a totally ordered set is also called a *linearly ordered set.*

A *diagram* (sometimes called the Hasse diagram) of a finite partially ordered set $(X, \leq)$ is obtained by taking a point in the plane for each element of X, being careful to put the point for x below the point for y if $x <_c y$, and connecting x and y by a line segment if and only if x is covered by y. (We put x below y to signify that x is covered by y.)

Figure 4.6

Example. A totally ordered set of five elements is represented by the diagram, shown in Figure 4.6, of five vertical points, with four vertical line segments connecting the points. □

Example. The partially ordered set of subsets of the set $\{1, 2, 3\}$ ordered by containment is represented by the diagram, shown in Figure 4.7, of a cube "sitting" on one of its corners. □

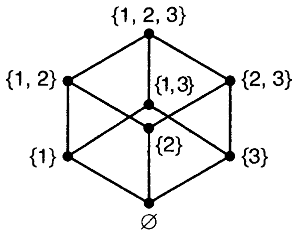

Figure 4.7

Example. The set of the first eight positive integers, partially ordered by "is a divisor of," is represented by the diagram in Figure 4.8. □

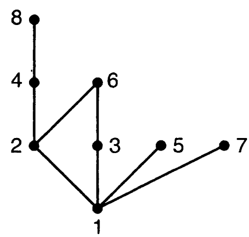

Figure 4.8

Let $\leq_1$ and $\leq_2$ be two partial orders on the same set X. Then the partially ordered set $(X, \leq_2)$ is an *extension* of the partially ordered set $(X, \leq_1)$, provided that whenever $a \leq_1 b$ holds, $a \leq_2 b$ also holds. In particular, an extension of a partially ordered set has more comparable pairs. We show that every finite partially ordered set $(X, \leq)$ has a *linear extension*; that is, an extension which is a linearly ordered set. This means that it is possible to list the elements of X in a linear order $x_1, x_2, \ldots, x_n$ so that x_i is listed before x_j whenever $x_i < x_j$; that is, if $x_i < x_j$, then $i < j$ (here $i < j$ means that i is a smaller integer than j).

Theorem 4.5.2 *Let $(X, \leq)$ be a finite partially ordered set. Then there is a linear extension of $(X, \leq)$.*

Proof. There is a very simple algorithm for listing the elements of X in a linear order $x_1, x_2, \ldots, x_n$ to obtain a linear extension of $(X, \leq)$:

Algorithm for a linear extension of a partially ordered set

(1) Choose a minimal element x_1 of X (with respect to the partial order $\leq$).

(2) Delete x_1 from X and choose a minimal element x_2 from among the remaining $n-1$ elements.

(3) Delete x_2 from X, and choose a minimal element x_3 from among the remaining $n-2$ elements.

(4) Delete x_3 from X, and choose a minimal element x_4 from among the remaining $n-3$ elements.

$\vdots$

(n) Delete x_{n-1} from X, leaving exactly one element x_n.

We show that $x_1, x_2, \ldots, x_n$ is a linear extension of $(X, \leq)$ by arguing by contradiction. Suppose there are x_i and x_j such that $x_i < x_j$ but $j < i$. Then, in step (j) of the preceding algorithm, when we chose x_j, x_i was among the remaining elements, and since $x_i < x_j$, x_j was not a minimal element as required by the algorithm. Thus, $x_1, x_2, \ldots, x_n$ is a linear extension of $(X, \leq)$. $\square$

Example. Let $X = \{1, 2, \ldots, n\}$ be the set consisting of the first n positive integers, and consider the partially ordered set $(X, |)$, where, as before, $|$ means "is a divisor of." Since, if $i \mid j$, then i is smaller than j, it follows that $1, 2, \ldots, n$ is a linear extension of $(X, \leq)$. $\square$

Example. Let X be a set of n elements, and consider the partially ordered set $(\mathcal{P}(X), \subseteq)$ of all subsets of X partially ordered by containment. Since $A \subseteq B$ implies that $|A| \leq |B|$, it follows that, if we start with the empty set and list all the one-element subsets in some order, then the two-element subsets in some order, then the three-element subsets in some order, and so on, we obtain a linear extension of $(\mathcal{P}(X), \subseteq)$. For instance, if $n = 3$ and $X = \{1, 2, 3\}$, then

$$\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$$

is a linear extension of $(\mathcal{P}(X), \subseteq)$. $\square$

We continue our discussion of partially ordered sets in Chapter 5.

We now define another special class of relations. Let X be a set. A relation R on X is an *equivalence relation* provided that it is reflexive, symmetric, and transitive. (Thus, an equivalence relation differs from a partial order only in that an equivalence relation is symmetric and a partial order is antisymmetric.) A relation that is an equivalence relation is usually denoted by "$\sim$". If $a \sim b$, then we say that a is *equivalent* to b. Just as a partial order can be considered as a generalization of the usual order "$\leq$" of numbers, an equivalence relation can be considered as a generalization of equality "$=$" of numbers. We now show that equivalence relations on X naturally correspond to partitions of X into nonempty sets.

Let $\sim$ be an equivalence relation on X. For each a in X, the *equivalence class* of a is the set

$$[a] = \{x : x \sim a\}$$

of all elements of X that are equivalent to a. Since $a \sim a$, the equivalence class of a contains a and thus is nonempty.

Example. Let X be a set of people, and define a relation R on X by $a\,R\,b$ provided that a and b have the same age. Then it is easy to check that R is an equivalence relation, and the equivalence class of person a is the subset of X consisting of all people with the same age as a. Observe that two equivalence classes that have a common person are, in fact, identical; thus the distinct equivalence classes partition X. The next theorem verifies that this phenomenon holds for all equivalence relations. $\square$

Theorem 4.5.3 *Let $\sim$ be an equivalence relation on a set X. Then the distinct equivalence classes partition X into nonempty parts. Conversely, given any partition of X into nonempty parts, there is an equivalence relation on X whose equivalence classes are the parts of the partition.*

Proof. First let "$\sim$" be an equivalence relation on X. We need to show that the different equivalence classes are pairwise disjoint and that their union is X. Each equivalence class is nonempty, and each element of X is contained in an equivalence class (the equivalence class of a contains a). It remains only to show that the distinct equivalence classes are pairwise disjoint, or, equivalently, that if two equivalence classes have a nonempty intersection, then they are identical sets. Suppose $[a] \cap [b] \neq \emptyset$, and let c be an element common to both $[a]$ and $[b]$. Then $c \sim a$ (and so $a \sim c$) and $c \sim b$ (and so $b \sim c$). Let x be contained in $[a]$. Then $x \sim a$. Since $a \sim c$ and $c \sim b$, transitivity implies that $a \sim b$ and then that $x \sim b$; hence x is contained in $[b]$. We conclude that $[a] \subseteq [b]$. In a similar way we conclude that $[b] \subseteq [a]$ and hence that $[a] = [b]$.

Conversely, let $A_1, A_2, \ldots, A_s$ be a partition of X into nonempty sets. For x and y in X, define $x \sim y$ if and only if x and y are in the same part of the partition. Then it is straightforward to check that "$\sim$" is an equivalence relation on X whose distinct equivalence classes are $A_1, A_2, \ldots, A_s$. See Exercise 44. $\square$

Example. Consider the set of $n!$ permutations of $1, 2, \ldots, n$. Define a relation R on this set by $i_1 i_2 \ldots i_n \; R \; j_1 j_2 \ldots j_n$ provided that there is an integer k such that $j_1 j_2 \ldots j_n = i_k \ldots i_n i_1 \ldots i_{k-1}$. This defines an equivalence relation (Check it!) where the set of equivalence classes are in one-to-one correspondence with the set of circular permutations of $1, 2, \ldots, n$. $\square$

4.6 Exercises

1. Which permutation of $\{1, 2, 3, 4, 5\}$ follows 31524 in using the algorithm described in Section 4.1? Which permutation comes before 31524?

2. Determine the mobile integers in

$$\overrightarrow{4}\ \overleftarrow{8}\ \overrightarrow{3}\ \overleftarrow{1}\ \overrightarrow{6}\ \overleftarrow{7}\ \overleftarrow{2}\ \overrightarrow{5}\,.$$

3. Use the algorithm of Section 4.1 to generate the first 50 permutations $\{1,2,3,4,5\}$, starting with $\overleftarrow{1}\ \overleftarrow{2}\ \overleftarrow{3}\ \overleftarrow{4}\ \overleftarrow{5}$.

4. Prove that in the algorithm of Section 4.1, which generates directly the permutations of $\{1,2,\ldots,n\}$, the directions of 1 and 2 never change.

5. Let $i_1i_2\cdots i_n$ be a permutation of $\{1,2,\ldots,n\}$ with inversion sequence $b_1,b_2,\ldots,b$ and let $k=b_1+b_2+\cdots+b_n$. Show by induction that we cannot bring $i_1i_2\cdots i_n$ to $12\cdots n$ by fewer than k successive switches of adjacent terms.

6. Determine the inversion sequences of the following permutations of $\{1,2,\ldots,8\}$:

   (a) 35168274

   (b) 83476215

7. Construct the permutations of $\{1,2,\ldots,8\}$ whose inversion sequences are

   (a) $2,5,5,0,2,1,1,0$

   (b) $6,6,1,4,2,1,0,0$

8. How many permutations of $\{1,2,3,4,5,6\}$ have

   (a) exactly 15 inversions?

   (b) exactly 14 inversions?

   (c) exactly 13 inversions?

9. Show that the largest number of inversions of a permutation of $\{1,2,\ldots,n\}$ equals $n(n-1)/2$. Determine the unique permutation with $n(n-1)/2$ inversions. Also determine all those permutations with one fewer inversion.

10. Bring the permutations 256143 and 436251 to 123456 by successive switches of adjacent numbers.

11. Let $S=\{x_7,x_6,\ldots,x_1,x_0\}$. Determine the 8-tuples of 0s and 1s corresponding to the following subsets of S:

    (a) $\{x_5,x_4,x_3\}$

    (b) $\{x_7,x_5,x_3,x_1\}$

    (c) $\{x_6\}$

12. Let $S = \{x_7, x_6, \ldots, x_1, x_0\}$. Determine the subsets of S corresponding to the following 8-tuples:

    (a) 00011011

    (b) 01010101

    (c) 00001111

13. Generate the 5-tuples of 0s and 1s by using the base 2 arithmetic generating scheme and identify them with subsets of the set $\{x_4, x_3, x_2, x_1, x_0\}$.

14. Repeat Exercise 13 for the 6-tuples of 0s and 1s.

15. For each of the following subsets of $\{x_7, x_6, \ldots, x_1, x_0\}$, determine the subset that immediately follows it by using the base 2 arithmetic generating scheme:

    (a) $\{x_4, x_1, x_0\}$

    (b) $\{x_7, x_5, x_3\}$

    (c) $\{x_7, x_5, x_4, x_3, x_2, x_1, x_0\}$

    (d) $\{x_0\}$

16. For each of the subsets (a), (b), (c), and (d) in the preceding exercise, determine the subset that immediately *precedes* it in the base 2 arithmetic generating scheme.

17. Which subset of $\{x_7, x_6, \ldots, x_1, x_0\}$ is 150th on the list of subsets of S when the base 2 arithmetic generating scheme is used? 200th? 250th? (As in Section 4.3, the places on the list are numbered beginning with 0.)

18. Build (the corners and edges of) the 4-cube, and indicate the reflected Gray code on it.

19. Give an example of a noncyclic Gray code of order 3.

20. Give an example of a cyclic Gray code of order 3 that is not the reflected Gray code.

21. Construct the reflected Gray code of order 5 by

    (a) using the inductive definition, and

    (b) using the Gray code algorithm.

22. Determine the reflected Gray code of order 6.

23. Determine the immediate successors of the following 9-tuples in the reflected Gray code of order 9:

(a) 010100110

(b) 110001100

(c) 111111111

24. Determine the predecessors of each of the 9-tuples in Exercise 23 in the reflected Gray code of order 9.

25. $*$ The reflected Gray code of order n is properly called the reflected *binary* Gray code since it is a listing of the n-tuples of 0s and 1s. It can be generalized to any base system, in particular the ternary and decimal system. Thus, the reflected decimal Gray code of order n is a listing of all the decimal numbers of n digits such that consecutive numbers in the list differ in only one place and the absolute value of the difference is 1. Determine the reflected decimal Gray codes of orders 1 and 2. (Note that we have not said precisely what a reflected decimal Gray code is. Part of the problem is to discover what it is.) Also, determine the reflected ternary Gray codes of orders $1, 2$, and 3.

26. Generate the 2-subsets of $\{1, 2, 3, 4, 5\}$ in lexicographic order by using the algorithm described in Section 4.4.

27. Generate the 3-subsets of $\{1, 2, 3, 4, 5, 6\}$ in lexicographic order by using the algorithm described in Section 4.4.

28. Determine the 6-subset of $\{1, 2, \ldots, 10\}$ that immediately follows $2, 3, 4, 6, 9, 10$ in the lexicographic order. Determine the 6-subset that immediately precedes $2, 3, 4, 6, 9, 10$.

29. Determine the 7-subset of $\{1, 2, \ldots, 15\}$ that immediately follows $1, 2, 4, 6, 8, 14, 15$ in the lexicographic order. Then determine the 7-subset that immediately precedes $1, 2, 4, 6, 8, 14, 15$.

30. Generate the inversion sequences of the permutations of $\{1, 2, 3\}$ in the lexicographic order, and write down the corresponding permutations. Repeat for the inversion sequences of permutations of $\{1, 2, 3, 4\}$.

31. Generate the 3-permutations of $\{1, 2, 3, 4, 5\}$.

32. Generate the 4-permutations of $\{1, 2, 3, 4, 5, 6\}$.

33. In which position does the subset 2489 occur in the lexicographic order of the 4-subsets of $\{1, 2, 3, 4, 5, 6, 7, 8, 9\}$?

34. Consider the r-subsets of $\{1, 2, \ldots, n\}$ in lexicographic order.

    (a) What are the first $(n - r + 1)$ r-subsets?

(b) What are the last $(r+1)$ r-subsets?

35. The *complement* $\overline{A}$ of an r-subset A of $\{1, 2, \ldots, n\}$ is the $(n-r)$-subset of $\{1, 2, \ldots, n\}$, consisting of all those elements that do not belong to A. Let $M = \binom{n}{r}$, the number of r-subsets and, at the same time, the number of $(n-r)$-subsets of $\{1, 2, \ldots, n\}$. Prove that, if

$$A_1, A_2, A_3, \ldots, A_M$$

are the r-subsets in lexicographic order, then

$$\overline{A_M}, \ldots, \overline{A_3}, \overline{A_2}, \overline{A_1}$$

are the $(n-r)$-subsets in lexicographic order.

36. Let X be a set of n elements. How many different relations on X are there? How many of these relations are reflexive? Symmetric? Antisymmetric? Reflexive and symmetric? Reflexive and anti-symmetric?

37. Let R' and R'' be two partial orders on a set X. Define a new relation R on X by $x\,R\,y$ if and only if both $x\,R'y$ and $x\,R''y$ hold. Prove that R is also a partial order on X. (R is called the *intersection* of R' and R''.)

38. Let $(X_1, \leq_1)$ and $(X_2, \leq_2)$ be partially ordered sets. Define a relation T on the set

$$X_1 \times X_2 = \{(x_1, x_2) : x_1 \text{ in } X_1, x_2 \text{ in } X_2\}$$

by

$$(x_1, x_2)\; T\; (x_1', x_2') \text{ if and only if } x_1 \leq_1 x_1' \text{ and } x_2 \leq_2 x_2'.$$

Prove that $(X_1 \times X_2, T)$ is a partially ordered set. $(X_1 \times X_2, T)$ is called the *direct product* of $(X_1, \leq_1)$ and $(X_2, \leq_2)$ and is also denoted by $(X_1, \leq_1) \times (X_2, \leq_2)$. More generally, prove that the direct product $(X_1, \leq_1) \times (X_2, \leq_2) \times \cdots \times (X_m, \leq_m)$ of partially ordered sets is also a partially ordered set.

39. Let $(J, \leq)$ be the partially ordered set with $J = \{0, 1\}$ and with $0 < 1$. By identifying the subsets of a set X of n elements with the n-tuples of 0s and 1s, prove that the partially ordered set $(X, \subseteq)$ can be identified with the n-fold direct product
$(J, \leq) \times (J, \leq) \times \cdots \times (J, \leq)$ (n factors).

40. Generalize Exercise 39 to the multiset of all combinations of the multiset $X = \{n_1 \cdot a_1, n_2 \cdot a_2, \ldots, n_m \cdot a_m\}$. (Part of this exercise is to determine the "natural" partial order of these multisets.)

41. Show that a partial order on a finite set is uniquely determined by its cover relation.

42. Describe the cover relation for the partial order $\subseteq$ on the collection $\mathcal{P}(X)$ of all subsets of a set X.

43. Let $X = \{a, b, c, d, e, f\}$ and let the relation R on X be defined by $a\,R\,b$, $b\,R\,c$, $c\,R\,d$, $a\,R\,e$, $e\,R\,f$, $f\,R\,d$. Verify that R is the cover relation of a partially ordered set, and determine all the linear extensions of this partial order.

44. Let $A_1, A_2, \ldots, A_s$ be a partition of a set X. Define a relation R on X by $x\,R\,y$ if and only if x and y belong to the same part of the partition. Prove that R is an equivalence relation.

45. Define a relation R on the set Z of all integers by $a\,R\,b$ if and only if $a = \pm b$. Is R an equivalence relation on Z? If so, what are the equivalence classes?

46. Let m be a positive integer and define a relation R on the set X of all nonnegative integers by $a\,R\,b$ if and only if a and b have the same remainder when divided by m. Prove that R is an equivalence relation on X. How many different equivalence classes does this equivalence relation have?

47. Let Π_n denote the set of all partitions of the set $\{1, 2, \ldots, n\}$ into *nonempty* sets. Given two partitions π and σ in Π_n, define $\pi \leq \sigma$, provided that each part of π is contained in a part of σ. Thus, the partition π can be obtained by partitioning the parts of σ. This relation is usually expressed by saying that π is a *refinement* of σ.

    (a) Prove that the relation of refinement is a partial order on Π_n.

    (b) By Theorem 4.5.3, we know that there is a one-to-one correspondence between Π_n and the set Λ_n of all equivalence relations on $\{1, 2, \ldots, n\}$. What is the partial order on Λ_n that corresponds to this partial order on Π_n?

    (c) Construct the diagram of $(\Pi_n, \leq)$ for $n = 1, 2, 3,$ and 4.

48. Consider the partial order $\leq$ on the set X of positive integers given by "is a divisor of." Let a and b be two integers. Let c be the largest integer such that $c \leq a$ and $c \leq b$, and let d be the smallest integer such that $a \leq d$ and $b \leq d$. What are c and d?

49. Prove that the intersection $R \cap S$ of two equivalence relations R and S on a set X is also an equivalence relation on X. Is the union of two equivalence relations on X always an equivalence relation?

50. Consider the partially ordered set $(X, \subseteq)$ of subsets of the set $X = \{a, b, c\}$ of three elements. How many linear extensions are there?

51. Let n be a positive integer, and let X_n be the set of $n!$ permutations of $\{1, 2, \ldots, n\}$ Let π and σ be two permutations in X_n, and define $\pi \leq \sigma$ provided that the set

of inversions of π is a subset of the set of inversions of σ. Verify that this defines a partial order on X_n, called the *inversion poset*. Describe the cover relation for this partial order and then draw the diagram for the inversion poset $(H_4, \leq)$.

52. Verify that a binary n-tuple $a_{n-1}\cdots a_1a_0$ is in place k in the Gray code order list where k is determined as follows: For $i = 0, 1, \ldots, n-1$, let
$$b_i = \begin{cases} 0 & \text{if } a_{n-1} + \cdots + a_i \text{ is even, and} \\ 1 & \text{if } a_{n-1} + \cdots + a_i \text{ is odd.} \end{cases}$$
Then
$$k = b_{n-1} \times 2^{n-1} + \cdots + b_1 \times 2 + b_0 \times 2^0.$$
Thus, $a_{n-1}\cdots a_1a_0$ is in the same place in the Gray code order list of binary n-tuples as $b_{n-1}\cdots b_1b_0$ is in the lexicographic order list of binary n-tuples.

53. Continuing with Exercise 52, show that $a_{n-1}\cdots a_1a_0$ can be recovered from $b_{n-1}\cdots b_1b_0$ by $a_{n-1} = b_{n-1}$, and for $i = 0, 1, \ldots, n-1$,
$$a_i = \begin{cases} 0 & \text{if } b_i + b_{i+1} \text{ is even, and} \\ 1 & \text{if } b_i + b_{i+1} \text{ is odd.} \end{cases}$$

54. Let $(X, \leq)$ be a finite partially ordered set. By Theorem 4.5.2 we know that $(X, \leq)$ has a linear extension. Let a and b be incomparable elements of X. Modify the proof of Theorem 4.5.2 to obtain a linear extension of $(X, \leq)$ such that $a < b$. (*Hint*: First find a partial order $\leq'$ on X such that whenever $x \leq y$, then $x \leq' y$ and, in addition, $a \leq' b$.)

55. Use Exercise 54 to prove that a finite partially ordered set is the intersection of all its linear extensions (see Exercise 37).

56. The *dimension* of a finite partially ordered set $(X, \leq)$ is the smallest number of its linear extensions whose intersection is $(X, \leq)$. By Exercise 55, every partially ordered set has a dimension. Those that have dimension 1 are the linear orders. Let n be a positive integer and let $i_1, i_2, \ldots, i_n$ be a permutation σ of $\{1, 2, \ldots, n\}$ that is different from $1, 2, \ldots, n$. Let $X = \{(1, i_1), (2, i_2), \ldots, (n, i_n)\}$. Now define a relation R on X by $(k, i_k)\, R\, (l, i_l)$ if and only if $k \leq l$ (ordinary integer inequality) and $i_k \leq i_l$ (again ordinary inequality); that is, (i_k, i_l) is not an inversion of σ. Thus, for instance, if $n = 3$ and $\sigma = 2, 3, 1$, then $X = \{(1,2), (2,3), (3,1)\}$, and $(1,2)\, R\, (2,3)$, but $(1,2) \not R\, (3,1)$. Prove that R is a partial order on X and that the dimension of the partially ordered set (X, R) is 2, provided that $i_1, i_2, \ldots, i_n$ is not the identity permutation $1, 2, \ldots, n$.

57. Consider the set of all permutations $i_1 i_2 \ldots, i_n$ of $1, 2, \ldots, n$ such that $i_k \neq k$ for $k = 1, 2, \ldots, n$. (Such permutations are called *derangements* and are discussed in Chapter 6.) Describe an algorithm for generating a random derangement (modify the algorithm given in Section 4.1 for generating a random permutation).

58. Consider the complete graph K_n defined in Chapter 2, in which each edge is colored either red or blue. Define a relation on the n points of K_n by saying that one point is related to another point provided that the edge joining them is colored red. Determine when this relation is an equivalence relation, and, when it is, determine the equivalence classes.

59. Let $n \geq 2$ be an integer. Prove that the total number of inversions of *all* $n!$ permutations of $1, 2, \ldots, n$ equals

$$\frac{1}{2}n!\binom{n}{2} = n!\frac{n(n-1)}{4}.$$

(*Hint*: Pair up the permutations so that the number of inversions in each pair is $n(n-1)/2$.)