SECTION 5.6 Generating Permutations and Combinations

This section is quite different from the rest of this chapter. It is really about algorithms and programming. These algorithms are not easy, and it would be worthwhile to "play computer" with them to get a feeling for how they work. In constructing such algorithms yourself, try assuming that you will list the permutations or combinations in a nice order (such as lexicographic order); then figure out how to find the "next" one in this order.

1. Lexicographic order is the same as numerical order in this case, so the ordering from smallest to largest is 14532, 15432, 21345, 23451, 23514, 31452, 31542, 43521, 45213, 45321.

3. We use Algorithm 1 to find the next permutation. Our notation follows that algorithm, with j being the largest subscript such that $a_j < a_{j+1}$ and k being the subscript of the smallest number to the right of a_j that is larger than a_j.

a) Since $4 > 3 > 2$, we know that the 1 is our a_j. The smallest integer to the right of 1 and greater than 1 is 2, so $k = 4$. We interchange a_j and a_k, giving the permutation 2431, and then we reverse the entire substring to the right of the position now occupied by the 2, giving the answer 2134.

b) The first integer from the right that is less than its right neighbor is the 2 in position 4. Therefore $j = 4$ here, and of course k has to be 5. The next permutation is the one that we get by interchanging the fourth and fifth numbers, 54132. (Note that the last phase of the algorithm, reversing the end of the string, was vacuous this time—there was only one element to the right of position 4, so no reversing was necessary.)

c) Since $5 > 3$, we know that the 4 is our a_j. The smallest integer to the right of 4 and greater than 4 is $a_k = 5$. We interchange a_j and a_k, giving the permutation 12543, and then we reverse the entire substring to the right of the position now occupied by the 5, giving the answer 12534.

d) Since $3 > 1$, we know that the 2 is our a_j. The smallest integer to the right of 2 and greater than 2 is $a_k = 3$. We interchauge a_j and a_k, giving the permutation 45321, and then we reverse the entire substring to the right of the position now occupied by the 3, giving the answer 45312.

e) The first integer from the right that is less than its right neighbor is the 3 in position 6. Therefore $j = 6$ here, and of course k has to be 7. The next permutation is the one that we get by interchanging the sixth and seventh numbers, 6714253. As in part **(b)**, no reversing was necessary.

f) Since $8 > 7 > 6 > 4$, we know that the 2 is our a_j, so $j = 4$. The smallest integer to the right of 2 and greater than 2 is $a_8 = 4$. We interchange a_4 and a_8, giving the permutation 31548762, and then we reverse the entire substring to the right of the position now occupied by the 4, giving the answer 31542678.

5. We begin with the permutation 1234. Then we apply Algorithm 1 23 times in succession, giving us the other 23 permutations in lexicographic order: 1243, 1324, 1342, 1423, 1432, 2134, 2143, 2314, 2341, 2413, 2431, 3124, 3142, 3214, 3241, 3412, 3421, 4123, 4132, 4213, 4231, 4312, and 4321. The last permutation is the one entirely in decreasing order. Each application of Algorithm 1 follows the pattern in Exercise 3.

7. We begin with the first 3-combination, namely $\{1,2,3\}$. Let us trace through Algorithm 3 to find the next. Note that $n = 5$ and $r = 3$; also $a_1 = 1$, $a_2 = 2$, and $a_3 = 3$. We set i equal to 3 and then decrease i until $a_i \neq 5 - 3 + i$. This inequality is already satisfied for $i = 3$, since $a_3 \neq 5$. At this point we increment a_i by 1 (so that now $a_3 = 4$), and fill the remaining spaces with consecutive integers following a_i (in this case there are no more remaining spaces). Thus our second 3-combination is $\{1,2,4\}$. The next call to Algorithm 3 works the same way, producing the third 3-combination, namely $\{1,2,5\}$. To get the fourth 3-combination, we again call Algorithm 3. This time the i that we end up with is 2, since $5 = a_3 = 5 - 3 + 3$. Therefore the second element in the list is incremented, namely goes from a 2 to a 3, and the third element is the next larger element after 3, namely 4. Thus this 3-combination is $\{1,3,4\}$. Another call to the algorithm gives us $\{1,3,5\}$, and another call gives us $\{1,4,5\}$. Now when we call the algorithm, we find $i = 1$ at the end of the **while** loop, since in this case the last two elements are the two largest elements in the set. Thus a_1 is increased to 2, and the remainder of the list is filled with the next two consecutive integers, giving us $\{2,3,4\}$. Continuing in this manner, we get the rest of the 3-combinations: $\{2,3,5\}$, $\{2,4,5\}$, $\{3,4,5\}$.

9. Clearly the next larger r-combination must differ from the old one in position i, since positions $i+1$, $i+2$, $\ldots$, r are occupied by the largest possible numbers (namely $i+n-r+1$ to n). Also $a_i + 1$ is the smallest possible number that can be put in position i if we want an r-combination greater than the given one, and then similarly $a_i + 2$, $a_i + 3$, $\ldots$, $a_i + r - i + 1$ are the smallest allowable numbers for positions $i+1$ to r. Therefore there is no r-combination between the given one and the one that Algorithm 3 produces, which is exactly what we had to prove.

11. One way to do this problem (and to have done Exercise 10) is to generate the r-combinations using Algorithm 3, and then to find all the permutations of each, using Algorithm 1 (except that now the elements to be permuted are not the integers from 1 to r, but are instead the r elements of the r-combination currently being used). Thus we start with the first 3-combination, $\{1,2,3\}$, and we list all 6 of its permutations: 123, 132, 213, 231, 312, 321. Next we find the next 3-combination, namely $\{1,2,4\}$, and list all of its permutations: 124, 142, 214, 241, 412, 421. We continue in this manner to generate the remaining 48 3-permutations of $\{1,2,3,4,5\}$: 125, 152, 215, 251, 512, 521; 134, 143, 314, 341, 413, 431; 135, 153, 315, 351, 513, 531; 145, 154, 415, 451, 514, 541; 234, 243, 324, 342, 423, 432; 235, 253, 325, 352, 523, 532; 245, 254, 425, 452, 524, 542; 345, 354, 435, 453, 534, 543. There are of course $P(5,3) = 5 \cdot 4 \cdot 3 = 60$ items in our list.

13. One way to show that a function is a bijection is to find its inverse, since only bijections can have inverses. Note that the sizes of the two sets in question are the same, since there are $n!$ nonnegative integers less than $n!$, and there are $n!$ permutations of $\{1,2,\ldots,n\}$. In this case, since Cantor expansions are unique, we need to take the digits a_1, a_2, $\ldots$, a_{n-1} of the Cantor expansion of a nonnegative integer m less than $n!$ (so that $m = a_1 1! + a_2 2! + \cdots + a_{n-1}(n-1)!$), and produce a permutation with these a_k's satisfying the definition given before Exercise 12—indeed the only such permutation.

We will fill the positions in the permutation one at a time. First we put n into position $n - a_{n-1}$; clearly a_{n-1} will be the number of integers less than n that follow n in the permutation, since exactly a_{n-1} positions remain empty to the right of where we put the n. Next we renumber the free positions (the ones other than the one into which we put n), from left to right, as $1, 2, \ldots, n-1$. Under this numbering, we put $n-1$ into position $(n-1) - a_{n-2}$. Again it is clear that a_{n-2} will be the number of integers less than $n-1$ that follow $n-1$ in the permutation. We continue in this manner, renumbering the free positions, from left to right, as $1, 2, \ldots, n-k+1$, and then placing $n-k+1$ in position $(n-k+1) - a_{n-k}$, for $k = 1, 2, \ldots, n-1$. Finally we place 1 in the only remaining position.

15. The algorithm is really given in our solution to Exercise 13. To produce all the permutations, we find the permutation corresponding to i, where $0 \leq i < n!$, under the correspondence given in Exercise 13. To do this, we need to find the digits in the Cantor expansion of i, starting with the digit a_{n-1}. In what follows, that digit will be called c. We use k to keep track of which digit we are working on; as k goes from 1 to $n-1$, we will be computing the digit a_{n-k} in the Cantor expansion and inserting $n-k+1$ into the proper location in the permutation. (At the end, we need to insert 1 into the only remaining position.) We will call the positions in the permutation $p_1, p_2, \ldots, p_n$. We write only the procedure that computes the permutation corresponding to i; obviously to get all the permutations we simply call this procedure for each i from 0 to $n!-1$.

```
procedure Cantor permutation(n, i : integers with n ≥ 1 and 0 ≤ i < n!)
x := n  {to help in computing Cantor digits}
for j := 1 to n  {initialize permutation to be all 0's}
        p_j := 0
for k := 1 to n − 1  {figure out where to place n − k + 1}
begin
        c := ⌊x/(n − k)!⌋  {the Cantor digit}
        x := x − c(n − k)!  {what's left of x}
        h := n  {now find the (c + 1)^th free position from the right}
        while p_h ≠ 0
                h := h − 1
        for j := 1 to c
        begin
                h := h − 1
                while p_h ≠ 0
                        h := h − 1
        end
        p_h := n − k + 1  {here is the key step}
end
h := 1  {now find the last free position}
while p_h ≠ 0
        h := h + 1
p_h := 1
{p_1, p_2, ..., p_n is the permutation corresponding to i}
```