LECTURE 27

PRE- AND POST-CONDITIONS OF AN ALGORITHM
LOOP INVARIANTS
LOOP INVARIANT THEOREM

ALGORITHM:
The word "algorithm" refers to a step-by-step method for performing some action. A computer program is, similarly, a set of instructions that are executed step-by-step for performing some specific task. Algorithm, however, is a more general term in that the term program refers to a particular programming language.

INFORMATION ABOUT ALGORITHM:
The following information is generally included when describing algorithms formally:
1.The name of the algorithm, together with a list of input and output variables.
2.A brief description of how the algorithm works.
3.The input variable names, labeled by data type.
4.The statements that make the body of the algorithm, with explanatory comments.
5.The output variable names, labeled by data type.
6.An end statement.

THE DIVISION ALGORITHM

THEOREM (Quotient-Remainder Theorem):
Given any integer n and a positive integer d, there exist unique integers q and r such that $n = d \cdot q + r$ and $0 \leq r < d$.

Example:
a) $n = 54, d = 4$ $\quad 54 = 4 \cdot 13 + 2;$ $\quad$ hence $q = 13, r = 2$
b) $n = -54, d = 4$ $\quad -54 = 4 \cdot (-14) + 2;$ $\quad$ hence $q = -14, r = 2$
c) $n = 54, d = 70$ $\quad 54 = 70 \cdot 0 + 54;$ $\quad$ hence $q = 0, r = 54$

ALGORITHM (DIVISION):
{Given a nonnegative integer a and a positive integer d, the aim of the algorithm is to find integers q and r that satisfy the conditions $a = d \cdot q + r$ and $0 \leq r < d$.
This is done by subtracting d repeatedly from a until the result is less than d but is still nonnegative.
The total number of d's that are subtracted is the quotient q. The quantity $a - d \cdot q$ equals the remainder r.}
Input: a {a nonnegative integer}, d {a positive integer}
Algorithm body: $r := a, q := 0$
{Repeatedly subtract d from r until a number less than d is obtained. Add 1 to d each time d is subtracted.}

while $(r \geq d)$
$r := r - d \qquad q := q + 1$
end while
Output: q, r
end Algorithm (Division)

TRACING THE DIVISION ALGORITHM

Example:

Trace the action of the Division Algorithm on the input variables $a = 54$ and $d = 11$

Solution

Iteration Number

Variable Names

	0	1	2	3	4
a	54				
d	11				
r	54	43	32	21	10
q	0	1	2	3	4

PREDICATE

Consider the sentence
"Aslam is a student at the Virtual University."
let *P* stand for the words
"is a student at the Virtual University"
and let *Q* stand for the words
"is a student at."
Then both *P* and *Q* are *predicate symbols*.
The sentences "x is a student at the Virtual University" and "x is a student at y" are symbolized as $P(x)$ and $Q(x, y)$, where x and y are predicate variables and take values in appropriate sets. When concrete values are substituted in place of predicate variables, a statement results.

DEFINITION:

A predicate is a sentence that contains a finite number of variables and becomes a statement when specific values are substituted for the variables.
The domain of a predicate variable is the set of all values that may be substituted in place of the variable.

PRE-CONDITIONS AND POST-CONDITIONS:

Consider an algorithm that is designed to produce a certain final state from a given state. Both the initial and final states can be expressed as predicates involving the input and output variables.
Often the predicate describing the initial state is called the **pre-condition of the algorithm** and the predicate describing the final state is called the **post-condition of the algorithm**.

EXAMPLE:

1.Algorithm to compute a product of two nonnegative integers
pre-condition: The input variables m and n are nonnegative integers.
pot-condition: The output variable p equals $m \cdot n$.

2.Algorithm to find the quotient and remainder of the division of one positive integer by another
pre-condition: The input variables a and b are positive integers.
pot-condition: The output variable q and r are positive integers such that

$$a = b \cdot q + r \text{ and } 0 \leq r < b.$$

3.Algorithm to sort a one-dimensional array of real numbers
Pre-condition: The input variable $A[1], A[2], \ldots A[n]$ is a one-dimensional array of real numbers.
post-condition:The input variable $B[1], B[2], \ldots B[n]$ is a one-dimensional array of real numbers with same elements as $A[1], A[2], \ldots A[n]$ but with the property that $B[i] \leq B[j]$ whenever $i \leq j$.

THE DIVISION ALGORITHM:

[pre-condition: a is a nonnegative integer and
d is a positive integer, $r = a$, and $q = 0$]
while $(r \geq d)$
1. $r := r - d$
2. $q := q + 1$

end while
[post-condition: q and r are nonnegative integers
with the property that $a = q \cdot d + r$ and $0 \leq r < d$.]

LOOP INVARIANTS:

The method of loop invariants is used to prove correctness of a loop with respect to certain pre and post-conditions. It is based on the principle of mathematical induction.
[pre-condition for loop]

while (G)
[Statements in body of loop. None contain branching statements that lead outside the loop.]

end while[post-condition for loop]

<u>DEFINITION:</u>

A loop is defined as **correct with respect to its pre- and post-conditions** if, and only if, whenever the algorithm variables satisfy the pre-condition for the loop and the loop is executed, then the algorithm variables satisfy the post-condition of the loop.

<u>THEOREM</u>:

Let a **while** loop with guard *G* be given, together with pre- and post conditions that are predicates in the algorithm variables.

Also let a predicate $I(n)$, called the **loop invariant**, be given. If the following four properties are true, then the loop is correct with respect to its pre- and post-conditions.

I.Basis Property: The pre-condition for the loop implies that $I(0)$ is true before the first iteration of the loop.

II.Inductive property: If the guard *G* and the loop invariant $I(k)$ are both true for an integer $k \geq 0$ before an iteration of the loop, then $I(k + 1)$ is true after iteration of the loop.

III.Eventual Falsity of Guard: After a finite number of iterations of the loop, the guard becomes false.

IV.Correctness of the Post-Condition: If *N* is the least number of iterations after which *G* is false and $I(N)$ is true, then the values of the algorithm variables will be as specified in the post-condition of the loop.

PROOF:

Let I(n) be a predicate that satisfies properties I-IV of the loop invariant theorem.

Properties I and II establish that:

For all integers $n \geq 0$, if the while loop iterates n times, then I(n) is true.

Property III indicates that the guard G becomes false after a finite number N of iterations.

Property IV concludes that the values of the algorithm variables are as specified by the post-condition of the loop.

LECTURE 28

CORRECTNESS OF:
LOOP TO COMPUTE A PRODUCT
THE DIVISION ALGORITHM
THE EUCLIDEAN ALGORITHM

A LOOP TO COMPUTE A PRODUCT:

[pre-condition: m is a nonnegative integer,
x is a real number, $i = 0$, and product = 0.]

while (i # m)
1. product := product + x
2. $i := i + 1$

end while
[post-condition: product = $m \cdot x$]

PROOF:

Let the loop invariant be
$I(n)$: $i = n$ and product = $n \cdot x$
The guard condition G of the while loop is
G: $i \# m$

I.Basis Property:

[$I(0)$ is true before the first iteration of the loop.]
$I(0)$: $i = 0$ and product = $0 \cdot x = 0$
Which is true before the first iteration of the loop.

II.Inductive property:

[If the guard G and the loop invariant $I(k)$ are both true before a loop iteration (where $k \geq 0$), then $I(k + 1)$ is true after the loop iteration.]
Before execution of statement 1,

$$product_{old} = k \cdot x.$$

Thus the execution of statement 1 has the following effect:

$$product_{new} = product_{old} + \mathrm{x} = k \cdot x + \mathrm{x} = (k + 1) \cdot x$$

Similarly, before statement 2 is executed,

$$i_{old} = k,$$

So after execution of statement 2,

$$i_{new} = i_{old} + 1 = k + 1.$$

Hence after the loop iteration, the statement $I(k +1)$ (i.e., $i = k + 1$ and product = $(k + 1) \cdot x$) is true. This is what we needed to show.

III.Eventual Falsity of Guard:

[After a finite number of iterations of the loop, the guard becomes false.]

IV.Correctness of the Post-Condition:

[If N is the least number of iterations after which G is false and $I(N)$ is true, then the values of the algorithm variables will be as specified in the post-condition of the loop.]

THE DIVISION ALGORITHM:

[pre-condition: a is a nonnegative integer and
d is a positive integer, $r = a$, and $q = 0$]

while ($r \geq d$)
1. $r := r - d$
2. $q := q + 1$

end while

[post-condition: q and r are nonnegative integers
with the property that $a = q \cdot d + r$ and $0 \leq r < d$.]

PROOF:

Let the loop invariant be
$I(n)$: $r = a - n \cdot d$ and $n = q$.

The guard of the **while** loop is
G: $r \geq d$

I.Basis Property:

[$I(0)$ is true before the first iteration of the loop.]
$I(0)$: $r = a - 0 \cdot d = a$ and $0 = q$.

II.Inductive property:

[If the guard G and the loop invariant $I(k)$ are both true before a loop iteration (where $k \geq 0$), then $I(k + 1)$ is true after the loop iteration.]

$I(k)$: $r = a - k \cdot d \geq 0$ and $k = q$

$I(k + 1)$: $r = a - (k + 1) \cdot d \geq 0$ and $k + 1 = q$

$r_{new} = r - d$
$= a - k \cdot d - d$
$= a - (k + 1) \cdot d$

$q = q + 1$
$= k + 1$

also

$r_{new} = r - d$
$\geq d - d = 0$ (since $r \geq 0$)

Hence $I(k + 1)$ is true.

III.Eventual Falsity of Guard:

[After a finite number of iterations of the loop, the guard becomes false.]

IV.Correctness of the Post-Condition:

[If N is the least number of iterations after which G is false and $I(N)$ is true, then the values of the algorithm variables will be as specified in the post-condition of the loop.]

G is false and I(N) is true.

That is, $r \geq d$ and $r = a - N \cdot d \geq 0$ and $N = q$.

or $\quad r = a - q \cdot d$

or $\quad a = q \cdot d + r$

Also combining the two inequalities involving r we get

$$0 \leq r < d$$

<u>THE EUCLIDEAN ALGORITHM:</u>

The greatest common divisor (gcd) of two integers a and b is the largest integer that divides both a and b. For example, the gcd of 12 and 30 is 6.

The Euclidean algorithm takes integers A and B with $A > B \geq 0$ and compute their greatest common divisor.

<u>HAND CALCULATION OF gcd</u>:

Use the Euclidean algorithm to find gcd(330, 156)

<u>SOLUTION:</u>

$$\begin{array}{r|r} & 2 \\ \hline 156 & 330 \\ & 312 \\ \hline & 18 \end{array} \qquad \begin{array}{r|r} & 8 \\ \hline 18 & 156 \\ & 144 \\ \hline & 12 \end{array}$$

$$\begin{array}{r|r} & 1 \\ \hline 12 & 18 \\ & 12 \\ \hline & 6 \end{array} \qquad \begin{array}{r|r} & 2 \\ \hline 6 & 12 \\ & 12 \\ \hline & 0 \end{array}$$

Hence gcd(330, 156) = 6

<u>EXAMPLE:</u>

Use the Euclidean algorithm to find gcd(330, 156)

<u>Solution:</u>
1.Divide 330 by 156:
This gives $330 = 156 \cdot 2 + 18$
2.Divide 156 by 18:
This gives $156 = 18 \cdot 8 + 12$
3.Divide 18 by 12:
This gives $18 = 12 \cdot 1 + 6$
4.Divide 12 by 6:
This gives $12 = 6 \cdot 2 + 0$
Hence gcd(330, 156) = 6.

<u>LEMMA</u>:
If a and b are any integers with b# 0 and q and r are nonnegative integers such that

$$a = q \cdot d + r$$

then

$$\gcd(a, b) = \gcd(b, r)$$

[pre-condition: A and B are integers with
$A > B \geq 0$, $a = A$, b = B, $r = B$.]

while (b # 0)
1. $r := a \bmod b$
2. $a := b$
3. b := r

end while[post-condition: a = gcd(A, B)]

PROOF:
Let the **loop invariant** be
$I(n)$: $\gcd(a, b) = \gcd(A, B)$ and $0 \leq b < a$.
The guard of the **while** loop is
G: b # 0

I.Basis Property:
[$I(0)$ is true before the first iteration of the loop.]
$I(0)$: $\gcd(a, b) = \gcd(A, B)$ and $0 \leq b < a$.
According to the precondition,
a = A, b = B, r = B, and $0 \leq B < A$.
Hence $I(0)$ is true before the first iteration of the loop.

II.Inductive property:
[If the guard G and the loop invariant $I(k)$ are both true before a loop iteration (where k $\geq$ 0), then $I(k + 1)$ is true after the loop iteration.]
Since $I(k)$ is true before execution of the loop we have,
$\gcd(a_{old}, b_{old}) = \gcd(A, B)$ and $0 \leq b_{old} < a_{old}$
After execution of statement 1,
$r_{new} = a_{old} \bmod b_{old}$Thus,
$a_{old} = b_{old} \cdot q + r_{new}$ for some integer q
with,

$$0 \le r_{new} < b_{old}.$$

But

$$\gcd(a_{old}, b_{old}) = \gcd(b_{old}, r_{old})$$

and we have,

$$\gcd(b_{old}, r_{new}) = \gcd(A, B)$$

When statements 2 and 3 are executed,

$$a_{new} = b_{old} \text{ and } b_{new} = r_{new}$$

It follows that

$$\gcd(a_{new}, b_{new}) = \gcd(A, B)$$

Also,

$$0 \le r_{new} < b_{old}$$

becomes

$$0 \le b_{new} < a_{new}$$

Hence $I(k + 1)$ is true.

III.Eventual Falsity of Guard:

[After a finite number of iterations of the loop, the guard becomes false.]

IV.Correctness of the Post-Condition:

[If N is the least number of iterations after which G is false and $I(N)$ is true, then the values of the algorithm variables will be as specified in the post-condition of the loop.]

LECTURE 29

COMBINATORICS:

THE SUM RULE
THE PRODUCT RULE

COMBINATORICS:
Combinatorics is the mathematics of counting and arranging objects.Counting of objects with certain properties (enumeration) is required to solve many different types of problem .For example,counting is used to:
(i) Determine number of ordered or unordered arrangement of objects.
(ii)Generate all the arrangements of a specified kind which is important in computer simulations.
(iii)Compute probabilities of events.
(iv)Analyze the chance of winning games, lotteries etc.
(v)Determine the complexity of algorithms.

THE SUM RULE:
If one event can occur in n_1 ways, a second event can occur in n_2 (different) ways, then the total number of ways in which exactly one of the events (i.e., first or second) can occur is $n_1 + n_2$.

EXAMPLE:
Suppose there are 7 different optional courses in Computer Science and 3 different optional courses in Mathematics. Then there are $7 + 3 = 10$
choices for a student who wants to take one optional course.

EXERCISE:
A student can choose a computer project from one of the three lists. The three lists contain 23, 15 and 19 possible projects, respectively. How many possible projects are there to choose from?

SOLUTION:
The student can choose a project from the first list in 23 ways, from the second list in 15 ways, and from the third list in 19 ways. Hence, there are
$23 + 15 + 19 = 57$ projects to choose from.

GENERALIZED SUM RULE:
If one event can occur in n_1 ways,
a second event can occur in n_2 ways,
a third event can occur in n_3 ways,
……………………………………..

then there are
$n_1 + n_2 + n_3 + \dots$
ways in which exactly one of the events can occur.

SUM RULE IN TERMS OF SETS:

If $A_1, A_2, \ldots, A_m$ are finite disjoint sets, then the number of elements in the union of these sets is the sum of the number of elements in them.
If $n(A_i)$ denotes the number of elements in set A_i, then

$n(A_1 \cup A_2 \cup \ldots \cup A_m) = n(A_1) + n(A_2) + \ldots + n(A_m)$
where $A_i \cap A_j = \phi$ if $i \# j$

THE PRODUCT RULE:

If one event can occur in n_1 ways and if for each of these n_1 ways, a second event can occur in n_2 ways, then the total number of ways in which both events occur is $n_1 \cdot n_2$.

EXAMPLE:

Suppose there are 7 different optional courses in Computer Science and 3 different optional courses in Mathematics. A student who wants to take one optional course of each subject, there are

$7 \times 3 = 21$ choices.

EXAMPLE:

The chairs of an auditorium are to be labeled with two characters, a letter followed by a digit.
What is the largest number of chairs that can be labeled differently?

SOLUTION:

The procedure of labeling a chair consists of two events, namely,
(i) Assigning one of the 26 letters: A, B, C, …, Z and
(ii)Assigning one of the 10 digits: 0, 1, 2, …, 9
By product rule, there are $26 \times 10 = 260$
different ways that a chair can be labeled by both a letter and a digit.

GENERALIZED PRODUCT RULE:

If some event can occur in n_1 different ways, and if, following this event, a second event can occur in n_2 different ways, and following this second event, a third event can occur in n_3 different ways, …, then the number of ways all the events can occur in the order indicated is

$$n_1 \cdot n_2 \cdot n_3 \cdot \ldots$$

PRODUCT RULE IN TERMS OF SETS:

If $A_1, A_2, \ldots, A_m$ are finite sets, then the number of elements in the Cartesian product of these sets is the product of the number of elements in each set.
If $n(A_i)$ denotes the number of elements in set A_i, then

$$n(A_1 \times A_2 \times \ldots \times A_m) = n(A_1) \cdot n(A_2) \cdot \ldots \cdot n(A_{m,})$$

EXERCISE:

Find the number n of ways that an organization consisting of 15 members can elect a president, treasurer, and secretary. (assuming no person is elected to more than one position)

SOLUTION:

The president can be elected in 15 different ways; following this, the treasurer can be elected in 14 different ways; and following this, the secretary can be elected in 13 different ways. Thus, by product rule, there are

$$n = 15 \times 14 \times 13 = 2730$$

different ways in which the organization can elect the officers.

EXERCISE:

There are four bus lines between A and B; and three bus lines between B and C. Find the number of ways a person can travel:

(a) By bus from A to C by way of B;

(b) Round trip by bus from A to C by way of B;

(c) Round trip by bus from A to C by way of B, if the person does not want to use a bus line more than once.

SOLUTION:

(a) There are 4 ways to go from A to B and 3 ways to go from B to C; hence there are $4 \times 3 = 12$ ways to go from A to C by way of B.

(b) The person will travel from A to B to C to B to A for the round trip.

i.e (A →B →C →B →A)

The person can travel 4 ways from A to B and 3 way from B to C and back.

i.e., $A \xrightarrow{4} B \xrightarrow{3} C \xrightarrow{3} B \xrightarrow{4} A$

Thus there are $4 \times 3 \times 3 \times 4 = 144$ ways to travel the round trip.

(c) The person can travel 4 ways from A to B and 3 ways from B to C, but only 2 ways from C to B and 3 ways from B to A, since bus line cannot be used more than once. Thus

i.e., $A \xrightarrow{4} B \xrightarrow{3} C \xrightarrow{2} B \xrightarrow{3} A$

Hence there are $4 \times 3 \times 2 \times 3 = 72$ ways to travel the round trip without using a bus line more than once.

EXERCISE:

A bit string is a sequence of 0's and 1's.How many bit string are there of length 4?

SOLUTION:

Each bit (binary digit) is either 0 or 1.

Hence, there are 2 ways to choose each bit.Since we have to choose four bits therefore, the product rule shows, there are a total of $2 \times 2 \times 2 \times 2 = 2^4 = 16$ different bit strings of length four.

EXERCISE:

How many bit strings of length 8

(i) begin with a 1? (ii) begin and end with a 1?

SOLUTION:

(i) If the first bit (left most bit) is a 1, then it can be filled in only one way. Each of the remaining seven positions in the bit string can be filled in 2 ways (i.e., either by 0 or 1). Hence, there are $1 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^7 = 128$ different bit strings of length 8 that begin with a 1.

(ii) If the first and last bit in an 8 bit string is a 1, then only the intermediate six bits can be filled in 2 ways, i.e. by a 0 or 1. Hence there are $1 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 1 = 2^6 = 64$ different bit strings of length 8 that begin and end with a 1.

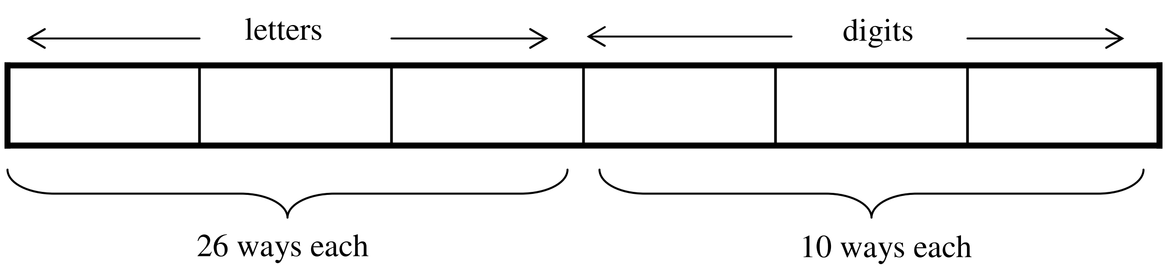
EXERCISE:

Suppose that an automobile license plate has three letters followed by three digits.

(a) How many different license plates are possible?

SOLUTION:

Each of the three letters can be written in 26 different ways, and each of the three digits can be written in 10 different ways.



Hence, by the product rule, there is a total of

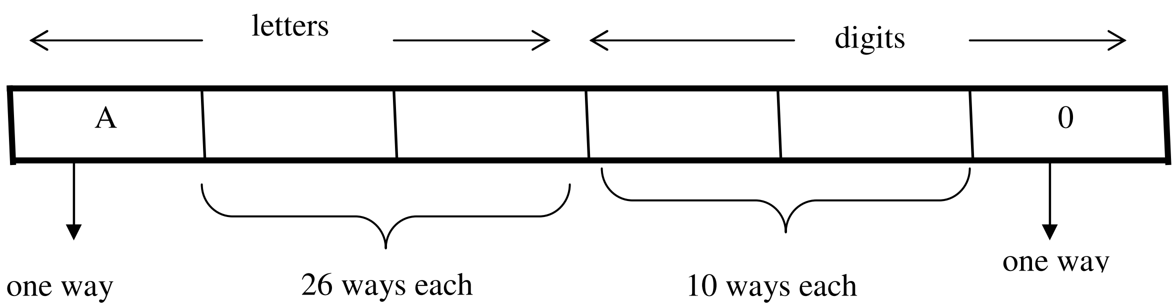
$$26 \times 26 \times 26 \times 10 \times 10 \times 10 = 17,576,000$$

different license plates possible.

(b) How many license plates could begin with A and end on 0?

SOLUTION:

The first and last place can be filled in one way only, while each of second and third place can be filled in 26 ways and each of fourth and fifth place can be filled in 10 ways.

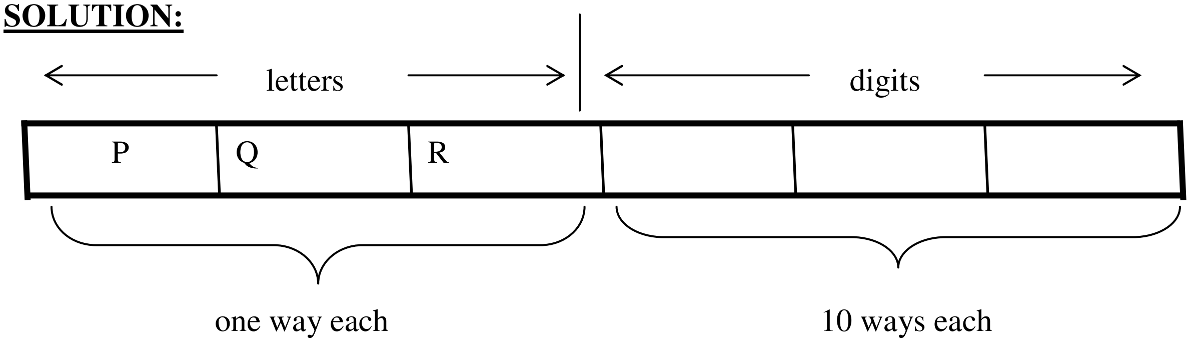


Number of license plates that begin with A and end in 0 are

$1 \times 26 \times 26 \times 10 \times 10 \times 1 = 67600$

(c) How many license plates begin with PQR

SOLUTION:



Number of license plates that begin with PQR are

$1 \times 1 \times 1 \times 10 \times 10 \times 10 = 1000$

(d) How many license plates are possible in which all the letters and digits are distinct?

SOLUTION:

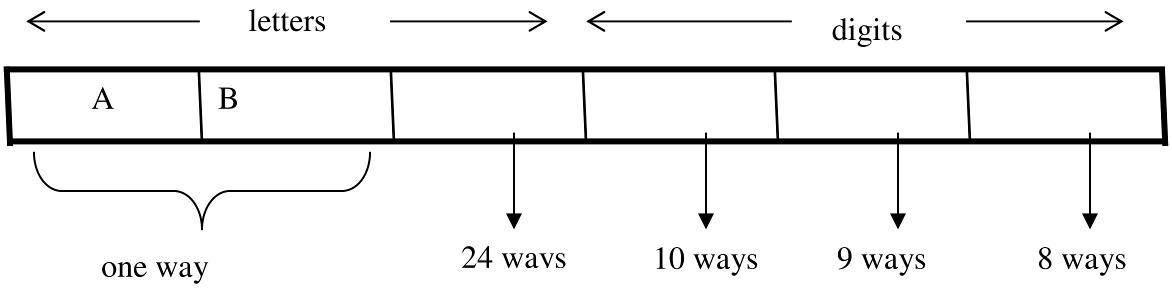
The first letter place can be filled in 26 ways. Since, the second letter place should contain a different letter than the first, so it can be filled in 25 ways. Similarly, the third letter place can be filled in 24 ways.And the digits can be respectively filled in 10, 9, and 8 ways.

Hence; number of license plates in which all the letters and digits are distinct are

$$26 \times 25 \times 24 \times 10 \times 9 \times 8 = 11, 232, 000$$

(e) How many license plates could begin with AB and have all three letters and digits distinct.

SOLUTION:



The first two letters places are fixed (to be filled with A and B), so there is only one way to fill them. The third letter place should contain a letter different from A & B, so there are 24 ways to fill it.
The three digit positions can be filled in 10 and 8 ways to have distinct digits.
Hence, desired number of license plates are

$$1 \times 1 \times 24 \times 10 \times 9 \times 8 = 17280$$

EXERCISE:

A variable name in a programming language must be either a letter or a letter followed by a digit. How many different variable names are possible?

SOLUTION:

First consider variable names one character in length. Since such names consist of a single letter, there are 26 variable names of length 1.
Next, consider variable names two characters in length. Since the first character is a letter, there are 26 ways to choose it. The second character is a digit, there are 10 ways to choose it. Hence, to construct variable name of two characters in length, there are

$26\times10 = 260$ ways.

Finally, by sum rule, there are $26 + 260 = 286$ possible variable names in the programming language.

EXERCISE:

(a) How many bit strings consist of from one through four digits?
(b)How many bit strings consist of from five through eight digits?

SOLUTION:

(a) Number of bit strings consisting of 1 digit = 2
Number of bit strings consisting of 2 digits = $2\cdot2 = 2^2$
Number of bit strings consisting of 3 digits = $2\cdot2\cdot2 = 2^3$
Number of bit strings consisting of 4 digits = $2\cdot2\cdot2\cdot2 = 2^4$
Hence by sum rule, the total number of bit strings consisting of one through four digit is

$$2+2^2+2^3+2^4 = 2 + 4 + 8 + 16 = 30$$

(b) Number of bit strings of 5 digits = 2^5
Number of bit strings of 6 digits = 2^6
Number of bit strings of 7 digits = 2^7
Number of bit strings of 8 digits = 2^8

Hence, by sum rule, the total number of bit strings consisting of five through eight digit is

$2^5 + 2^6 + 2^7 + 2^8 = 480$

EXERCISE:

How many three-digit integers are divisible by 5?

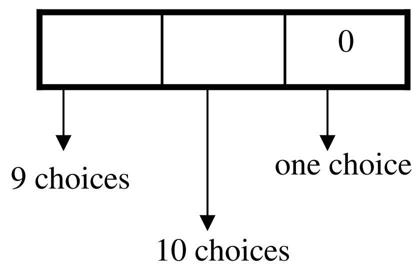
SOLUTION:

Integers that are divisible by 5, end either in 5 or in 0.

CASE-I (Integers that end in 0)

There are nine choices for the left-most digit (the digits 1 through 9) and ten choices for the middle digit.(the digits 0 through 9) Hence, total number of 3 digit integers that end in 0 is

$$9 \times 10 \times 1 = 90$$



CASE-II (Integer that end in 5)

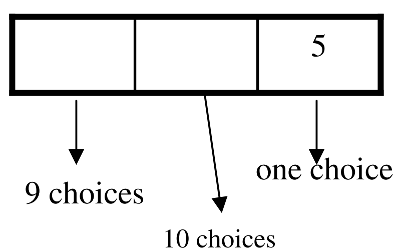
There are nine choices for the left-most digit and ten choices for the middle digit

Hence, total number of 3 digit integers that end in 5 is

$9 \times 10 \times 1 = 90$

Finally, by sum rule, the number of 3 digit integers that are divisible by 5 is

$$90 + 90 = 180$$



EXERCISE:

A computer access code word consists of from one to three letters of English alphabets with repetitions allowed.

How many different code words are possible.

SOLUTION:

Number of code words of length 1 = 26^1

Number of code words of length 2 = 26^2

Number of code words of length 3 = 26^3

Hence, the total number of code words = $26^1 + 26^2 + 26^3$

= 18,278

NUMBER OF ITERATIONS OF A NESTED LOOP:

Determine how many times the inner loop will be iterated when the following algorithm is implemented and run

```
        for    i: = 1 to 4
for     j : = 1 to 3
        [Statement in body of inner loop.
                                None contain branching statements
         that lead out of the inner loop.]
  next j
  next i
```

SOLUTION:

The outer loop is iterated four times, and during each iteration of the outer loop, there are three iterations of the inner loop. Hence, by product rules the total number of iterations of inner loop is $4 \cdot 3 = 12$

EXERCISE:

Determine how many times the inner loop will be iterated when the following algorithm is implemented and run.

```
        for    i = 5 to 50
for     j: = 10 to 20
       [Statement in body of inner loop.
                                None contain branching statements
        that lead out of the inner loop.]
  next j
  next i
```

SOLUTION:

The outer loop is iterated 50 - 5 + 1 = 46 times and during each iteration of the outer loop there are 20 - 10 + 1 = 11 iterations of the inner loop. Hence by product rule, the total number of iterations of the inner loop is 46.11 = 506

EXERCISE:

Determine how many times the inner loop will be iterated when the following algorithm is implemented and run.

```
              for    i: = 1 to 4
for j: = 1 to i
        [Statements in body of inner loop.
                                None contain branching statements
        that lead outside the loop.]
  next j
  next i
```

SOLUTION:

The outer loop is iterated 4 times, but during each iteration of the outer loop, the inner loop iterates different number of times.

For first iteration of outer loop, inner loop iterates 1 times.
For second iteration of outer loop, inner loop iterates 2 times.
For third iteration of outer loop, inner loop iterates 3 times.
For fourth iteration of outer loop, inner loop iterates 4 times.
Hence, total number of iterations of inner loop = 1 + 2 + 3 + 4 = 10

LECTURE 30

FACTORIAL
K-SAMPLE
K-PERMUTATION

FACTORIAL OF A POSITIVE INTEGER:

For each positive integer n, its factorial is defined to be the product of all the integers from 1 to n and is denoted n!. Thus $n! = n(n - 1)(n - 2) \ldots 3 \cdot 2 \cdot 1$

In addition, we define

$$0! = 1$$

REMARK:

n! can be recursively defined as

Base: $0! = 1$

Recursion $n! = n(n - 1)!$ for each positive integer n.

Compute each of the following

(i) $\dfrac{7!}{5!}$ (ii) $(-2)!$

(iii) $\dfrac{(n+1)!}{n!}$ (iv) $\dfrac{(n-1)!}{(n+1)!}$

SOLUTION:

(i) $\dfrac{7!}{5!} = \dfrac{7 \cdot 6 \cdot 5!}{5!} = 7 \cdot 6 = 42$

(ii) $(-2)!$ is not defined

(iii) $\dfrac{(n+1)!}{n!} = \dfrac{(n+1)n!}{n!} = n + 1$

(iv) $\dfrac{(n-1)!}{(n+1)!} = \dfrac{(n-1)!}{(n+1) \cdot n \cdot (n-1)!} = \dfrac{1}{(n+1)n} = \dfrac{1}{n^2 + n}$

EXERCISE:

Write in terms of factorials.

(i) $25 \cdot 24 \cdot 23 \cdot 22$ (ii) $n(n-1)(n-2) \ldots (n - r + 1)$

(iii) $\dfrac{n(n-1)(n-2)\cdots(n-r+1)}{1 \cdot 2 \cdot 3 \cdots (r-1) \cdot r}$

SOLUTION:

(i) $25 \cdot 24 \cdot 23 \cdot 22 = \dfrac{25 \cdot 24 \cdot 23 \cdot 22 \cdot 21!}{21!} = \dfrac{25!}{21!}$

(ii) $n(n-1)(n-2)\cdots(n-r+1) = \dfrac{n(n-1)(n-2)\cdots(n-r+1)(n-r)!}{(n-r)!}$

$$= \frac{n!}{(n-r)!}$$

$$(iii)\quad \frac{n(n-1)(n-2)\cdots(n-r+1)}{1\cdot 2\cdot 3\cdots(r-1)\cdot r} = \frac{n(n-1)(n-2)\cdots(n-r+1)}{r!}$$
$$= \frac{n(n-1)(n-2)\cdots(n-r+1)(n-r)!}{r!(n-r)!}$$
$$= \frac{n!}{r!(n-r)!}$$

COUNTING FORMULAS:
From a given set of n distinct elements, one can choose k elements in different ways. The number of selections of elements varies according as:
(i)elements may or may not be repeated.
(ii)the order of elements may or may not matter.
These two conditions therefore lead us to four counting methods summarized in the following table.

	ORDER MATTERS	ORDER DOES NOT MATTER
REPETITION ALLOWED	k-sample	k-selection
REPETITION NOT ALLOWED	k-permutation	k-combination

K-SAMPLE

A k-sample of a set of n elements is a choice of k elements taken from the set of n elements such that the order of elements matters and elements can be repeated.
REMARK:
With k-sample, repetition of elements is allowed, therefore, k need not be less than or equal to n. i.e. k is independent of n.
FORMULA FOR K-SAMPLE:
Suppose there are n distinct elements and we draw a k-sample from it.The first element of the k-sample can be drawn in n ways. Since, repetition of elements is allowed, so the second element can also be drawn in n ways.
Similarly each of third, fourth, …, k-th element can be drawn in n ways.
Hence, by product rule, the total number of ways in which a k-sample can be drawn from n distinct elements is

$$n \cdot n \cdot n \cdot \ldots \cdot n \qquad \text{(k-times)}$$
$$= n^k$$

EXERCISE:

How many possible outcomes are there when a fair coin is tossed three times.

SOLUTION:

Each time a coin is tossed it's outcome is either a head (H) or a tail (T). Hence in successive tosses, H and T are repeated. Also the order in which they appear is important. Accordingly, the problem is of 3-samples from a set of two elements H and T. [k = 3, n = 2]

Hence number of samples $= n^k$

$= 2^3 = 8$

These 8-samples may be listed as:

HHH, HHT, HTH, THH, HTT, THT, TTH, TTT

EXERCISE:

Suppose repetition of digits is permitted.

(a) How many three-digit numbers can be formed from the six digits 2, 3, 4, 5, 7 and 9

SOLUTION:

Given distinct elements = n = 6

Digits to be chosen = k = 3

While forming numbers, order of digits is important. Also digits may be repeated. Hence, this is the case of 3-sample from 6 elements.

Number of 3-digit numbers $= n^k = 6^3 = 216$

(b) How many of these numbers are less than 400?

SOLUTION:

From the given six digits 2, 3, 4, 5, 7 and 9, a three-digit number would be less than 400 if and only if its first digit is either 2 or 3.

The next two digits positions may be filled with any one of the six digits.

Hence, by product rule, there are

$$2 \cdot 6 \cdot 6 = 72$$

three-digit numbers less than 400.

(c) How many are even?

SOLUTION:

A number is even if its right most digit is even. Thus, a 3-digit number formed by the digits 2, 3, 4, 5, 7 and 9 is even if it last digit is 2 or 4. Thus the last digit position may be filled in 2 ways only while each of the first two positions may be filled in 6 ways.

Hence, there are

$$6 \cdot 6 \cdot 2 = 72$$

3-digit even numbers.

(d) How many are odd?

SOLUTION:

A number is odd if its right most digit is odd. Thus, a 3-digit number formed by the digits 2, 3, 4, 5, 7 and 9 is odd if its last digit is one of 3, 5, 7, 9. Thus, the last digit

position may be filled in 4 ways, while each of the first two positions may be filled in 6 ways.
Hence, there are $6 \cdot 6 \cdot 4 = 144$
3-digit odd numbers.

(e) How many are multiples of 5?

SOLUTION:

A number is a multiple of 5 if its right most digit is either 0 or 5. Thus, a 3-digit number formed by the digits 2, 3, 4, 5, 7 and 9 is multiple of 5 if its last digit is 5. Thus, the last digit position may be filled in only one way, while each of the first two positions may be filled in 6 ways.
Hence, there are $6 \cdot 6 \cdot 1 = 36$
3-digit numbers that are multiple of 5.

EXERCISE:

A box contains 10 different colored light bulbs. Find the number of ordered samples of size 3 with replacement.

SOLUTION:

Number of light bulbs = n = 10
Bulbs to be drawn = k = 3

Since bulbs are drawn with replacement, so repetition is allowed. Also while drawing a sample, order of elements in the sample is important.

Hence number of samples of size 3 $= n^k$
$= 10^3$
$= 1000$

EXERCISE:

A multiple choice test contains 10 questions; there are 4 possible answers for each question.

(a) How many ways can a student answer the questions on the test if every question is answered?

(b) How many ways can a student answer the questions on the test if the student can leave answers blank?

SOLUTION:

(a) Each question can be answered in 4 ways. Suppose answers are labeled as A, B, C, D. Since label A may be used as the answer of more than one question. So repetition is allowed. Also the order in which A, B, C, D are choosed as answers for 10 questions is important. Hence, this is the one of k-sample, in which

n = no. of distinct labels = 4
k = no. of labels selected for answering = 10

$\therefore$ No. of ways to answer 10 questions $= n^k$
$= 4^{10}$
$= 1048576$

(b) If the student can leave answers blank, then in addition to the four answers, a fifth option to leave answer blank is possible. Hence, in such case

n = 5
and k = 10 (as before)
∴ No. of possible answers= n^k
$= 5^{10}$
= 9765625

k-PERMUTATION:
A k-permutation of a set of n elements is a selection of k elements taken from the set of n elements such that the order of elements matters but repetition of the elements is not allowed. The number of k-permutations of a set of n elements is denoted P(n, k) or ${}^{n}p_{k}$.

REMARK:
1. With k-permutation, repetition of elements is not allowed, therefore $k \le n$.
2. The wording "number of permutations of a set with n elements" means that all n elements are to be permuted, so that k = n.

FORMULA FOR k-PERMUTATION:
Suppose a set of n elements is given. Formation of a k-permutation means that we have an ordered selection of k elements out of n, where elements cannot be repeated.
1st element can be selected in n ways
2nd element can be selected in (n-1) ways
3rd element can be selected in (n-2) ways
…………………………………
kth element can be selected in (n-(k-1)) ways
Hence, by product rule, the number of ways to form a k-permutation is

$$\begin{aligned} P(n,k) &= n\cdot(n-1)\cdot(n-2)\cdots(n-(k-1)) \\ &= n\cdot(n-1)\cdot(n-2)\cdots(n-k+1) \\ &= \frac{[n\cdot(n-1)\cdot(n-2)\cdots(n-k+1)][(n-k)(n-k-1)\cdots 3\cdot 2\cdot 1]}{[(n-k)(n-k-1)\cdots 3\cdot 2\cdot 1]} \\ &= \frac{n!}{(n-k)!} \end{aligned}$$

EXERCISE:
How many 2-permutation are there of {W, X, Y, Z}? Write them all.

SOLUTION:
Number of 2-permutation of 4 elements is

$$\begin{aligned} P(4,2) = {}^{4}p_{2} &= \frac{4!}{(4-2)!} \\ &= \frac{4\cdot 3\cdot 2!}{2!} \\ &= 4\cdot 3 = 12 \end{aligned}$$

These 12 permutations are:
WX, WY, WZ,

XW, XY, XZ,
YW, YX, YZ,
ZW, ZX, ZY.

EXERCISE:

Find (a) P(8, 3) (b) P(8,8)
(c) P(8,1) (d) P(6,8)

SOLUTION:

$$(a)\quad P(8,3) = \frac{8!}{(8-3)!} = 8\cdot 7\cdot 6 = 336$$

$$(b)\quad P(8,8) = \frac{8!}{(8-8)!} = \frac{8!}{0!} = 8! = 40320 \qquad (as\ \ 0! = 1)$$

$$(c)\quad P(8,1) = \frac{8!}{(8-1)!} = \frac{8\cdot 7!}{7!} = 8$$

(d) $P(6,8)$ is not defined, since the second integer cannot exceed the first integer.

EXERCISE:

Find n if

(a) P(n,2) = 72 **(b)** P(n,4) = 42 P (n, 2)

SOLUTION:

(a) Given P(n,2) = 72

$\Rightarrow$ $n \cdot (n-1) = 72$ (by using the definition of permutation)

$\Rightarrow$ $n^2 - n = 72$

$\Rightarrow$ $n^2 - n - 72 = 0$

$\Rightarrow$ $n = 9, -8$

Since n must be positive, so the only acceptable value of n is 9.

(b) Given P(n,4) = 42P(n,2)

$\Rightarrow$ $n (n-1) (n-2) (n-3) = 42\, n (n - 1)$ (by using the definition of permutation)

$\Rightarrow$ $(n-2) (n-3) = 42$ if $n \neq 0, n \neq 1$

$\Rightarrow$ $n^2 - 5n + 6 = 42$

or $n^2 - 5n - 36 = 0$

or $(n-9) (n+4) = 0$

$\Rightarrow$ $n = 9, -4$

Since n must be positive, the only answer is n = 9

EXERCISE:

Prove that for all integers n ≥ 3

P (n + 1, 3) - P(n, 3) = 3 P (n, 2)

SOLUTION:

Suppose n is an integer greater than or equal to 3

Now L.H.S = P (n + 1, 3) - P(n, 3)
= (n + 1) (n) (n-1) - n (n - 1)(n - 2)
= n (n - 1) [(n + 1) - (n - 2)]

$= n (n - 1) [n + 1 - n + 2]$
$= 3 n (n - 1)$
R.H.S $= 3P (n, 2)$
$= 3 \cdot n(n-1)$

Thus L.H.S = R.H.S. Hence the result.

EXERCISE:

(a) How many ways can five of the letters of the word ALGORITHM be selected and written in a row?

(b) How many ways can five of the letters of the word ALGORITHM be selected and written in a row if the first two letters must be TH?

SOLUTION:

(a) The answer equals the number of 5-permutation of a set of 9 elements and

$$P(9,5) = \frac{9!}{(9-5)!} = 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 = 15120$$

(b) Since the first two letters must be TH hence we need to choose the remaining three letters out of the left 9 - 2 = 7 alphabets.
Hence, the answer is the number of 3-permutations of a set of seven elements which is

$$P(7,3) = \frac{7!}{(7-3)!} = 7 \cdot 6 \cdot 5 = 210$$

EXERCISE:

Find the number of ways that a party of seven persons can arrange themselves in a row of seven chairs.

SOLUTION:

The seven persons can arrange themselves in a row in P(7,7) ways.
Now

$$P(7,7) = \frac{7!}{(7-7)!} = \frac{7!}{0!} = 7!$$

EXERCISE:

A debating team consists of three boys and two girls. Find the number n of ways they can sit in a row if the boys and girls are each to sit together.

SOLUTION:

There are two ways to distribute them according to sex: BBBGG or GGBBB. In each case
the boys can sit in a row in P(3,3) = 3! = 6 ways, and
the girls can sit in
P(2,2) = 2! = 2 ways and

Every row consist of boy and girl which is = 2!=2
Thus

The total number of ways=n = $2 \cdot 3! \cdot 2!$
$= 2 \cdot 6 \cdot 2 = 24$

EXERCISE:

Find the number n of ways that five large books, four medium sized book, and three small books can be placed on a shelf so that all books of the same size are together.

SOLUTION:

In each case, the large books can be arranged among themselves in P(5,5)= 5! ways, the medium sized books in P(4,4) = 4! ways, and the small books in P(3,3) = 3! ways.

The three blocks of books can be arranged on the shelf in P(3,3) = 3! ways.

Thus

$n = 3! \cdot 5! \cdot 4! \cdot 3!$
$= 103680$