6·1 : THE BASICS OF COUNTING

Combinatorics, the study of arrangements of objects, is an important part of discrete mathematics.

COMBINATORICS is the branch of mathematics studying the

- Enumeration
- Combination
- permutation and
- Mathematical relations that characterize their properties.

ENUMERATION; the Counting of objects with certain propeties, is an important part of combinatorics.

Counting Problems arise throughout mathematics and Computer Science:

- Counting is used to determine the complexity of algorithms.
  We need to Count the number of operations used by algorithm to study its time Complexity.
- Counting is also required to determine whether there are enough telephone numbers or internet protocols addresses to meet demand.
- To determine probabilities of discrete events, we must Count the successful outcomes of experiments and all the possible outcomes of these experiments.
- Counting has played a key role in mathematical biology, especially in sequencing DNA.

BASIC COUNTING PRINCIPLES

1. THE PRODUCT RULE
2. THE SUM RULE

THE PRODUCT RULE:

The product rule applies when a procedure is made up of separate tasks.

Suppose that a procedure can be broken down into a sequence of two tasks. If there are n_1 ways to do the first task and for each of these ways of doing the first task, there are n_2 ways to do the second task, then there are $n_1 n_2$ ways to do the procedure.

Example 1:

A new company with two employees, Sanchez and Patel, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?

Sol:

ways of assigning an office to Sanchez : 12

then ways of assigning an office to Patel difference from Sanchez : 11

By the product Rule:

The ways to assign offices to these two employees : $12 \times 11 = 132$

Example 2:

The chairs of an auditorium are to be labeled with an uppercase English letter followed by a positive integer not exceeding 100. What is the largest number of chairs that can be labelled differently?

Sol:

letter integer

_ _

A: A1, A2, ... A100

B: B1, B2, ... B100

⋮

Z: Z1, Z2, ... Z100

e.g.,

B76

D99

H1

P29

Q3 etc.

The procedure has two tasks:

n_1 : Assigning one of the uppercase letter $= 26$

n_2 : For each uppercase letter, assigning to it one of the 100 possible integers $= 100$

By Product rule:

The largest number of chairs that can be labelled $= 26 \times 100 = 2600$ (# of chairs)

Example 3 :

There are 32 Computers in a data Center in the Cloud. Each of these Computers has 24 ports. How many different Computer ports are there ?

Sol :-

# of Computers = 32

# of ports on each Computer = 24

total # of ports in data Center = 32 × 24 = 768 ports.

Example 4 :

How many different bit strings of length seven are there?

Sol :-

— . — . — . — . — . — . —

Each bit Canbe Chosen in two ways (0 or 1).

There are Separate 7 tasks, $n_1, n_2, \cdots n_7$.

Task n_1 Contains 0 or 1, So there are 2 different bit strings.

Task n_2 Contains 0 or 1, So there are 2 diffe—t bit strin—

≡

Task n_7 " " " , " " " " " " "

By the product rule,

Total # of different bit strings : $n_1 \times n_2 \times \cdots n_7$

$$= 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$$

$$= 2^7 = 128$$

Example 5 :

How many different License plates Canbe made if each plate Contains a Sequence of three Uppercase English letters followed by three digits (and no Sequences of letters are prohibited, even if they are obscene) ?

Sol :

Letters Digits

— — — — — —

26 26 26 10 10 10

- Sequence of 3 uppercase letter each has 26 different forms.
- Sequence of 3 digits each has 10 different forms :

By Product rule :

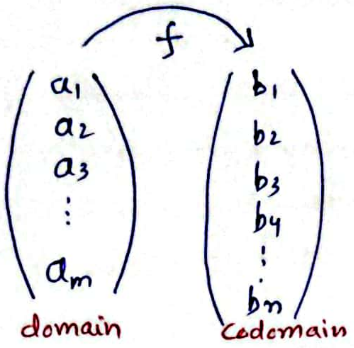
$$26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 = 17,576,000$$

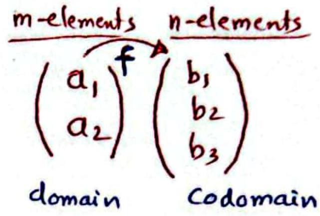
Possible license plates.

EXAMPLE 6: (Counting Functions : Product Rule)

How many functions are there from a set with m elements to a set with n elements?

Sol:-





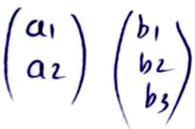
A function Corresponds to a choice of one of the n elements in Codomain for each ~~element~~ of the m elements in the domain.

By product rule, there are

$$\overset{1}{n} \cdot \overset{2}{n} \cdot \overset{3}{n} \cdot \cdots \cdot \overset{m}{n} = n^m$$

functions from a set with m elements to ~~a set~~ one ~~of~~ with n elements.

For example



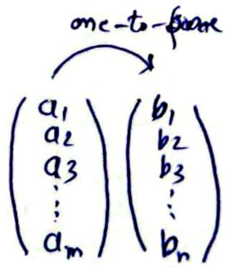
possible functions $= 3 \cdot 3 = 3^2 = 9$ functions (from a set with 2 elements to a set with 3 elements)

EXAMPLE 7 : (Counting one-to-one functions)

How many functions (one-to-one) are there from a set with m elements to one with n elements.

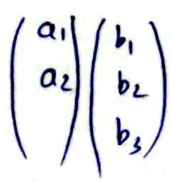
Sol: Case 1 : $m > n$, there are no one-to-one function

Case 2 : $m \leq n$



Suppose the elements in the domain are $a_1, a_2, \ldots a_m$

- there are n ways to choose a_1 value
- There are $(n-1)$ ways to choose a_2 (As the function is one-to-one, the value for a_1 cannot be used again)
- Similarly $(n-2)$ ways for a_3, $(n-3)$ ways for a_4 etc.



By Product rule, there are

$$n(n-1)(n-2) \cdot \cdots \cdot (n-m+1)$$

one-to-one functions from a set with m elements to one with n elements.

For example: for $m=3$ $n=5$, $5 \cdot 4 \cdot 3 = 60$ ~~one-to-f~~ one-to-one functions.

CamScanner

EXAMPLE 8 : The Telephone Numbering Plan

The format of telephone numbers in North America is Specified by

$$\text{10 digit number} = \underset{\text{(Area Code)}}{\overset{\text{(3-digit)}}{- - -}}\ \underset{\text{(office Code)}}{\overset{\text{(3-digit)}}{- - -}}\ \underset{\text{(Station Code)}}{\overset{\text{(4-digit)}}{- - - -}}$$

Because of Signaling Considerations, there are Certain restrictions on Some of these digits.

To specify allowable format, Let

X denote digit values : 0 through 9

N denote digit values : 2 through 9

Y denote digit values : 0 or 1

Two Numbering plans :

Old Plan format : $\underset{\text{area Code}}{NYX} - \underset{\text{off. Code}}{NNX} - \underset{\text{Station Code}}{XXXX}$

New Plan format : NXX - NXX - XXXX

old Plan (in use in 1960s) has been replaced by the new plan, but recent rapid growth of mobile phones make this new plan obsolete.

We want to know, how many different telephone numbers ~~was~~ were possible under the OLD and NEW PLAN?

Sol:

By product rule : OLD PLAN

Total # of Telephones : $\underset{8\ 2\ 10}{NYX} - \underset{8\ 8\ 10}{NNX} - \underset{10\ 10\ 10\ 10}{XXXX}$

$$= \underline{8*2*10} * 8\times8\times10 * 10*10*10*10$$

$$= 160 * 640 * 10000$$

$= 1,024,000,000$ different numbers available

NEW PLAN

Total Phone #s : NXX - NXX - XXXX

$$= 8\times10\times10 * 8\times10\times10 * 10\times10\times10\times10$$

$$= 800 * 800 * 10000 =$$

$= 6,400,000,000$ different #s available.

Example 9 : (Counting Subsets of a Finite Set)

How many different subsets of a finite Set S?

Sol :

Let S be a finite set. List the elements in arbitrary order.

As there is one-to-one Correspondence between Subsets of S and bit strings of length $|S|$.

Let

$S = \{a_1, a_2, a_3\}$

All subsets of $S = \{\phi, \{a_1\}, \{a_2\}, \{a_3\}, \{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_3\}\}$

bit strings $= \{000, 001, 010, 100, 110, 101, 011, 111\}$ one-to-one Correspondence

$\#$ of bits $= 3$

$$\underset{2}{\underline{(0,1)}}\ \underset{2}{\underline{(0,1)}}\ \underset{2}{\underline{(0,1)}}$$

$= 2 \times 2 \times 2 = 8$ Subsets

i.e Total different Subsets $= 2^{|S|} = 2^3 = 8$

xxx

SUM RULE

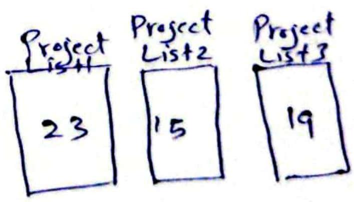
EXAMPLE 1 : (Sum Rule)

A Student Can Choose a Computer project from one of three lists. The three lists Contain 23, 15 and 19 possible projects, respectively. No project is on more than one list.

How many possible projects are there to Choose from?

Sol:-

By Sum rule :



Total $\#$ of projects to Choose $= 23 + 15 + 19$

$= 57$ ways to Choose a project.

$$|A_1 \cup A_2 \cup A_3 \cdots \cup A_m| = |A_1| + |A_2| + \cdots + |A_m|$$

MORE COMPLEX COUNTING PROBLEMS:

EXAMPLE 1:

In a version of the Computer Language BASIC, the name of a variable is a string of one or two alphanumeric characters, where uppercase and lowercase are not distinguished. (An alphanumeric Character is either one of the 26 English letters or one of the 10 digits.)
Moreover, a variable name must begin with a letter and must be different from the five strings of two Characters that are reserved for programing use. How many different variable names are there in this version of BASIC?

Sol:

V_1: __

letter (one of the 26 letters)

V_2: __ __

letter (one of the 26 letters) alphanumeric character (one of the 36 letters 26 + 10 = 36)

String of one or two letters

① — variable with one character long

② — — variable with 2 characters long

variable name must start with character

variable name must start with letter (Contains one or two letters).

$V_1 = 26$

$V_2 = 26 * 36 = 936$ (Product rule)

one or two characters long, So By Sum rule

$V_1 + V_2 = 26 + 936 = 962$

5 strings are reserved, So total variable names will be

$= 962 - 5 = 957$

EXAMPLE 2 : (Complex Counting Problem)

Each user on a computer system has passwords, which is six to eight characters long, where each character is an uppercase letter or digit. Each password must contain at least one digit. How many possible passwords are there ?

Sol :

Passwords = six to eight characters long

P_6 = Six letters long passwords

P_7 = Seven letters long passwords

P_8 = Eight letters long passwords

Total number of passwords :

$$P = P_6 + P_7 + P_8 \quad \text{(By Sum rule)}$$

P_6 : Finding P_6 directly is difficult : (6 length containing letter or digit with at least one digit.)

Easier to find :

- # of strings of uppercase letters and digits that are six characters long

$$\underline{36}\ \underline{36}\ \underline{36}\ \underline{36}\ \underline{36}\ \underline{36}$$

(Letter or digit) one of (26 + 10)

# of password $= 36 \times 36 \times 36 \times 36 \times 36 \times 36 = 2176782336$ (Product rule)

- Each password must contain atleast one digit i.e no password having only letters.

(Product rule)

# of string with no digit $= 26 \times 26 \times 26 \times 26 \times 26 \times 26 = 308915776$

So,

# of strings of uppercase letters and digits of 6 length having atleast 1 digit :

$$= 2,176,782,336 - 308,915,776$$
$$= 1,867,866,560$$

Similarly,

$$P_7 = 36^7 - 26^7 = 78,364,164,096 - 8,031,810,176 = 70,332,353,920$$
$$P_8 = 36^8 - 26^8 = 2,821,109,907,456 - 208,827,064,576 = 2,612,282,842,880$$

And $\quad P = P_6 + P_7 + P_8 = 2,684,483,063,360$

EXAMPLE 3: (Counting Internet Addresses)

How many different IPV4 addresses are available for computers on internet?

In version 4 of the internet protocol (IPV4), an addres of 32 bits begin with a network number (netid) followed by a host number (hostid) which identifies a computer as a member of particular network.

Three forms of addresses are used:

Class A: (used for the largest networks):
Consist of 0 followed by 7-bit netid, and 24-bit hostid

Class B: (used for medium-sized networks):
Consist of 10 followed by 14-bit netid and 16-bit hostid

Class C: (used for smallest networks):
Consist of 110 followed by 21-bit netid and 8-bit hostid

Restrictions:

1111111 is not available as the netid of a Class A network

hostids consisting of all 0's and all 1's are not available in any network

Sol: x_A = # of Class A addresses
x_B = " " Class B " "
x_C = " " Class C " "

$x = x_A + x_B + x_C$ (By Sum Rule)

$$x_A = \underset{\text{netid}}{(2^7 - 1)} \cdot \underset{\text{hostid}}{(2^{24} - 2)}$$

Not Available
- one netid
- two hostids

$= 127 * 16,777,214 = 2,130,706,178$

$$x_B = 2^{14} * (2^{16} - 2)$$

$= 16,384 * 65,534 = 1,073,709,056$

$$x_C = 2^{21} \cdot (2^8 - 2)$$

$= 2,097,152 * 254$

$= 532,676,608$

Class	0	1 2 3 4 5 6 7	8	16	24	31
A	0	netid	hostid			
B	1	0	$^{2\,3}$ netid ^{15}	^{16} hostid ^{31}		
C	1	1	0	^{3} netid ^{23}	^{24} hostid ^{31}	
D	1	1	1	0	^{4} multicast ^{31}	
E	1	1	1	1	0	^{5} Address ^{31}

Class D:
Reserved for use in multicasting

Class E:
Reserved for future use

$x = x_A + x_B + x_C = 2,130,706,178 + 1,073,709,056 + 532,676,608$

$= 3,737,091,842$

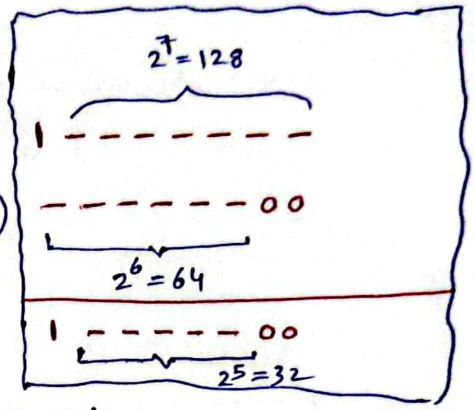
Total number of available addresses for computer on the internet

THE INCLUSION-EXCLUSION PRINCIPLE (Subtraction Principle):

Example 1 : (Subtraction Principle)

How many bit strings of length eight either start with 1 or end with two bits 00.

Sol:-



\# of bit strings of length 8 start with 1 : either (0 or 1)

$$2^7 = 128 \quad (\text{Product rule: } \overline{2}\times\overline{2}\times\overline{2}\times\overline{2}\times\overline{2}\times\overline{2}\times\overline{2})$$

\# of bit strings of length 8 end with 00 :

$$2^6 = 64 \quad (\text{Product rule: } \overline{2}\times\overline{2}\times\overline{2}\times\overline{2}\times\overline{2}\times\overline{2})$$

Some of the strings to construct a bit string of length 8 starting with 1 are the same as the strings to construct a bit string that end with 00.

There are

$$2^5 = 32 \text{ such strings.}$$

So, the number of bit strings of length 8 either start with 1 or end with 00 :

$$= 128 + 64 - 32$$

$$= 160$$

NOTE : Inclusion-Exclusion Principle

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

EXAMPLE : (Subtraction Principle)

A Computer Company receives 350 applications from Computer graduates for a job planing a line of new web servers. Suppose that 220 of these people majored in Computer Science, 147 majored in Business, and 51 majored both in Computer Science and in business. How many of these applications majored neither in CS nor in business.

Sol :

Total # of applications = 350

A_1 = Computer Science majored Applicants = 220
A_2 = Business majored Applicants = 147
Both, Computer Science, business majored = 51

# of Students who majored either in Computer Sc. or in business (or both):

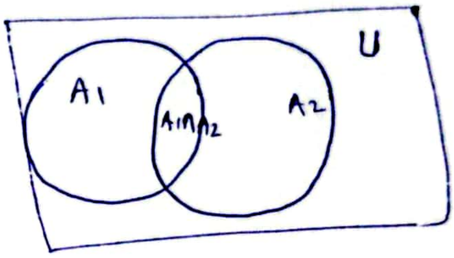
$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$$

$$= 220 + 147 - 51 = 316$$

# of applicants who majored neither in Computer Sc. nor in business =

$$= 350 - 316$$

$$= 34$$



$$= |U| - |A_1 \cup A_2|$$

$$= 350 - (|A_1| + |A_2| - |A_1 \cap A_2|)$$

$$= 350 - (220 + 147 - 51)$$

$$= 350 - 316$$