The Basics of Counting

Niloufar Shafiei

Counting applications

- Counting has many applications in computer science and mathematics.

For example,

- Counting the number of operations used by an algorithm to study its time complexity
- Counting the successful outcomes of experiments
- Counting all the possible outcomes of experiments
- …

Basic counting principles

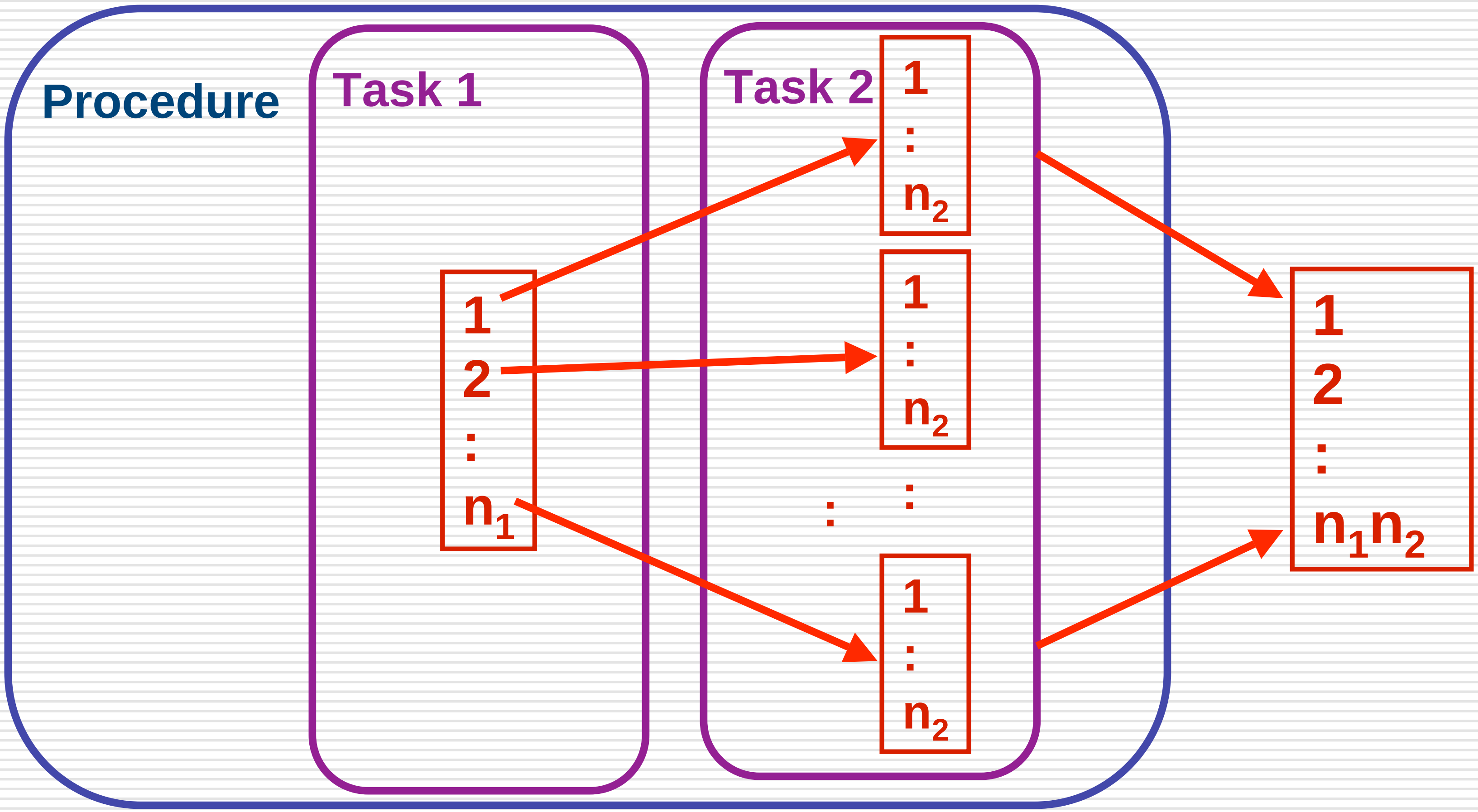
- Two basic counting principles
  1. The product rule
     - The product rule applies when a procedure is made up of separate tasks.
  2. The sum rule

The product rule

The product rule:

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

The product rule



Example

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

Solution:

- Break the procedure into tasks
  - Task 1: assigning an office to employee 1
  - Task 2: assigning an office to employee 2
- Count different ways of doing each task and then use the product rule
  - Task 1 can be done in 12 different ways and for each of these ways, Task 2 can be done in 11 different ways.
  - By product rule, There are 12 (11) = 132 ways to assign offices to two employees.

Example

The chairs of an auditorium are to be labeled with a letter and a positive integer not exceeding 100. How many chairs can be labeled differently?

Solution:

- Break the procedure into tasks
  - Task 1: assigning one of the 26 letters
  - Task 2: assigning one of the 100 possible integers
- Count different ways of doing each task and then use the product rule
  - Task 1 can be done in 26 different ways and for each of these ways, Task 2 can be done in 100 different ways.
  - By product rule, There are 26(100) = 2600 ways to assign labels to the chairs.

Extended version of the product rule

Extended version of the product rule:

- A procedure can be broken down into a sequence of tasks $T_1, T_2, \ldots, T_m$.
- Assume each task T_i $(i=1,2,\ldots,m)$, can be done in n_i different ways, regardless of how the previous tasks were done.
- The procedure can be done in $n_1 n_2 \ldots n_m$ different ways.

(prove it using induction as exercise.)

Example

How many different bit strings of length seven are there?

Solution:

- Break the procedure into tasks
  - Task 1: assigning bit 1 to 0 or 1
  - Task 2: assigning bit 2 to 0 or 1
  - …
  - Task 7: assigning bit 7 to 0 or 1
- Count different ways of doing each task and then use the product rule
  - Each task can be done in 2 different ways.
  - By product rule, There are $2^7 = 128$ different bit strings of length seven.

Example

How many different license plates are available if each plate contains a sequence of three letters followed by 3 digits?

Solution:

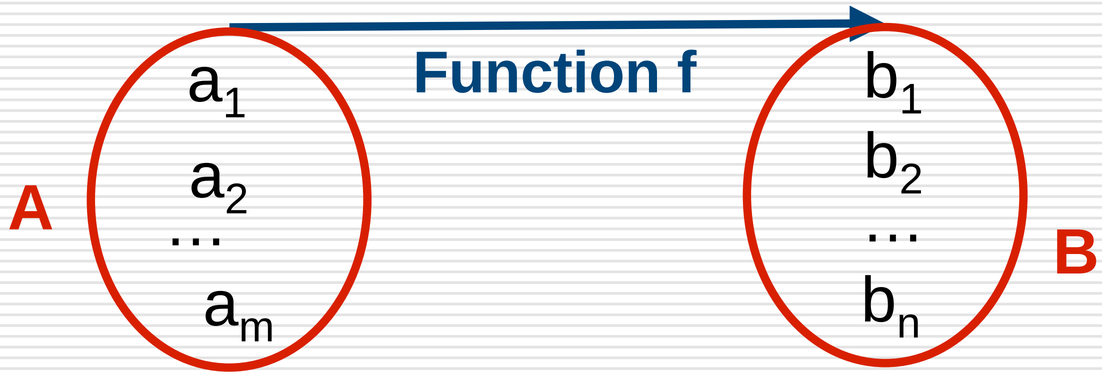
- Break the procedure into tasks
  - Task 1: choose letter 1
  - Task 2: choose letter 2
  - Task 3: choose letter 3
  - Task 4: choose digit 1
  - Task 5: choose digit 2
  - Task 6: choose digit 3
- Count different ways of doing each task and then use the product rule
  - By product rule, There are 26 . 26 . 26 .10 .10 . 10 = 17576000 different license plates.

Example

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

Solution:

- Break the procedure into tasks
  - Task i: assign an element in codomain to element a_i in domain $\forall$ i=1,2,…,m
- Count different ways of doing each task and then use the product rule
  - By product rule, There are n^m different functions.



Example

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

Solution:

- When m>n, there are no one-to-one functions.
- Let $m \leq n$.
- Break the procedure into tasks
  - Task i: assign an element in codomain to element a_i in domain $\forall$ i=1,2,…,m
- Count different ways of doing each task and then use the product rule
  - Since the function is one-to-one, so a_1 can be chosen in n different ways.
  - $f(a_1) \neq f(a_2)$, so a_2 can be chosen in n-1 different ways.
  - $f(a_2) \neq f(a_3)$ and $f(a_1) \neq f(a_3)$, so a_3 can be chosen in n-2 different ways.
  - ….
  - By product rule, There are n(n-1)(n-2)…(n-(m-1)) different one-to-one functions.

Example

The format of telephone numbers in North America is specified by a numbering plan.

- Let X denote a digit between 0 and 9.
- Let N denote a digit between 2 and 9.
- Let Y denote a digit between 0 and 1.
- In the old plan, The format of telephone numbers is NYX-NNX-XXXX.
- In the new plan, The format of telephone numbers is NXX-NXX-XXXX.

How many north American telephone numbers are possible under the old plan and under the new plan.

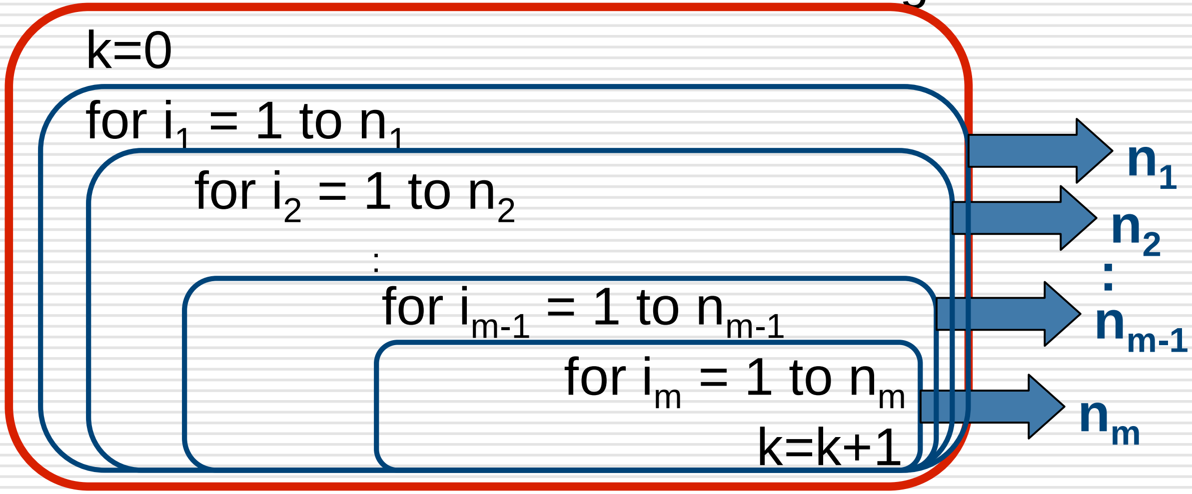
Solution:

Tel. numbers under the old plan: (8 . 2 . 10) (8 . 8 . 10) (10 .10 .10 . 10) = 160 . 640 . 10000 = 1,024,000,000

Tel. numbers under the new plan: (8 .10 . 10) (8 . 10 . 10) (10 .10 .10 . 10) = 800 . 800 . 10000 = 6,400,000,000

Example

What is the value of k after the following code has been executed?



Solution:

Task i: traversing the i-th loop. ($1 \le i \le m$)

By the product rule, the nested loops traversed $n_1 n_2 \ldots n_m$ times.

So the final value of k is $n_1 n_2 \ldots n_m$.

Example

Use the product rule to show that the number of different subsets of a finite set S is $2^{|S|}$.

Solution:

List the elements of S.

To form a subset x, we decide if x includes element a_i in S or not where $1 \leq i \leq |S|$.

Task i: decide if x includes a_i or not.

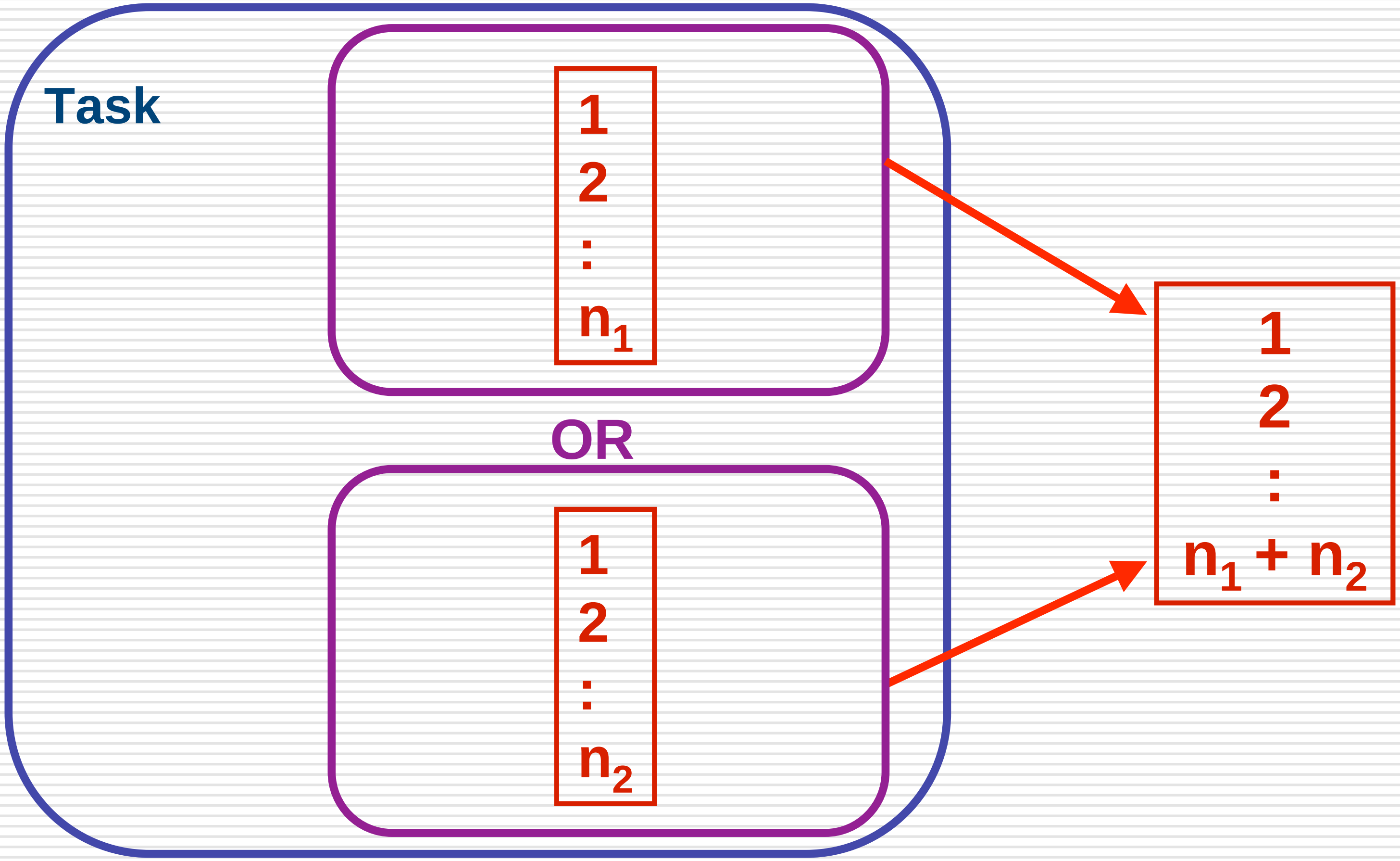
By the product rule, there are $2^{|S|}$ different subsets, in other words, $|P(S)| = 2^{|S|}$.

The sum rule

The sum rule:

- Assume a task can be done either in one of n_1 ways or in one of n_2 ways.
- Assume none of the set of n_1 ways is the same as any of the set n_2 ways.
- So, there are $n_1 + n_2$ ways to do the task.

The sum product



Example

Assume there are 37 members of the mathematics faculty and 83 mathematics majors.

Suppose that either a member of the mathematics faculty or a student who is a mathematics major is chosen as a representative to a university committee.

How many different ways to choose this representative?

Solution:

- There are 37 ways to choose a faculty member.
- There are 83 ways to choose a student.
- By the sum rule, there are 37+83=120 different ways to choose such a representative.

Extended version of the sum rule

Extended version of the sum rule:

- Suppose a task can be done in one of n_1 ways, in one of n_2 ways, …, or in one of n_m ways.
- Assume none of the set of n_i ways of doing the task is the same as any of the set of n_j ways, for all pairs I and j with $1 \leq i < j \leq m$.
- The task can be done in $n_1+n_2+\ldots+n_m$ different ways.

Example

A student can choose a computer project from one of three lists. The three lists contains 23, 15 and 19 possible projects. No project is on more than one list.

How many possible projects are there to choose from?

Solution:

- The student can choose a project from the first list, the second list or the firs list.
- By the sum rule, there are 23+15+19=57 ways to choose a project.

Example

What is the value of k after the following code has been executed?

```
k=0
for i1 = 1 to n1
    k=k+1
for i2 = 1 to n2
    k=k+1
:
for im = 1 to nm
    k=k+1
```

n_1

n_2

:

n_m

Solution:

Task i: traversing the i-th loop. ($1 \le i \le m$)

By the sum rule, the final value of k is $n_1+n_2+\ldots+n_m$.

Example

Assume $A_1, A_2, \ldots, A_m$ are m disjoint finite sets.

Use the sum rule to find the cardinality of the union of these sets.

Solution:

Each element in the set of union of these sets belong to $A_1, A_2, \ldots$ or A_m.

List the elements of the union of $A_1, A_2, \ldots, A_m$.

Task 1: a_i is a member of A_1. $(1 \le i \le |A_1|)$

Task 2: a_i is a member of A_2. $(1 \le i \le |A_2|)$

:

Task m: a_i is a member of A_m. $(1 \le i \le |A_m|)$

By the sum rule, $|A_1 \cup A_2 \cup \ldots \cup A_m| = |A_1| + |A_2| + \ldots + |A_m|$.

The product rule and the sum rule

Some complicated counting problems can be solved using both the product rule and the sum rule.

Example

In the computer language BASIC,

- The name of variable is a string of one or two alphanumeric characters (letters or digits).
- Uppercase and lowercase letters are not distinguished.
- A variable name must begin with a letter.
- A variable name must be different from the five strings of two characters that are reserved for programming use.

How many different variable names are there in this version of BASIC?

Solution:

Task 1: choose a one character long variable name.

Task 2: choose a two characters long variable name.

- Since the first character of the variable name must be a letter, task 1 can be done in 26 different ways.

Example

In the computer language BASIC,

- The name of variable is a string of one or two alphanumeric characters (letters or digits).
- Uppercase and lowercase letters are not distinguished.
- A variable name must begin with a letter.
- A variable name must be different from the five strings of two characters that are reserved for programming use.

How many different variable names are there in this version of BASIC?

Solution:

- To choose a two characters long variable name, the first character can be a letter and the second can be a letter or a digit.
- A two character long variable name must also be different from the five reserved strings.
- By the product rule, task 2 can be done in 26 . 36 - 5 = 931 different ways.
- By the sum rule, there are 931+26= 957 different variable names.

Example

In a computer system, each user has a password.

- The password must be six to eight characters long.
- Each character is an uppercase letter or a digit.
- Each password must contain at least one digit.

How many possible passwords are there?

Solution:

Task 1: choose a six characters long password.

Task 2: choose a seven characters long password.

Task 3: choose a eight characters long password.

- The number of task 1: the number of passwords of six characters where each character can be an uppercase letter or a digit is 36^6 excluding the number of six characters long passwords with no digits which is 26^6.
- Task 1 can be done in $36^6 - 26^6 = 1{,}867{,}866{,}560$ ways.

Example

How many possible passwords are there?

Solution:

- The number of task 2: the number passwords of seven characters where each character can be an uppercase letter or a digit is 36^7 excluding the number of seven characters long passwords with no digits which is 26^7.
- Task 2 can be done in $36^7 - 26^7 = 70,332,353,920$ ways.
- The number of task3: the number passwords of eight characters where each character can be an uppercase letter or a digit is 36^8 excluding the number of eight characters long passwords with no digits which is 26^8.
- Task 3 can be done in $36^8 - 26^8 = 2,612,282,842,880$ ways.
- By the sum rule, the number of possible passwords are the number of task 1 + the number of task 2 + the number of task 3 = 2,684,483,063,360.

Example

On the internet, each computer is assigned an internet address.

- An address is a string of 32 bits. It begins with a network number (netid) followed by a host number (hostid).
- There are three forms of addresses.
- Class A addresses consists of 0, followed by a 7-bit netid and a 24-bit hostid.
- Class B addresses consists of 10, followed by a 14-bit netid and a 16-bit hostid.
- Class C addresses consists of 110, followed by a 21-bit netid and a 8-bit hostid.
- There are several restrictions:
    - 1111111 is not available as the netid of a class A network.
    - Hostids do not consists of all 0s and all 1s are not available for use in any network.

How many different addresses are available for computers on the Internet.

Example

0 1 2 3 8 16 24 31

Class A	0	netid	hostid		
Class B	1	0	netid	hostid	
Class C	1	1	0	netid	hostid

Task 1: choose a Class A address.

Task 2: choose a Class B address.

Task 3: choose a Class C address.

- Since the netid of class A cannot be 1111111, the number of Class A netids is $2^7 - 1$.
- Since the hostid cannot consist all 0s and all 1s, the number of Class A hostids is $2^{24} - 2$.
- By the product rule, the number of task 1 is $(2^7 - 1)(2^{24} - 2) =$ 2,130,706,178.

Example

- By the product rule, the number of Class B netids is 2^{14}.
- Since the hostid cannot consist all 0s and all 1s, the number of Class B hostids is $2^{16} - 2$.
- By the product rule, the number of task 2 is $2^{14} (2^{16} - 2) =$ 1,073,709,056.
- By the product rule, the number of Class C netids is 2^{21}.
- Since the hostid cannot consist all 0s and all 1s, the number of Class C hostids is $2^{8} - 2$.
- By the product rule, the number of task 1 is $2^{21} (2^{8} - 2) =$ 532,676,608.
- By the sum product, the number of Internet addresses are the number of task 1 + the number of task 2 + the number of task 3 = 3,737,091,842.

The inclusion-exclusion principle

The principle of inclusion-exclusion:

- Suppose a task can be done in n_1 or in n_2 ways.
- However, some of the set of n_1 ways are the same as some of the n_2 other ways.
- To count the number of ways to the task, we add n_1 and n_2 and subtract the number of ways that is common in n_1 ways and n_2 ways.

Example

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

Solution:

Task 1: choose a bit strings of length eight starts with 1.

Task 2: choose a bit strings of length eight ends with 00.

- Since the first bit must be 1 in task 1, by the product rule, the number of task 1 is 2^7.
- Since the last two bits must be 00 in task 2, by the product rule, the number of task 2 is 2^6.
- The strings of length eight that start with 1 and end 00 are common in task 1 and task 2 and the number of them are 2^5.
- By inclusion-exclusion principles, the number of such strings is $2^7 + 2^6 - 2^5 = 160$

Example

$|A \cup B|$?

Solution:

Task 1: choose an element from A.

Task 2: choose an element from B.

- Since A has $|A|$ elements, the number of task 1 is $|A|$.
- Since B has $|B|$ elements, the number of task 2 is $|B|$.
- The elements that are in both A and B are common in task 1 and task 2 and the number of them are $|A \cap B|$.
- By inclusion-exclusion principles, the number of task 1 and task 2 (= $|A \cup B|$) is $|A|+|B|- |A \cap B|$.

Example

A computer company receives 350 applications for a job. Suppose that 220 of them majored in computer science, 147 of them majored in business and 51 of them majored both in computer science and business.

How many of these applicants majored in neither computer science and business?

Solution:

- First find the number of applicants that are majored in computer science or business, then subtract it from the total number of applicants

Task 1: choose an applicant majored in computer science

Task 2: choose an applicant majored in business.

- The number of task 1 is 220.
- The number of task 2 is 147.

Example

A computer company receives 350 applications for a job. Suppose that 220 of them majored in computer science, 147 of them majored in business and 51 of them majored both in computer science and business.

How many of these applicants majored in neither computer science and business?

Solution:

- The applicants that are majored both in computer science and business are common in task 1 and task 2 and the number of them are 51.
- By inclusion-exclusion principles, the number of task 1 and task 2 is 220+147-51=316.
- So, the number of applicants majored in neither computer science and business is 350-316=34.

Recommended exercises

3,7,11,13,15,21,31,38,39,59