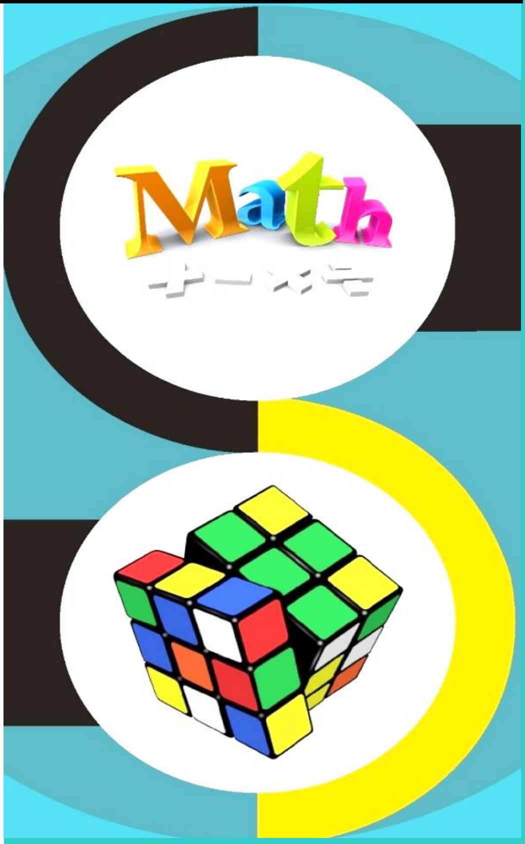
Discrete Structures



Computer Science Department/
College of Science for Women/
University of Baghdad

2023

Dr. Safa Sami Abdul-Jabbar

Why study discrete mathematics in computer science? It does not directly help us write programs. At the same time, it is the mathematics underlying almost all of computer science. Here are a few examples:

• Designing high-speed networks and message routing paths.

• Finding good algorithms for sorting.

• Performing web searches.

• Analyzing algorithms for correctness and efficiency.

• Formalizing security requirements.

• Designing cryptographic protocols.

Discrete mathematics uses various techniques, some of which are seldom found in its continuous counterpart. This course will roughly cover the following topics.

1. Set theory/ Set operations.
2. Algebra of sets.
3. set & Classes of sets.
4. Computer Representation of Sets.
5. Finite Sets and Counting Principle.
6. Mathematic induction.
7. Relations/ Pictorial representation of relations.
8. Properties of binary relations.
9. Composition of relations/ Partial ordered relation.
10. Functions.
11. Classification of functions.
12. Geometrical Characterization of One-to-One and Onto Functions.
13. Recurrence Relations.
14. Special Integer Sequences.

Set Theory

Set Theory starts very simply: it examines whether an object *belongs*, or does *not belong*, to a *set* of objects which has been described in some non-ambiguous way. From this simple beginning, an increasingly complex series of ideas can be developed, which leads to notations and techniques with many varied applications.

The present definition of a set may sound very vague. A set can be defined as an *unordered* collection of entities that are related because they obey a certain rule. 'Entities' may be anything, literally: numbers, people, shapes, cities, bits of text, ... etc

The key fact about the 'rule' they all obey is that it must be well-defined. In other words, it must describe clearly what the entities obey. If the entities we're talking about are words, for example, a well-defined rule is:

X is English

A rule which is not well-defined (and therefore couldn't be used to define a set) might be:

X is hard to spell

Where X is any word.

- Elements

An entity that belongs to a given set is called an element of that set. For example:

Henry VIII is an element of the set of Kings of England.

kings of England {Henry VIII}

- Set Notation

To list the elements of a set, we enclose them in curly brackets, separated by commas. For example:

{-3,-2,-1,0,1,2,3}

The elements of a set may also be described verbally:

{ integers between −3 and 3 inclusive}

The set builder notation may be used to describe sets that are too tedious to list explicitly. To denote any particular set, we use the letter

$\{x|x \text{ is an integer and } |x|<4\}$

or equivalently

$\{x|\ x \in Z\ ,|x|<4\}$

The symbols ∈ and ∉ denote the inclusion and exclusion of elements, respectively:

dog ∈{quadrupeds}

Washington DC ∉ {European capital cities}

Sets can contain an infinite number of elements, such as a set of prime numbers. Ellipses are used to denote the infinite continuation of a pattern:

{2,3,5,7,11,…}

Note that using ellipses may cause ambiguities, the set above may be taken as the set of integers indivisible by 4.

On the other hand, sets will usually be denoted using upper case letters: A,B, ...

Elements will usually be denoted using lower case letters: x, y, ...

In general, constructing a set is the first and foremost part of set theory. Naturally, we cannot use sets, or perform operations on sets, without having a set in the first place. We will see 3 ways to construct a set:

1. Roster method
2. Set builder notation
3. Interval notation

Roster Method:

The roster method for set construction is the simplest. We simply write down all of the members of a set between curly braces.

- A= {a,b,c,d}

The set A consists of four members: a, b, c, and d.

When there is an obvious pattern to the elements of a set, one can abbreviate the roster method using an ellipse (i.e. …).

- For example, the set of positive integers less than 100 may be written as:

{1,2,3,…,100}

- The set of all positive multiples of 3 could be written as:

{3,6,9,12,…}

Notice that the set does not need to be finite to use the roster method. Generally, sets can have a finite or infinite number of members, as we will see.

Moreover, sets do not necessarily need to contain numbers or only numbers. Sets are collections of any “thing” or “object”. Those objects could be any combination of letters, strings, colors, shapes, matrices, etc. Sets can even contain other sets!

- As an example, the following as also valid sets:

{“Alice”,”Bob”,”Ahmed”,”Sara”}

{1,2,”red”,”blue”, “fish”}

{{a,b},”red”,{“bar”,2}}

{a,b,c,d,….,z}

Using the roster method, we can describe some well-known mathematical sets as follows.

- Roster method for sets of numbers

N= {0,1,2,3,4,…}

Z= {…,-2,-1,0,1,2,….}

Z+= {1,2,3,4,…}

Set Builder:

The second, and more general way of describing a set is using set builder notation. You have likely seen this notation used implicitly in previous math courses.

- Set builder notation

Set builder notation specifies elements that should be included in a set based on a variable (respectively, some expression) and a condition on that variable (respectively, expression). The variable and condition are separated by a vertical bar |.

{x | some condition on X}

{some expression | some conditions or properties of that expression}

In this notation, the vertical bar | is read as “such that”. The following notation defines the set of positive integers less than 100 using set builder notation. It can be read as “the set containing x, where x is an integer, such that x is greater than 0 and less than 100”.

$$\{x \in z \mid 1 < x < 100\}$$

This notation may have included a new symbol for you: ∈. This symbol means “in” or “belongs to” and denotes that an object is a member of a set. Therefore, $x \in z$ can be read as “x belongs to the integers” or “x in the set of integers” or “x is an integer”. We will examine membership in the next section.

The set of rational numbers or complex numbers is easily defined using set builder notation but are quite complicated to define using the roster method.

- Set of rational numbers and complex numbers

$$Q=\{m/n \mid m,n \in Z \text{ and } n \neq 0\}$$

$$C=\{a+bi \mid a,b \in R, i^2=-1\}$$

Note: In set builder notation, conditions which should simultaneously be satisfied are often simply listed after the | and separated by commas. Their conjunction is implied. Alternatively, one can explicitly write “and”.

- Set builder and conjunctions

The following sets are equivalent.

$$\{x \in Z+ \mid x<10 \wedge x \text{ is even}\}$$

$$\{x \in Z+ \mid x<10 \text{ and } x \text{ is even}\}$$

$$\{x \in Z+ \mid x<10, x \text{ is even}\}$$

{2,4,6,8}

Set builder notation has one strong benefit over the roster method: you do not need to know the precise members of a set to construct the set. For example, maybe you want the set of roots of a polynomial.

Natural language description

Just as we can translate between Natural language and predicates, we can similarly use natural language in place of the explicit set builder notation. This is often called a semantic description of the set, where a sentence is used to describe the properties of objects contained in a set.

- Semantic descriptions

The following are valid semantic descriptions of sets.

Let A be the set of three primary colors.

Let B be the set of five smallest positive integers.

Let C be the set of positive rational numbers with 1 as a numerator.

A={red, blue, yellow}

B={1,2,3,4,5}

C= {1,1/2,1/3, 1/4,…}

When using semantic descriptions, the sentence must be as clear as possible and unambiguous. We have already seen in Section 1 how natural language can cause troubles and ambiguity when taken formally.

Interval Notation

For sets of real numbers, a particularly useful notation is interval notation. We are probably familiar with this notation from calculus. For two real numbers a and b, such that a<b, we have the following possible intervals:

$$(a,b)=\{x\in R \mid a<x<b\}$$

$$(a,b]=\{x\in R \mid a<x\leq b\}$$

$$[a,b)=\{x\in R \mid a\leq x<b\}$$

$$[a,b]=\{x\in R \mid a\leq x\leq b\}$$

The choice of including the endpoint or an interval or not determines if the interval is closed or open.

closed interval:

A closed interval is the set of all numbers between two end points, including those endpoints. It is denoted by square brackets. [a,b] is all numbers x such that a≤x≤b.

open interval:

An open interval is the set of all numbers between two end points, excluding those endpoints. It is denoted by round brackets. (a,b) is all numbers x such that a<x<b.

A half-open or half-closed interval is where one end point is included and one end point is excluded. To be more precise, we can say left-open or right-closed to mean only the right end point is included, e.g. (a,b]. We can say left-closed or right-open to mean only the left end point is included, e.g. [a,b).

When we want to describe intervals, which are unbounded on one side (i.e. go until positive or negative infinity) we use a round bracket and the infinity symbol.

$$(-\infty, a) = \{x \in R \mid x < a\}$$

$$(-\infty, a] = \{x \in R \mid x \leq a\}$$

$$(a, \infty) = \{x \in R \mid x > a\}$$

$$[a, \infty) = \{x \in R \mid x \geq a\}$$

Integer intervals:

In computer science, it is increasingly common to define intervals over integers. Such notations have grown as programming languages become higher-level and compilers and interpreters get more sophisticated.

In Python, the function call range(a,b) defines the integer interval (i.e. the set) starting at and including a, and up to but excluding b: {a, a+1, a+2, ..., b-2, b-1}. We can also use the function call range(a,b,step) to define the integer interval {a, a + step, a + 2*step, b-step}.

In MATLAB, we can use the colon operator. i:j defines the closed integer interval from i to j: i, i+1, i+2, ..., j. One can also increment by a step size other than one using i: step: j to obtain the interval I,, i + step, i + 2*step, ..., i + m*step, where m is (j-i) /step rounded down.

Special Sets

- **The universal set**

The set of all the entities in the current context is called the universal set, or simply the universe. It is denoted by U.

The context may be a homework exercise, for example, where the Universal set is limited to the particular entities under its consideration. Also, it may be an arbitrary problem, where we know where it is applied.

- **The empty set**

The set containing no elements at all is called the *null set*, or *empty set*. It is denoted by a pair of empty braces: {} or by the symbol ∅ .

It may seem odd to define a set that contains no elements. Bear in mind, however, that one may be looking for solutions to a problem where it isn't clear at the outset whether or not such solutions even exist. If it turns out that there isn't a solution, then the set of solutions is empty.

For example:

If U = {words in the English language} then {words with more than 50 letters}=∅

If U = {whole numbers} then $\{x|x^2=10\}=\emptyset$.

- **Operations on the empty set**

Operations performed on the empty set (as a set of things to be operated upon) can also be confusing. (Such operations are nullary operations.) For example, the sum of the elements of the empty set is zero, but the product of the elements of the empty set is one (see empty product). This may seem odd since there are no elements of the empty set, so how could it matter whether they are added or multiplied (since "they" do not exist)? Ultimately, the results of these operations say more about the operation in question than about the empty set. For instance, notice that zero is the identity element for addition, and one is the identity element for multiplication.

- **Special numerical sets**

Several sets are used so often, they are given special symbols.

1- Natural numbers

The 'counting' numbers (or whole numbers) starting at 1, are called the natural numbers. This set is sometimes denoted by N. So, N = {1, 2, 3, ...}

Note that, when we write this set by hand, we can't write in bold type so we write an N in blackboard bold font: $\mathbb{N}$

2- Integers

All whole numbers, positive, negative and zero form the set of integers. It is sometimes denoted by Z. So Z = {..., -3, -2, -1, 0, 1, 2, 3, ...}
In blackboard bold, it looks like this: $\mathbb{Z}$

3- Real numbers

If we expand the set of integers to include all decimal numbers, we form the set of real numbers. The set of reals is sometimes denoted by R.

A real number may have a finite number of digits after the decimal point (e.g. 3.625), or an infinite number of decimal digits. In the case of an infinite number of digits, these digits may:

recur; e.g. 8.127127127...

... or they may not recur; e.g. 3.141592653...

In blackboard bold: $\mathbb{R}$

4- Rational numbers

Those real numbers whose decimal digits are finite in number, or which recur, are called rational numbers. The set of rationales is sometimes denoted by the letter Q.

A rational number can always be written as an exact fraction p/q; where p and q are integers. If q equals 1, the fraction is just the integer p. Note that q may NOT equal zero as the value is then undefined.

For example, 0.5, -17, 2/17, 82.01, 3.282828... are all rational numbers.

In blackboard bold: $\mathbb{Q}$

5- Irrational numbers

If a number can't be represented exactly by a fraction p/q, it is said to be irrational.

Examples include: $\sqrt{2}$, $\sqrt{3}$, π

Relationships Between Sets

We'll now look at various ways in which sets may be related to one another.

1. Equality

Two sets A and B are said to be equal if and only if they have the same elements. In this case, we simply write:

$$A = B$$

Note two further facts about equal sets:

- The *order* in which elements are listed does not matter.
- If an element is listed *more than once*, any repeat occurrences are ignored.

So, for example, the following sets are all equal:

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

(You may wonder why one would ever come to write a set like {1, 1, 2,3,2,2}. You may recall that when we defined the *empty set* we noted that there may be no solutions to a particular problem - hence the need for an empty set. Well, here we may be trying several different approaches to solving a problem, some of which lead us to the same solution. When we come to consider the *distinct* solutions, however, any such repetitions would be ignored.)

2. Subsets

If all the elements of a set A are also elements of a set B, then we say that is A a ***subset*** of B and we write:

$$A \subseteq B$$

For example:

In the examples below:

If T= {2,4, 6, 8,10} and E= {even integers}, then $T \subseteq E$

If A= {alphanumeric characters} and P= {printable characters}, then $A \subseteq P$

If Q= {quadrilaterals} and F= {plane figures bounded by four straight lines} , then

$$Q \subseteq F$$

Notice that $A \subseteq B$ does not imply that B must necessarily contain extra elements that are not in A; the two sets could be equal – as indeed Q and F are above. However, if, in addition, B does contain at least one element that isn't in A, then we say that A is a ***proper subset*** of B. In such a case we would write:

$$A \subset B$$

In the examples above:

E contains ... -4, -2, 0, 2, 4, 6, 8, 10, 12, 14, ... , so $T \subset E$

P contains \$, ;, &, ..., so $A \subset P$

But Q and F are just different ways of saying the same thing, so Q=F.

1. Subsets

If all the elements of a set A are also elements of a set B, then we say that A is a ***subset*** of B and we write:

$$A \subseteq B$$

For example:

In the examples below:

If T= {2,4, 6, 8,10} and E={even integers} , then $T \subseteq E$

If A= {alphanumeric characters} and P={printable characters} , then $A \subseteq P$

If Q= {quadrilaterals} and F={plane figures bounded by four straight lines} , then $Q \subseteq F$

Notice that $A \subseteq B$ does not imply that B must necessarily contain extra elements not in A; the two sets could be equal – as indeed Q and F are above. However, if, in addition, B does contain at least one element that isn't in A, then we say that A is a proper subset of B. In such a case we would write:

$$A \subset B$$

In the examples above:

E contains ... -4, -2, 0, 2, 4, 6, 8, 10, 12, 14, ... , so $T \subset E$

P contains $, ;, &, ..., so $A \subset P$

But Q and F are just different ways of saying the same thing, so Q=F.

The use of ⊂ and ⊆; is clearly analogous to the use of < and ≤ when comparing two numbers.

Notice also that *every* set is a subset of the *universal set*, and the *empty set* is a subset of *every* set.

(You might be curious about this last statement: how can the empty set be a subset of *anything*, when it doesn't contain any elements? The point here is that for every set A, the empty set **doesn't** contain any elements that **aren't** in A. So $\emptyset \subseteq A$ for all sets A.)

Finally, note that if $A \subseteq B$ and $B \subseteq A$ then A and B must contain exactly the same elements, and are therefore equal. In other words:

$$A \subseteq B \text{ and } B \subseteq A \text{ then } A=B$$

2. Disjoint

Two sets are said to be ***disjoint*** if they have no elements in common. For example:

If A= {even numbers} and B={1,3,5,11,19}, then A and B are disjoint sets

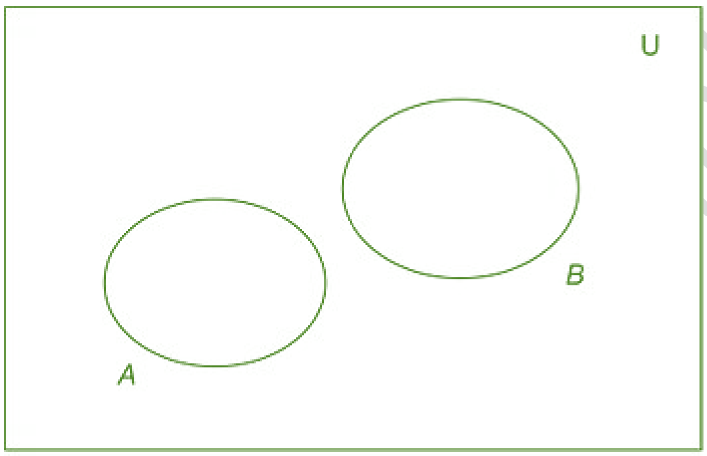
3. Venn Diagrams

A *Venn diagram* can be a useful way of illustrating relationships between sets.

In a Venn diagram:

- The *universal set* is represented by a *rectangle*. Points inside the rectangle represent elements that are in the universal set; points outside represent things not in the universal set. You can think of this rectangle, then, as a 'fence' keeping unwanted things out - and concentrating our attention on the things we're talking about.

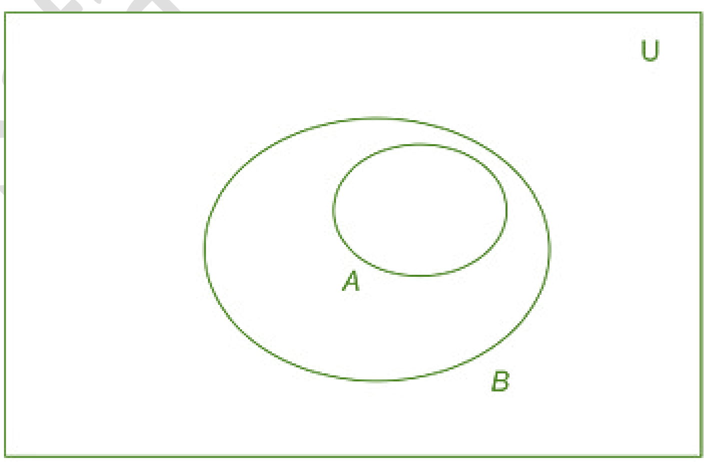
- Other sets are represented by *loops*, usually oval or circular in shape, drawn inside the rectangle. Again, points inside a given loop represent elements in the set it represents; points outside represent things *not* in the set.



A and *B* are *disjoint*

Venn diagrams: Fig. 1

This figure illustrates that the sets *A* and B are disjoint, because the loops don't overlap.

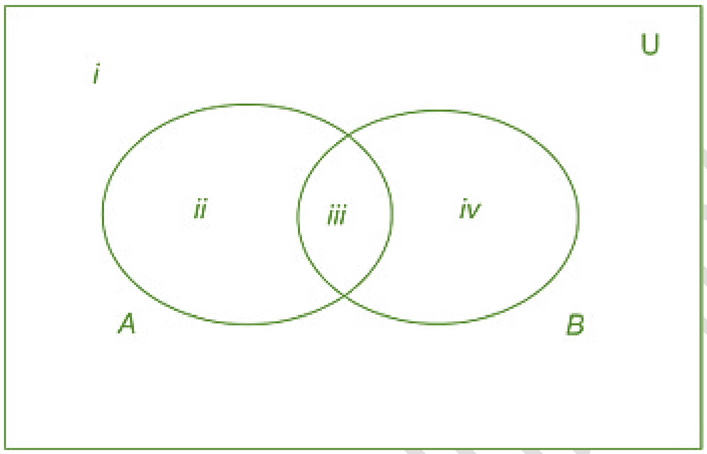


$A \subset B$

Venn diagrams: Fig. 2

On the other hand, figure 2 presents A is a subset of B, because the loop representing set A is entirely enclosed by loop B.

Venn diagrams: Worked Examples



Venn diagrams: Fig. 3

❖ **Example 1**

Fig. 3 represents a Venn diagram showing two sets A and B, in the general case where nothing is known about any relationships between the sets. Note that the rectangle representing the universal set is divided into four regions, labelled i, ii, iii and *iv*.

What can be said about the sets A and B if it turns out that:

(a) region ii is empty?

(b) region iii is empty?

Solution:

(a) If region ii is empty, then A contains no elements that are not in B. So, A is a subset of B, and the diagram should be re-drawn like Fig 2 above.

(b) If region iii is empty, then A and B have no elements in common and are therefore disjoint. The diagram should then be re-drawn like Fig 1 above.

❖ **Example 2**

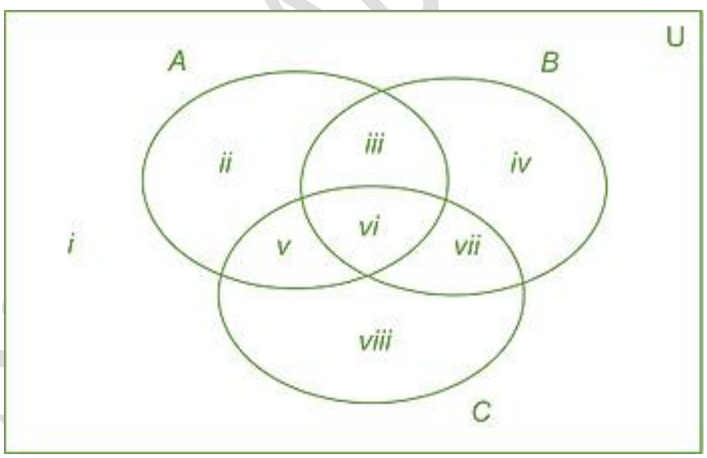
(a) Draw a Venn diagram to represent three sets A, B and C, in the general case where nothing is known about possible relationships between the sets.

(b) Into how many regions is the rectangle representing U divided now?

(c) Discuss the relationships between the sets A, B and C, when various combinations of these regions are empty.

Solution:

(a) The diagram in Fig. 4 shows the general case of three sets where nothing is known about any possible relationships between them.



Venn diagrams: Fig. 4

(b) The rectangle representing U is now divided into 8 regions, indicated by the Roman numerals i to viii.

(c) Various combinations of empty regions are possible. In each case, the Venn diagram can be re-drawn so that empty regions are no longer included. For example:

- If region ii is empty, the loop representing A should be made smaller, and moved inside B and C to eliminate region ii.
- If regions ii, iii and iv are empty, make A and B smaller, and move them so that they are both inside C (thus eliminating all three of these regions), but do so in such a way that they still overlap each other (thus retaining region vi).
- If regions iii and vi are empty, 'pull apart' loops A and B to eliminate these regions, but keep each loop overlapping loop C.
- ...and so on. Drawing Venn diagrams for each of the above examples is left as an exercise for the reader.

❖ **Example 3**

The following sets are defined:

U = {1, 2, 3, …, 10}

A = {2, 3, 7, 8, 9}

B = {2, 8}

C = {4, 6, 7, 10}

Using the two-stage technique described below, draw a Venn diagram to represent these sets, marking all the elements in the appropriate regions.

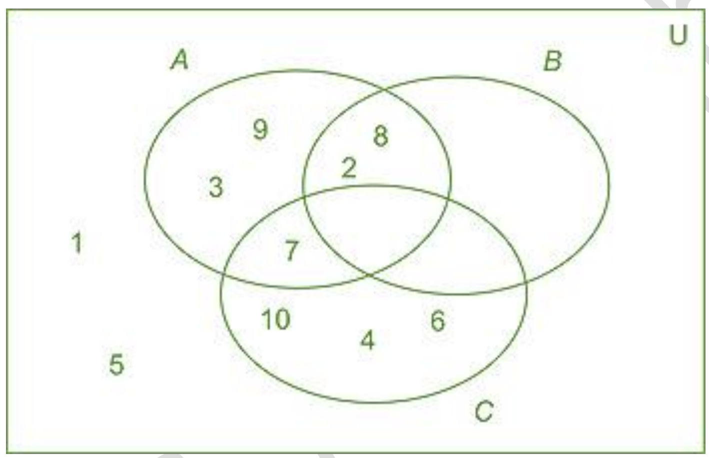
The technique is as follows:

- Draw a 'general' 3-set Venn diagram, like the one in Example 2.
- Go through the elements of the universal set one at a time, once only, entering each one into the appropriate region of the diagram.
- Re-draw the diagram, if necessary, moving loops inside one another or apart to eliminate any empty regions.

Don't begin by entering the elements of set *A*, then set *B*, then *C* – you'll risk missing elements out or including them twice!

<u>*Solution*</u>

After drawing the three empty loops in a diagram looking like *Fig. 4* (but without the Roman numerals!), go through each of the ten elements in **U** - the numbers 1 to 10 - asking each one three questions; like this:



Venn diagrams: Fig. 5

First element: 1

Are you in A? No

Are you in B? No

Are you in *C*? No

A 'no' to all three questions means that the number 1 is outside all three loops. So write it in the appropriate region (region number *i* in *Fig. 4*).

Second element: 2

Are you in A? Yes

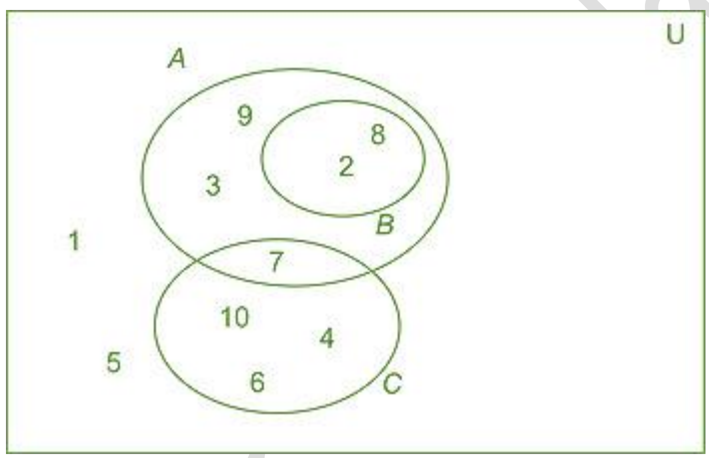
Are you in B? Yes

Are you in *C*? No

Yes, yes, no: so the number 2 is inside *A* and *B* but outside *C*. Goes in region *iii* then.

...and so on, with elements 3 to 10.

The resulting diagram looks like *Fig. 6*.



Venn diagrams: Fig. 6

The final stage is to examine the diagram for empty regions - in this case the regions we called *iv*, *vi* and *vii* in *Fig. 4* - and then re-draw the diagram to eliminate these regions. When we've done so, we shall see the relationships between the three sets.

So we need to:

- pull *B* and C apart, since they don't have any elements in common.
- push *B* inside *A* since it doesn't have any elements outside *A*.

The finished result is shown in *Fig. 6*.

4. The regions in a Venn Diagram and Truth Tables

Perhaps you've realized that adding an additional set to a Venn diagram *doubles* the number of regions into which the rectangle representing the universal set is divided. This gives us a very simple pattern, as follows:

- With one set loop, there will be just two regions: the inside of the loop and its outside.
- With two set loops, there'll be four regions.
- With three loops, there'll be eight regions.
- ...and so on.

It's not hard to see why this should be so. Each new loop we add to the diagram divides each existing region into two, thus doubling the number of regions altogether.

But there's another way of looking at this, and it's this. In the solution to *Example 3* above, we asked three questions of each element: *Are you in A? Are you in B?* and *Are you in C?* Now there are obviously two possible answers to each of these questions: *yes* and *no*. When we *combine* the answers to three questions like this, one after the other, there are then $2^3 = 8$ possible sets of answers altogether. Each of these eight possible combinations of answers corresponds to a different region on the Venn diagram.

The complete set of answers resembles very closely a *Truth Table* - an important concept in Logic, which deals with statements which may be *true* or *false*. The table on the right shows the eight possible combinations of answers for 3 sets *A*, *B* and *C*.

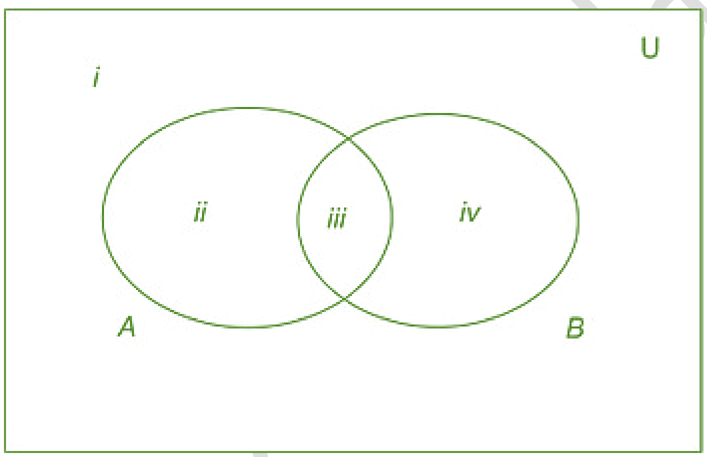
In *A*?	**In *B*?**	**In *C*?**
Y	Y	Y
Y	Y	N
Y	N	Y
Y	N	N
N	Y	Y
N	Y	N
N	N	Y
N	N	N

You'll find it helpful to study the patterns of Y's and N's in each column.

- As you read down column, *C*, the letter changes on every row: Y, N, Y, N, Y, N, Y, N
- Reading down column, *B*, the letters change on every other row: Y, Y, N, N, Y, Y, N, N
- Reading down column, *A*, the letters change every four rows: Y, Y, Y, Y, N, N, N, N

Operations on Sets

Just as we can combine two numbers to form a third number, with operations like 'add', 'subtract', 'multiply' and 'divide', so we can combine two sets to form a third set in various ways. We'll begin by looking again at the Venn diagram which shows two sets *A* and *B* in a general position, where we don't have any information about how they may be related.



Venn diagrams: Fig. 7

In *A*?	In *B*?	Region
Y	Y	*iii*
Y	N	*ii*
N	Y	*iv*
N	N	*i*

The first two columns in the table on the right show the four sets of possible answers to the questions *Are you in A?* and *Are you in B?* for two sets *A* and *B*; the Roman numerals in the third column show the corresponding region in the Venn diagram in *Fig. 7*.

1. Intersection

Region *iii*, where the two loops overlap (the region corresponding to 'Y' followed by 'Y'), is called the *intersection* of the sets *A* and *B*. It is denoted by $A \cap B$. So we can define intersection as follows:

- The ***intersection*** of two sets *A* and *B*, written $A \cap B$, is the set of elements that are in *A* **and** in *B*.

(Note that in ***symbolic logic***, a similar symbol, $\wedge$, is used to connect two logical propositions with the **AND** operator.)

For example, if $A = \{1, 2, 3, 4\}$ and $B = \{2, 4, 6, 8\}$, then $A \cap B = \{2, 4\}$.

We can say, then, that we have combined two sets to form a third set using the *operation of intersection*.

2. Union

In a similar way we can define the *union* of two sets as follows:

- The **union** of two sets *A* and *B*, written $A \cup B$, is the set of elements that are in *A* **or** in *B* (or both).

The union, then, is represented by regions *ii*, *iii* and *iv* in *Fig. 7*.

(Again, in logic a similar symbol, ∨, is used to connect two propositions with the **OR** operator.)

- So, for example, {1, 2, 3, 4} ∪ {2, 4, 6, 8} = {1, 2, 3, 4, 6, 8}.

You'll see, then, that in order to get into the intersection, an element must answer 'Yes' to *both* questions, whereas to get into the union, *either* answer may be 'Yes'.

The ∪ symbol looks like the first letter of 'Union' and like a cup that will hold a lot of items. The ∩ symbol looks like a spilled cup that won't hold a lot of items, or possibly the letter 'n', for the intersection. Take care not to confuse the two.

3. Difference

- The ***difference*** of two sets *A* and *B* (also known as the ***set-theoretic difference*** of *A* and *B*, or the ***relative complement*** of *B* in *A*) is the set of elements that are **in** *A* **but not in** *B*.

This is written *A* - *B*, or sometimes *A* \ B.

The elements in the difference, then, are the ones that answer 'Yes' to the first question *Are you in A?*, but 'No' to the second *Are you in B?*. This combination of answers is on row 2 of the above table, and corresponds to region *ii* in *Fig.7*.

- For example, if *A* = {1, 2, 3, 4} and *B* = {2, 4, 6, 8}, then *A* - *B* = {1, 3}.

4. Complement

So far, we have considered operations in which *two* sets combine to form a third: *binary* operations. Now we look at a *unary* operation - one that involves just *one* set.

- The set of elements that are **not** in a set *A* is called the ***complement*** of *A*. It is written A' (or sometimes A^C, or $\overline{A}$).

  Clearly, this is the set of elements that answer 'No' to the question *Are you in A?*.
- For example, if **U** = **N** and *A* = {odd numbers}, then A' = {even numbers}.
- Notice the spelling of the word *complement*: its literal meaning is 'a complementary item or items'; in other words, 'that which completes'. So, if we already have the elements of *A*, the complement of *A* is the set that *completes* the universal set.

Properties of set operations

- Commutative

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

- Associative

$$A\cup(B\cap C) = (A\cup B) \cap (A\cup C)$$

$$A\cap(B\cup C) = (A\cap B) \cup (A\cap C)$$

- Distributive

$$A \cup (B \cup C) = (A \cup B) \cup C$$

$$A \cap (B \cap C) = (A \cap B) \cap C$$

- Special properties of complements

$$(A')'=A$$

$$U'=\emptyset$$

$$\emptyset'=U$$

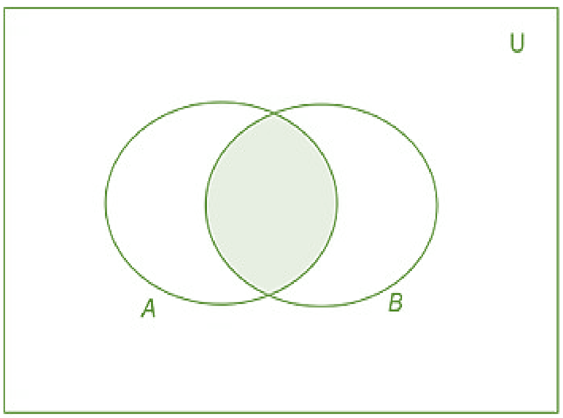
$$A \cap B'= A- B$$

- De Morgan's Law

$$(A \cap B)' = A' \cup B'$$

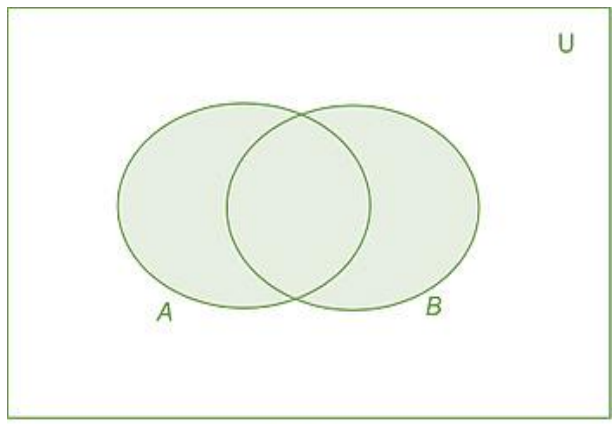
$$(A \cup B)' = A' \cap B'$$

Summary



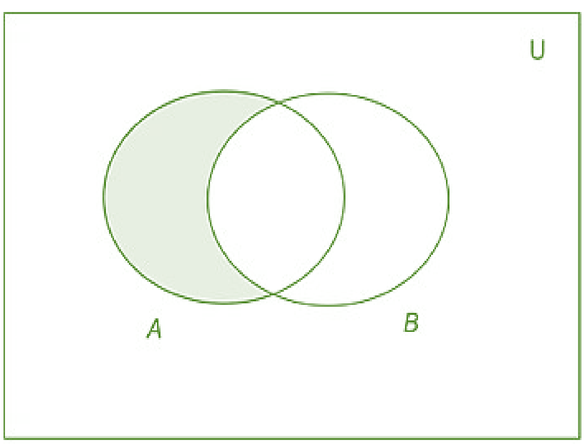
$A \cap B$

Intersection: things that are in *A* **and** in *B*



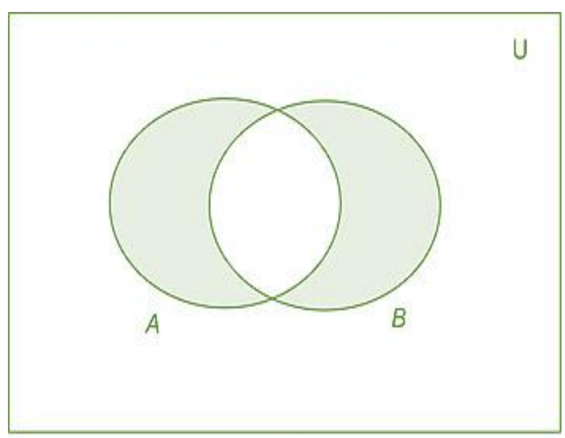
$A \cup B$

Union: things that are in *A* **or** in *B* (or both)



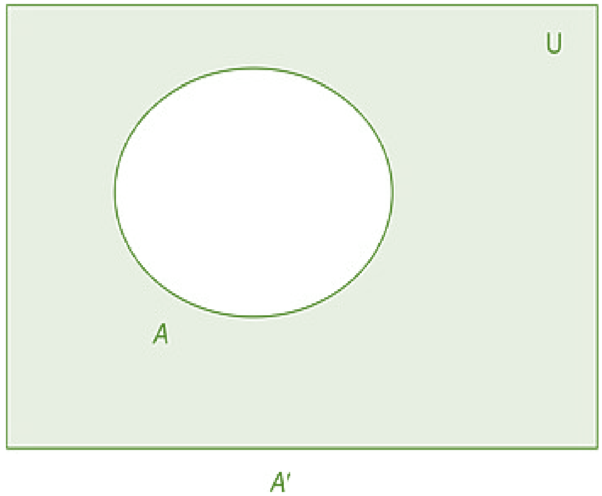
$A - B$

Difference: things that are in *A* **and not** in *B*



$A \Delta B$

Symmetric Difference: things that are in *A* **or** in *B* **but not** both



Complement: things that are **not** in *A*

Cardinality

Finally, in this section on Set Operations we look at an operation on a set that yields not another set, but an integer.

- The cardinality of a finite set A, written | A | (sometimes #(A) or n(A)), is the number of (distinct) elements in A. So, for example:

If A = {lower case letters of the alphabet}, | A | = 26.

Generalized set operations

If we want to denote the intersection or union of n sets, A1, A2, ..., An (where we may not know the value of n) then the following generalized set notation may be useful:

A1 ∩ A2 ∩ ... ∩ An = $\bigcap_{i=1}^{n}$ Ai

A1 ∪ A2 ∪ ... ∪ An = $\bigcup_{i=1}^{n}$ Ai

In the symbol $\bigcap_{i=1}^{n}$ Ai, then, i is a variable that takes values from 1 to n, to indicate the repeated intersection of all the sets A1 to An.

<u>As a Summary for these Sections:</u>

Sets Memorize: A set is a well-defined collection of objects called elements or members of the set.

If x is a member of the set S, we write $x \in S$, and if x is a not member of the set S, we write $x \in S$ Here, well-defined means that any given object must either be an element of the set, or not be an element of the set.

Memorize: We say sets A and B are equal, and write $A = B$ if $x \in A \Leftrightarrow x \in B$ (that is, have exactly the same elements). Here are three ways of specifying a set:

1. Explicit listing: list its elements between brackets, as in $\{2, 3, 5, 7\}$.

2. Implicit listing: list enough of its elements to establish a pattern and use an elipsis (...). At least two elements must be listed to establish the pattern, sometimes more are needed. As examples, consider $\{\ldots -3, -1, 1, 3, \ldots\}$ and $\{0, 2, 4, \ldots, 120\}$, the set of odd integers and the set of non-negative even integers less than or equal to 120, respectively.

Regarding the ways of specifying a set:

- **Explicit listing:** This method involves listing all the elements of the set between braces. For example, $\{2,3,5,7\}$ explicitly lists the elements of the set.
- **Implicit listing:** In this method, enough elements are listed to establish a pattern, and an ellipsis (...) is used to indicate the continuation of the pattern. At least two elements must be listed to establish the pattern, but sometimes more are needed. For example, $\{\ldots -3, -1, 1, 3, \ldots\}$ represents the set of all odd integers, and $\{0, 2, 4, \ldots, 120\}$ represents the set of non-negative even integers less than or equal to 120.

3. Set builder notation: specify the set as the set of all x (say) that make some propositional function true, as in $\{x : (x \text{ is prime}) \wedge (x < 10)\}$.

Note that these are all ways of describing the set, but the set itself does not depend on the description. It just exists, how you describe it is a choice. In particular, for each object, what matters is whether or not it belongs to the set. This is why {1, 2, 2, 3}, {1, 2, 3, 3} and {1, 2, 3} all describe the same set.

Memorize: We say that a set A is a subset of a set B if every element of A is an element of B (i.e., $x \in A \Rightarrow x \in B$). If A is a subset of B we write $A \subseteq B$, and otherwise we write $A \subseteq B$.

Memorize: The empty set is the set that contains no elements. It is denoted by $\emptyset$ or {}. For every set A, we have $A \subseteq A$ and $\emptyset \subseteq A$. Both statements follow from the definition of subset. The second statement is true because the condition $x \in \emptyset$ is never true. (You should be able to explain this if asked.)

Memorize: We say that A is a proper subset of B, and write $A \subset B$, if $A \subseteq B$ and $A = B$. That is, A is a proper subset of B if $A \subseteq B$ and there is an element of B which is not an element of A. This is consistent with the general use of the word "proper" in mathematics - roughly speaking it is used for "not equal to the whole thing". **Notice** that two sets A and B are equal if $x \in A \Leftrightarrow x \in B$. This is the same as $x \in A \Rightarrow x \in B$ and $x \in B \Rightarrow x \in A$. That is $A = B$ is the same as $A \subseteq B$ and $B \subseteq A$.

How to prove two sets A and B are equal. Here are two ways.

1. Showing that each is a subset of the other. A proof like this has two parts. First you show $A \subseteq B$ by starting with "Assume $x \in A$" and then arguing that $x \in B$, and then you show $B \subseteq A$ by starting with "Assume $x \in B$" and then arguing that $x \in A$.

The argument will usually have to make use of other information you know (and/or are given).

2. Using set builder notation to demonstrate that the sets can be described by logically equivalent propositional functions. You must be able to distinguish between $\in$ and $\subseteq$. The first one makes the assertion that a particular object belongs to a set; the second one says that every element of one set belongs to another set. The confusion usually creeps in when the sets in question contain other sets as elements.

Memorize: The power set of a set X is the set P(X) whose elements are the subsets of X. You need to keep the following facts straight:

• P(X) is a set.

• the elements of P(X) are sets (too).

• $A \in P(X) \Leftrightarrow A \subseteq X$ (this is the definition).

• In particular, $\emptyset \in P(X)$ and $X \in P(X)$.

We always assume our sets are subsets of some (large) set called the universe (or universal set), and denoted by U.

Memorize: Let A and B be sets:

• The union of A and B is the set $A \cup B = \{x : x \in A \vee x \in B\}$.

• The intersection of A and B is the set $A \cap B = \{x : x \in A \wedge x \in B\}$.

• The difference of A and B is the set $A - B = \{x : x \in A \wedge x \in B\}$.

• The complement of A is the set $A = \{x : x \in A\} = U - A$.

• The symmetric difference of A and B is the set $A \Delta B = (A - B) \cup (B - A)$.

Note that $A - B$ is, in general, not equal to $B - A$. Set identities. These arise from using set builder notation and the logical equivalences from before (that is, they can all be proved that way).

You should memorize them.

- $A \cap U = A$, $A \cap \emptyset = \emptyset$
- $A \cup U = U$, $A \cup \emptyset = A$
- $A \cup A = A$, $A \cap A = A$
- $A \cup B = B \cup A$, $A \cap B = B \cap A$
- $(A \cap B) \cap C = A \cap (B \cap C)$, $(A \cup B) \cup C = A \cup (B \cup C)$
- Law of Double Complement: $A = A$
- Distributive Laws: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$, $A \cap (B \cup C)=(A \cap B) \cup (A \cap C)$
- DeMorgan's Laws: $A \cup B = A \cap B$, $A \cap B = A \cup B$

You should be able to prove each of the above in two ways (set builder notation and showing that each side is a subset of the other).

Venn diagrams. These are a pictorial representation of sets and a good way to get intuition about (possible) set equalities. You should be able to use Venn diagrams to investigate whether two sets are equal. If they are equal, you should be able to prove this using one of the methods discussed before (a Venn diagram does not suffice as a proof). If the sets are not equal, you should be able to use the Venn diagram to get a particular example showing they are not equal.

1. Subsets

If all the elements of a set A are also elements of a set B, then we say that A is a ***subset*** of B and we write:

$$A \subseteq B$$

For example:

In the examples below:

If T= {2,4, 6, 8,10} and E={even integers} , then $T \subseteq E$

If A= {alphanumeric characters} and P={printable characters} , then $A \subseteq P$

If Q= {quadrilaterals} and F={plane figures bounded by four straight lines} , then $Q \subseteq F$

Notice that $A \subseteq B$ does not imply that B must necessarily contain extra elements not in A; the two sets could be equal – as indeed Q and F are above. However, if, in addition, B does contain at least one element that isn't in A, then we say that A is a proper subset of B. In such a case we would write:

$$A \subset B$$

In the examples above:

E contains ... -4, -2, 0, 2, 4, 6, 8, 10, 12, 14, ... , so $T \subset E$

P contains $, ;, &, ..., so $A \subset P$

But Q and F are just different ways of saying the same thing, so Q=F.

The use of $\subset$ and $\subseteq$; is clearly analogous to the use of $<$ and $\leq$ when comparing two numbers.

Notice also that *every* set is a subset of the *universal set*, and the *empty set* is a subset of *every* set.

(You might be curious about this last statement: how can the empty set be a subset of *anything*, when it doesn't contain any elements? The point here is that for every set A, the empty set **doesn't** contain any elements that **aren't** in A. So $\varnothing \subseteq A$ for all sets A.)

Finally, note that if $A \subseteq B$ and $B \subseteq A$ then A and B must contain exactly the same elements, and are therefore equal. In other words:

$$A \subseteq B \text{ and } B \subseteq A \text{ then } A=B$$

2. Disjoint

Two sets are said to be ***disjoint*** if they have no elements in common. For example:

If A= {even numbers} and B={1,3,5,11,19}, then A and B are disjoint sets

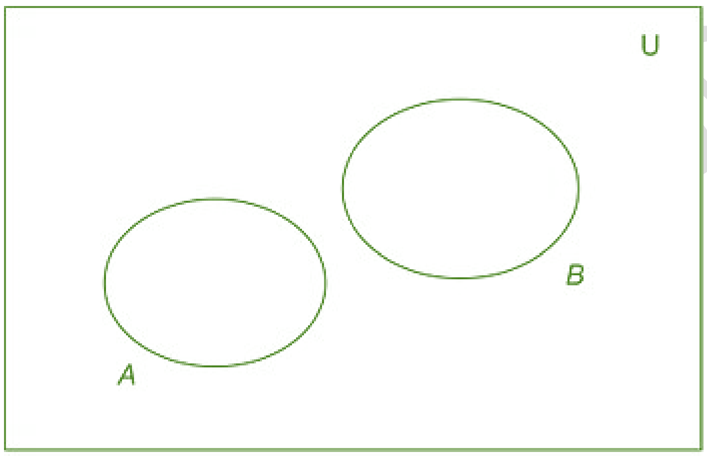
3. Venn Diagrams

A *Venn diagram* can be a useful way of illustrating relationships between sets.

In a Venn diagram:

- The *universal set* is represented by a *rectangle*. Points inside the rectangle represent elements that are in the universal set; points outside represent things not in the universal set. You can think of this rectangle, then, as a 'fence' keeping unwanted things out - and concentrating our attention on the things we're talking about.

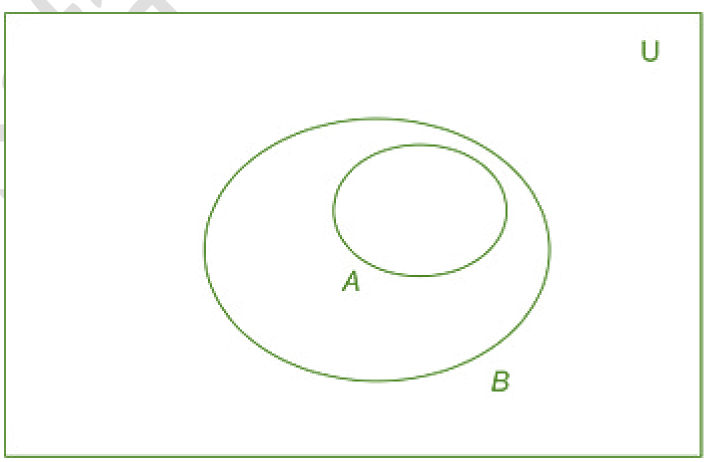
- Other sets are represented by *loops*, usually oval or circular in shape, drawn inside the rectangle. Again, points inside a given loop represent elements in the set it represents; points outside represent things *not* in the set.



A and *B* are *disjoint*

Venn diagrams: Fig. 1

This figure illustrates that the sets *A* and B are disjoint because the loops don't overlap.

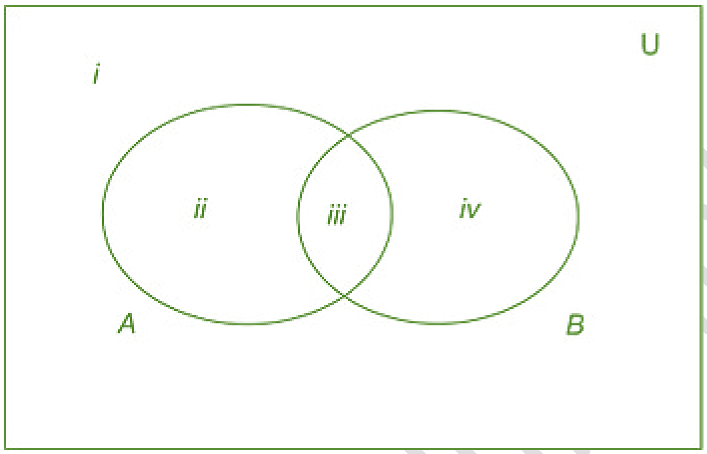


$A \subset B$

Venn diagrams: Fig. 2

On the other hand, figure 2 presents A is a subset of B, because the loop representing set A is entirely enclosed by loop B.

Venn diagrams: Worked Examples



Venn diagrams: Fig. 3

❖ **Example 1**

Fig. 3 represents a Venn diagram showing two sets A and B, in the general case where nothing is known about any relationships between the sets. Note that the rectangle representing the universal set is divided into four regions, labelled i, ii, iii and *iv*.

What can be said about the sets A and B if it turns out that:

(a) region ii is empty?

(b) region iii is empty?

Solution:

(a) If region ii is empty, then A contains no elements that are not in B. So, A is a subset of B, and the diagram should be re-drawn like Fig 2 above.

(b) If region iii is empty, then A and B have no elements in common and are therefore disjoint. The diagram should then be re-drawn like Fig 1 above.

❖ **Example 2**

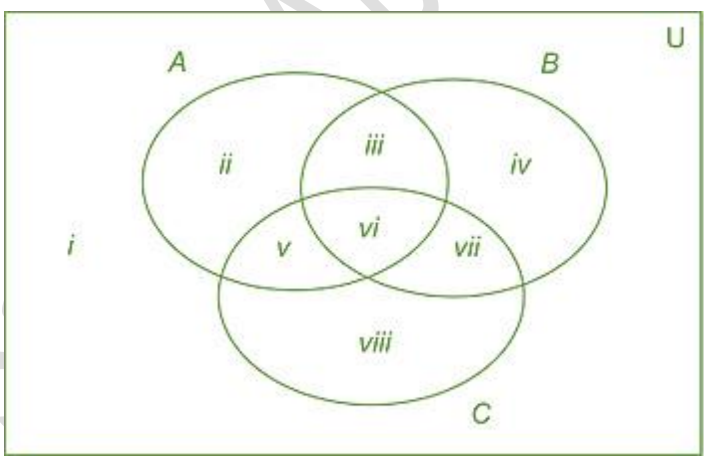
(a) Draw a Venn diagram to represent three sets A, B and C, in the general case where nothing is known about possible relationships between the sets.

(b) Into how many regions is the rectangle representing U divided now?

(c) Discuss the relationships between the sets A, B and C, when various combinations of these regions are empty.

Solution:

(a) The diagram in Fig. 4 shows the general case of three sets where nothing is known about any possible relationships between them.



Venn diagrams: Fig. 4

(b) The rectangle representing U is now divided into 8 regions, indicated by the Roman numerals i to viii.

(c) Various combinations of empty regions are possible. In each case, the Venn diagram can be re-drawn so that empty regions are no longer included. For example:

- If region ii is empty, the loop representing A should be made smaller, and moved inside B and C to eliminate region ii.
- If regions ii, iii and iv are empty, make A and B smaller, and move them so that they are both inside C (thus eliminating all three of these regions), but do so in such a way that they still overlap each other (thus retaining region vi).
- If regions iii and vi are empty, 'pull apart' loops A and B to eliminate these regions, but keep each loop overlapping loop C.
- ...and so on. Drawing Venn diagrams for each of the above examples is left as an exercise for the reader.

❖ **Example 3**

The following sets are defined:

U = {1, 2, 3, …, 10}

A = {2, 3, 7, 8, 9}

B = {2, 8}

C = {4, 6, 7, 10}

Using the two-stage technique described below, draw a Venn diagram to represent these sets, marking all the elements in the appropriate regions.

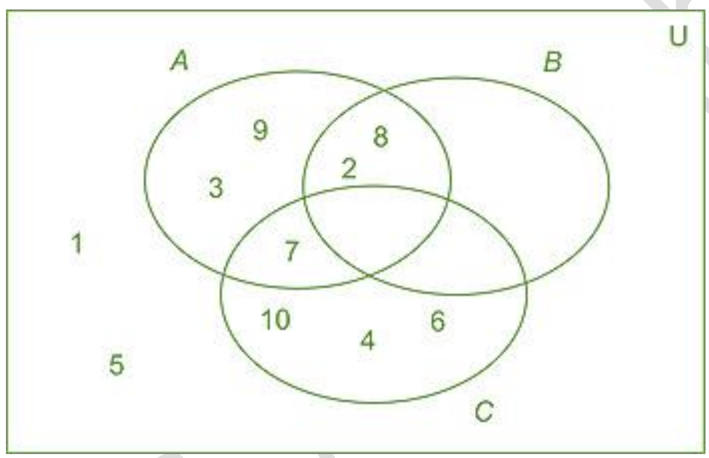
The technique is as follows:

- Draw a 'general' 3-set Venn diagram, like the one in Example 2.
- Go through the elements of the universal set one at a time, once only, entering each one into the appropriate region of the diagram.
- Re-draw the diagram, if necessary, moving loops inside one another or apart to eliminate any empty regions.

Don't begin by entering the elements of set *A*, then set *B*, then *C* – you'll risk missing elements out or including them twice!

<u>*Solution*</u>

After drawing the three empty loops in a diagram looking like *Fig. 4* (but without the Roman numerals!), go through each of the ten elements in **U** - the numbers 1 to 10 - asking each one three questions; like this:



Venn diagrams: Fig. 5

First element: 1

Are you in A? No

Are you in B? No

Are you in *C*? No

A 'no' to all three questions means that the number 1 is outside all three loops. So write it in the appropriate region (region number *i* in *Fig. 4*).

Second element: 2

Are you in A? Yes

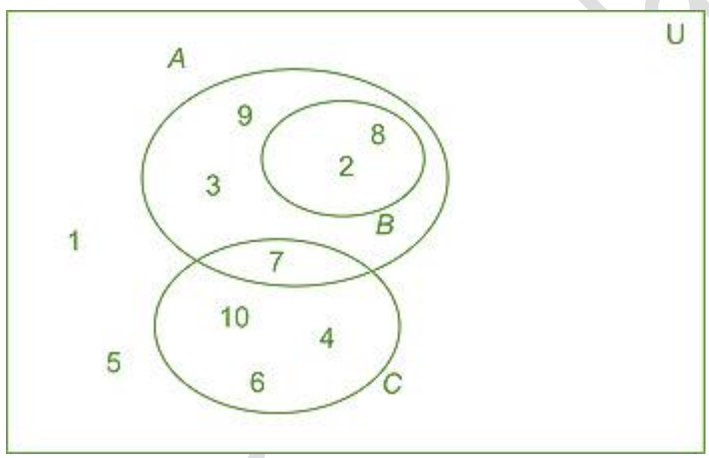
Are you in B? Yes

Are you in *C*? No

Yes, yes, no: so the number 2 is inside *A* and *B* but outside *C*. Goes in region *iii* then.

...and so on, with elements 3 to 10.

The resulting diagram looks like *Fig. 5*.



Venn diagrams: Fig. 6

The final stage is to examine the diagram for empty regions - in this case the regions we called *iv*, *vi* and *vii* in *Fig. 4* - and then re-draw the diagram to eliminate these regions. When we've done so, we shall clearly see the relationships between the three sets.

So we need to:

- pull *B* and C apart, since they don't have any elements in common.

- push *B* inside *A* since it doesn't have any elements outside *A*.

The finished result is shown in *Fig. 6*.

4. The regions in a Venn Diagram and Truth Tables

Perhaps you've realized that adding an additional set to a Venn diagram *doubles* the number of regions into which the rectangle representing the universal set is divided. This gives us a very simple pattern, as follows:

- With one set loop, there will be just two regions: the inside of the loop and its outside.
- With two set loops, there'll be four regions.
- With three loops, there'll be eight regions.
- ...and so on.

It's not hard to see why this should be so. Each new loop we add to the diagram divides each existing region into two, thus doubling the number of regions altogether.

But there's another way of looking at this, and it's this. In the solution to *Example 3* above, we asked three questions of each element: *Are you in A? Are you in B?* and *Are you in C?* Now there are obviously two possible answers to each of these questions: *yes* and *no*. When we *combine* the answers to three questions like this, one after the other, there are then $2^3 = 8$ possible sets of answers altogether. Each of these eight possible combinations of answers corresponds to a different region on the Venn diagram.

In *A*?	**In *B*?**	**In *C*?**
Y	Y	Y
Y	Y	N
Y	N	Y
Y	N	N
N	Y	Y
N	Y	N
N	N	Y
N	N	N

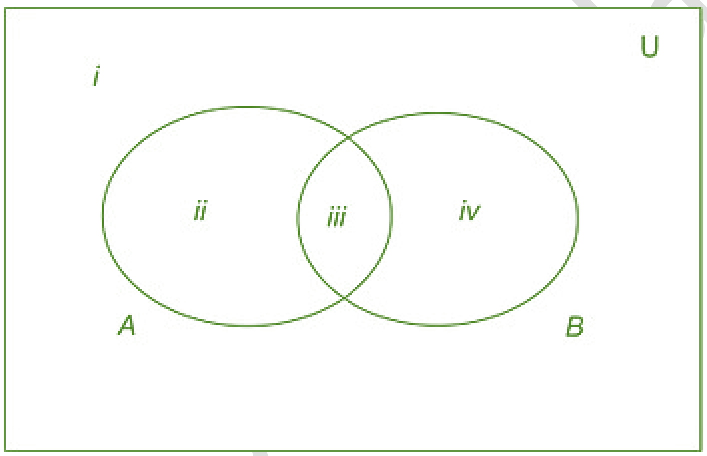
The complete set of answers resembles very closely a *Truth Table* - an important concept in Logic, which deals with statements which may be *true* or *false*. The table on the right shows the eight possible combinations of answers for 3 sets *A*, *B* and *C*.

You'll find it helpful to study the patterns of Y's and N's in each column.

- As you read down column, *C*, the letter changes on every row: Y, N, Y, N, Y, N, Y, N
- Reading down column, *B*, the letters change on every other row: Y, Y, N, N, Y, Y, N, N
- Reading down column, *A*, the letters change every four rows: Y, Y, Y, Y, N, N, N, N

Operations on Sets

Just as we can combine two numbers to form a third number, with operations like 'add', 'subtract', 'multiply' and 'divide', so we can combine two sets to form a third set in various ways. We'll begin by looking again at the Venn diagram which shows two sets A and B in a general position, where we don't have any information about how they may be related.



Venn diagrams: Fig. 7

In *A*?	In *B*?	Region
Y	Y	*iii*
Y	N	*ii*
N	Y	*iv*
N	N	*i*

The first two columns in the table on the right show the four sets of possible answers to the questions *Are you in A?* and *Are you in B?* for two sets *A* and *B*; the Roman numerals in the third column show the corresponding region in the Venn diagram in *Fig. 7*.

1. Intersection

Region *iii*, where the two loops overlap (the region corresponding to 'Y' followed by 'Y'), is called the *intersection* of the sets *A* and *B*. It is denoted by $A \cap B$. So we can define intersection as follows:

- The ***intersection*** of two sets *A* and *B*, written $A \cap B$, is the set of elements that are in *A* **and** in *B*.

(Note that in ***symbolic logic***, a similar symbol, $\wedge$, is used to connect two logical propositions with the **AND** operator.)

For example, if $A = \{1, 2, 3, 4\}$ and $B = \{2, 4, 6, 8\}$, then $A \cap B = \{2, 4\}$.

We can say, then, that we have combined two sets to form a third set using the *operation of intersection*.

2. Union

In a similar way we can define the *union* of two sets as follows:

- The **union** of two sets *A* and *B*, written $A \cup B$, is the set of elements that are in *A* **or** in *B* (or both).

The union, then, is represented by regions *ii*, *iii* and *iv* in *Fig. 7*.

(Again, in logic a similar symbol, ∨, is used to connect two propositions with the **OR** operator.)

- So, for example, {1, 2, 3, 4} ∪ {2, 4, 6, 8} = {1, 2, 3, 4, 6, 8}.

You'll see, then, that in order to get into the intersection, an element must answer 'Yes' to *both* questions, whereas to get into the union, *either* answer may be 'Yes'.

The ∪ symbol looks like the first letter of 'Union' and like a cup that will hold a lot of items. The ∩ symbol looks like a spilled cup that won't hold a lot of items, or possibly the letter 'n', for the intersection. Take care not to confuse the two.

3. Difference

- The ***difference*** of two sets *A* and *B* (also known as the ***set-theoretic difference*** of *A* and *B*, or the ***relative complement*** of *B* in *A*) is the set of elements that are **in** *A* **but not in** *B*.

This is written *A* - *B*, or sometimes *A* \ B.

The elements in the difference, then, are the ones that answer 'Yes' to the first question *Are you in A?*, but 'No' to the second *Are you in B?*. This combination of answers is on row 2 of the above table, and corresponds to region *ii* in *Fig.7*.

- For example, if *A* = {1, 2, 3, 4} and *B* = {2, 4, 6, 8}, then *A* - *B* = {1, 3}.

4. Complement

So far, we have considered operations in which *two* sets combine to form a third: *binary* operations. Now we look at a *unary* operation - one that involves just *one* set.

- The set of elements that are **not** in a set *A* is called the ***complement*** of *A*. It is written *A′* (or sometimes A^C, or $\overline{A}$).

  Clearly, this is the set of elements that answer 'No' to the question *Are you in A?*.

- For example, if **U** = **N** and *A* = {odd numbers}, then *A′* = {even numbers}.
- Notice the spelling of the word *complement*: its literal meaning is 'a complementary item or items'; in other words, 'that which completes'. So, if we already have the elements of *A*, the complement of *A* is the set that *completes* the universal set.

Properties of set operations

- Commutative

$$A \cup B = B \cup A$$

$$A \cap B = B \cap A$$

- Associative

$$A \cup (B \cap C) = (A \cup B) \cup C$$

$$A \cap (B \cup C) = A \cap (B \cap C)$$

- Distributive

$$A \cup (B \cup C) = (A \cup B) \cup C$$

$$A \cap (B \cap C) = A \cap (B \cap C)$$

- Special properties of complements

$$(A')'=A$$

$$U'=\varnothing$$

$$\varnothing'=U$$

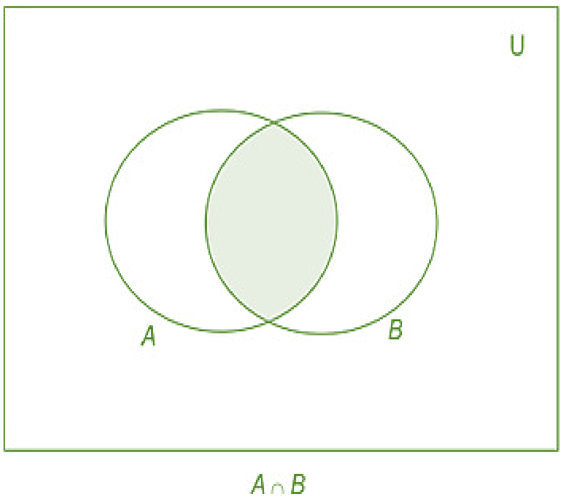
$$A \cap B'= A- B$$

- De Morgan's Law

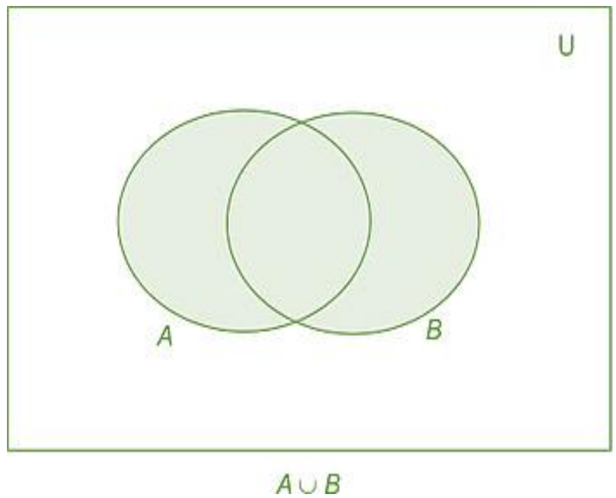
$$(A \cap B)' = A' \cap B'$$

$$(A \cup B)' = A' \cup B'$$

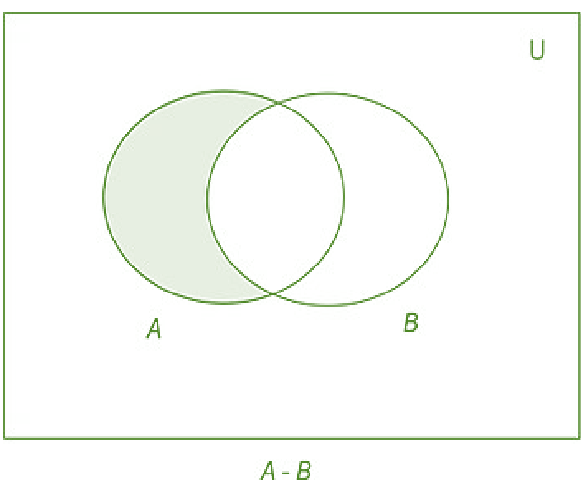
Summary



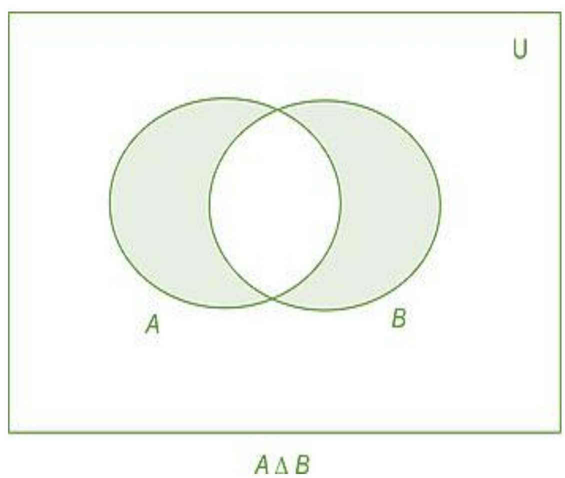
Intersection: things that are in *A* **and** in *B*



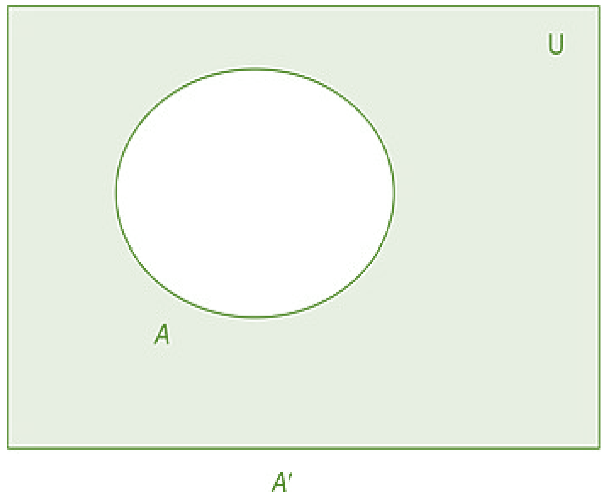
Union: things that are in *A* **or** in *B* (or both)



Difference: things that are in *A* **and not** in *B*



Symmetric Difference: things that are in *A* **or** in *B* **but not** both



Complement: things that are **not** in *A*

Cardinality

Finally, in this section on Set Operations we look at an operation on a set that yields not another set, but an integer.

- The cardinality of a finite set A, written | A | (sometimes #(A) or n(A)), is the number of (distinct) elements in A. So, for example:

If A = {lower case letters of the alphabet}, | A | = 26.

Generalized set operations

If we want to denote the intersection or union of n sets, A1, A2, ..., An (where we may not know the value of n) then the following generalized set notation may be useful:

A1 ∩ A2 ∩ ... ∩ An = $\bigcap_{i=1}^{n}$ Ai

A1 ∪ A2 ∪ ... ∪ An = $\bigcup_{i=1}^{n}$ Ai

In the symbol $\bigcap_{i=1}^{n}$ Ai, then, i is a variable that takes values from 1 to n, to indicate the repeated intersection of all the sets A1 to An.

As a Summary for these Sections:

Sets Memorize: A set is a well-defined collection of objects called elements or members of the set.

If x is a member of the set S, we write $x \in S$, and if x is a not member of the set S, we write $x \in S$ Here, well-defined means that any given object must either be an element of the set, or not be an element of the set.

Memorize: We say sets A and B are equal, and write $A = B$ if $x \in A \Leftrightarrow x \in B$ (that is, have exactly the same elements). Here are three ways of specifying a set:

1. Explicit listing: list its elements between brackets, as in $\{2, 3, 5, 7\}$.

2. Implicit listing: list enough of its elements to establish a pattern and use an elipsis (...). At least two elements must be listed to establish the pattern, sometimes more are needed. As examples, consider $\{\ldots -3, -1, 1, 3, \ldots\}$ and $\{0, 2, 4, \ldots, 120\}$, the set of odd integers and the set of non-negative even integers less than or equal to 120, respectively.

3. Set builder notation: specify the set as the set of all x (say) that make some propositional function true, as in $\{x : (x \text{ is prime}) \wedge (x < 10)\}$.

Note that these are all ways of describing the set, but the set itself does not depend on the description. It just exists, how you describe it is a choice. In particular, for each object, what matters is whether or not it belongs to the set. This is why $\{1, 2, 2, 3\}$, $\{1, 2, 3, 3\}$ and $\{1, 2, 3\}$ all describe the same set.

Memorize: We say that a set A is a subset of a set B if every element of A is an element of B (i.e., $x \in A \Rightarrow x \in B$). If A is a subset of B we write $A \subseteq B$, and otherwise we write $A \subseteq B$.

Memorize: The empty set is the set that contains no elements. It is denoted by $\emptyset$ or $\{\}$. For every set A, we have $A \subseteq A$ and $\emptyset \subseteq A$. Both statements follow from the definition of subset. The second statement is true because the condition $x \in \emptyset$ is never true. (You should be able to explain this if asked.)

Memorize: We say that A is a proper subset of B, and write $A \subset B$, if $A \subseteq B$ and $A = B$. That is, A is a proper subset of B if $A \subseteq B$ and there is an element of B which is not an element of A. This is consistent with the general use of the word "proper" in mathematics - roughly speaking it is used for "not equal to the whole thing". **Notice** that two sets A and B are equal if $x \in A \Leftrightarrow x \in B$. This is the same as $x \in A \Rightarrow x \in B$ and $x \in B \Rightarrow x \in A$. That is $A = B$ is the same as $A \subseteq B$ and $B \subseteq A$.

How to prove two sets A and B are equal. Here are two ways.

1. Showing that each is a subset of the other. A proof like this has two parts. First you show $A \subseteq B$ by starting with "Assume $x \in A$" and then arguing that $x \in B$, and then you show $B \subseteq A$ by starting with "Assume $x \in B$" and then arguing that $x \in A$. The argument will usually have to make use of other information you know (and/or are given).

2. Using set builder notation to demonstrate that the sets can be described by logically equivalent propositional functions. You must be able to distinguish between $\in$ and $\subseteq$. The first one makes the assertion that a particular object belongs to a set; the second one says that every element of one set belongs to another set.

The confusion usually creeps in when the sets in question contain other sets as elements.

Memorize: The power set of a set X is the set P(X) whose elements are the subsets of X. You need to keep the following facts straight:

• P(X) is a set.

• the elements of P(X) are sets (too).

• $A \in P(X) \Leftrightarrow A \subseteq X$ (this is the definition).

• In particular, $\emptyset \in P(X)$ and $X \in P(X)$.

We always assume our sets are subsets of some (large) set called the universe (or universal set), and denoted by U.

Memorize: Let A and B be sets:

• The union of A and B is the set $A \cup B = \{x : x \in A \vee x \in B\}$.

• The intersection of A and B is the set $A \cap B = \{x : x \in A \wedge x \in B\}$.

• The difference of A and B is the set $A - B = \{x : x \in A \wedge x \in B\}$.

• The complement of A is the set $A = \{x : x \in A\} = U - A$.

• The symmetric difference of A and B is the set $A \Delta B = (A - B) \cup (B - A)$.

Note that A − B is, in general, not equal to B − A. Set identities. These arise from using set builder notation and the logical equivalences from before (that is, they can all be proved that way).

You should memorize them.

• $A \cap U = A$, $A \cap \emptyset = \emptyset$

• $A \cup U = U$, $A \cup \emptyset = A$

- A ∪ A = A, A ∩ A = A
- A ∪ B = B ∪ A, A ∩ B = B ∩ A
- (A ∩ B) ∩ C = A ∩ (B ∩ C), (A ∪ B) ∪ C = A ∪ (B ∪ C)
- Law of Double Complement: A = A
- Distributive Laws: A ∪ (B ∩ C)=(A ∪ B) ∩ (A ∪ C), A ∩ (B ∪ C)=(A ∩ B) ∪ (A ∩ C)
- DeMorgan's Laws: A ∪ B = A ∩ B, A ∩ B = A ∪ B

You should be able to prove each of the above in two ways (set builder notation and showing that each side is a subset of the other).

Venn diagrams. These are a pictorial representation of sets and a good way to get intuition about (possible) set equalities. You should be able to use Venn diagrams to investigate whether two sets are equal. If they are equal, you should be able to prove this using one of the methods discussed before (a Venn diagram does not suffice as a proof). If the sets are not equal, you should be able to use the Venn diagram to get a particular example showing they are not equal.

Cartesian Product

The final set operation we will consider is their Cartesian product. But first, we need a new kind of discrete structure.

A tuple is a finite ordered sequence of elements (from a set). An n-tuple is a tuple with n elements.

We have special names for tuples of certain lengths:

a 1-tuple is a monuple or a singleton,

a 2-tuple is a couple or a pair,

a 3-tuple is a triple or triplet,

a 4-tuple is a quadruple,

a 5-tuple is a pentuple, and so on

Tuples are denoted using parentheses. For example, (a,b) is a tuple, (c,b,a) is a tuple, and (c,c,a,b,a) is a tuple.

Note: Unlike sets, ordering is important for tuples. Moreover, the same element may appear more than once in a tuple. Therefore (a,b) , (b,a), and (a,b,a) are all different tuples.

The elements of a tuple are called coordinates. In a tuple (1,2,3), a is the first coordinate, b is the second coordinate, and c is the third coordinate. Two tuples are equal if and only if every coordinate is equal.

$$(a_1, b_1) = (a_2, b_2) \longleftrightarrow a_1 = a_2 \land b_1 = b_2$$

Tuple equality: The tuples (3,7,5) and (3,7,5) are equal. However, $(3,7,5) \neq (3,5,7)$.

The Cartesian product of two (not necessarily different) sets is a new set of tuples constructed from the elements of the two input sets.

Cartesian product: The Cartesian product of two sets A and B is the set of all pairs (a, b) where $a \in A$ and $b \in B$. The Cartesian product is denoted $A \times B$.

Cartesian product General Formula:

Let A= {1,2,3} and B= {x, y}.

A×B= {(1, x), (2, x), (3, x), (1, y), (2, y), (3, y)}

B×A= {(x, 1), (y, 1), (x, 2), (y, 2), (x, 3), (y, 3)}

One way of viewing the Cartesian product A×B is as a table. The columns range over all possible values of A and the rows range over all possible values of B.

If A= {1, 2, 3} and B= {x, y, z} we have:

	x	y	z
1	$(1, x)$	$(1, y)$	$(1, z)$
2	$(2, x)$	$(2, y)$	$(2, z)$
3	$(3, x)$	$(3, y)$	$(3, z)$

A Cartesian product also extends to more than two sets.

$$A \times B \times C = \{(a, b, c) \mid a \in A, b \in B, c \in C\}$$

$$A_1 \times A_2 \times \cdots \times A_n = \{(a_1, a_2, \ldots, a_n) \mid a_i \in A_i \text{ for } i = 1, \ldots, n\}$$

When a Cartesian product is made between a set and itself, i.e. A×A, we can also write it as A^2. This continues for A^3, A^4, ... , A^n.

Cartesian and Complex planes

The Cartesian plane can be described by $\boldsymbol{R^2}$, all ordered pairs of real numbers. You are probably already familiar with this from calculus. Three-dimensional space is similarly described by $\boldsymbol{R^3}$.

We can also identify the complex plane with $\boldsymbol{R^2}$. Indeed, the complex numbers $\boldsymbol{C}$ are of the form $a+b_i$ where a,b Є R. Therefore, each complex number $a+b_i$ can be identified with a tuple (a,b) Є $\boldsymbol{R^2}$ (that is, $\boldsymbol{R^2}$ and $\boldsymbol{C}$ are isomorphic).

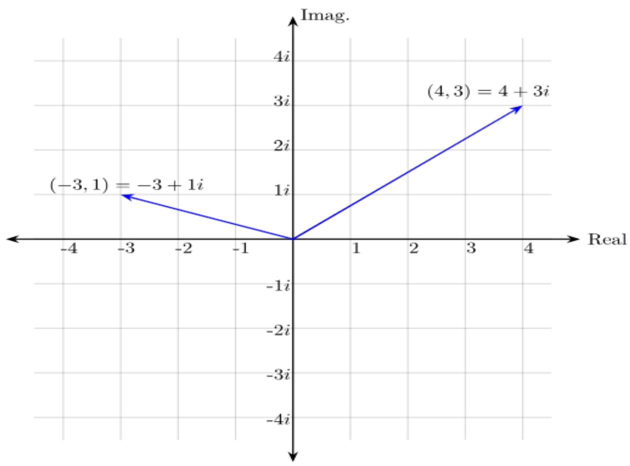


Figure 1. The complex plane as a $\boldsymbol{R}\times\boldsymbol{R}$ coordinate system

H.W1// Write out the following sets using the roster method.

Let Q(x,y) := x(2y+4) = 0

1. {a | x Є Z^+ and x < 5}

2. {x Є Q | 2x Є Z}

H.w2 // Let A = {x | x is an odd integer and 2x + 5 > 20}.
Let B = {x Є Z+ | x > 7 and ∃k Є Zx = 2k+ 1}.

Is A a subset of B? Is B a subset of A?

Graphs:

Graphs are discrete structures consisting of vertices and edges that connect these vertices, so a graph G(V,E) consists of:
(i) V, a nonempty set of *vertices* (or *nodes*).
(ii) *E*, a set of *edges*.

Each *edge* has one or two vertices associated with it, called its *endpoints*. Graphs are used in a wide variety of applications with computer science such as communication networks, logical design, transportation networks, formal languages, compiler writing and retrieval.

A graph is one of the most foundational and useful data structures in computer science. If there are two things in discrete mathematics to take into the rest of your computer science studies it is induction and graphs.

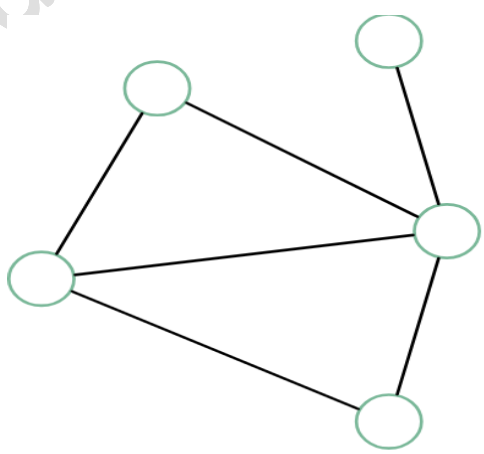


Fig. 1 A graph with 5 vertices and 6 edges.

In their simplest form, a graph is nothing more than vertices (nodes, dots, circles, squares, shapes), connected by edges (lines between vertices).

Graphs are handy in practice and have countless applications. To name just a few:

- Computers in a network (e.g. the Internet)
- Destinations in a transport network (e.g. bus stops, airports)
- Social connections (e.g. friends on Facebook)
- Semantic and conceptual connections (e.g. fish is “related” to the ocean)
- Components of a software system (e.g. classes, modules, libraries)

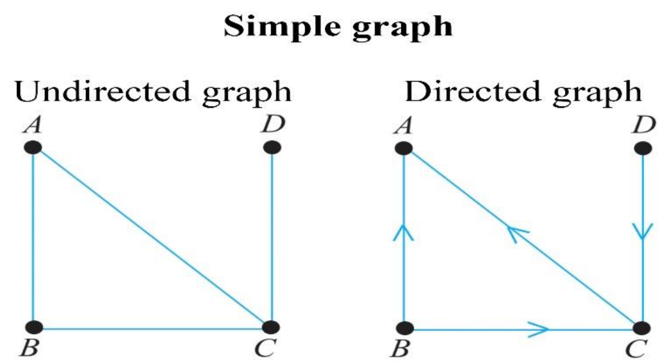
In this Lectures, we will explore graphs as mathematical objects, their properties, how to encode graphs on computers, and their applications in practice.

Graphs

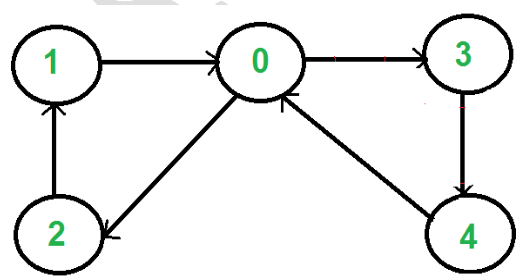
Types of the graphs:

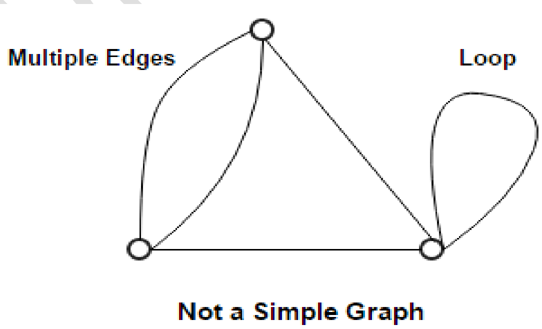
1. simple graph
2. multigraphs
3. directed graph.

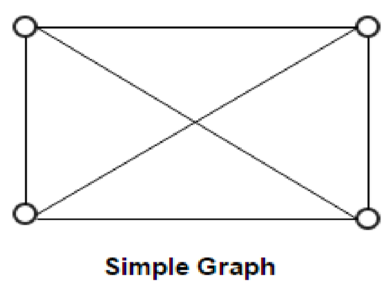
For example: in a communication network, where computers can be represented by vertices and communication links by edges. A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices is called a **simple graph**.



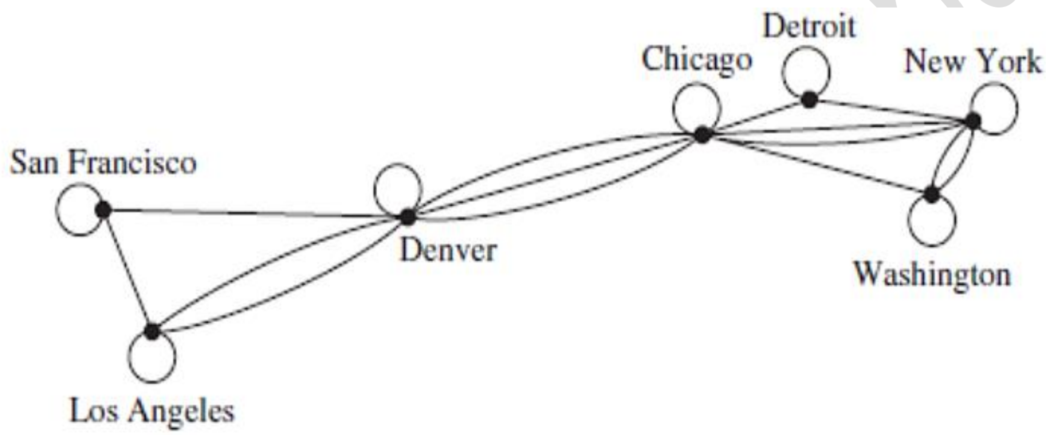
Directed Graph



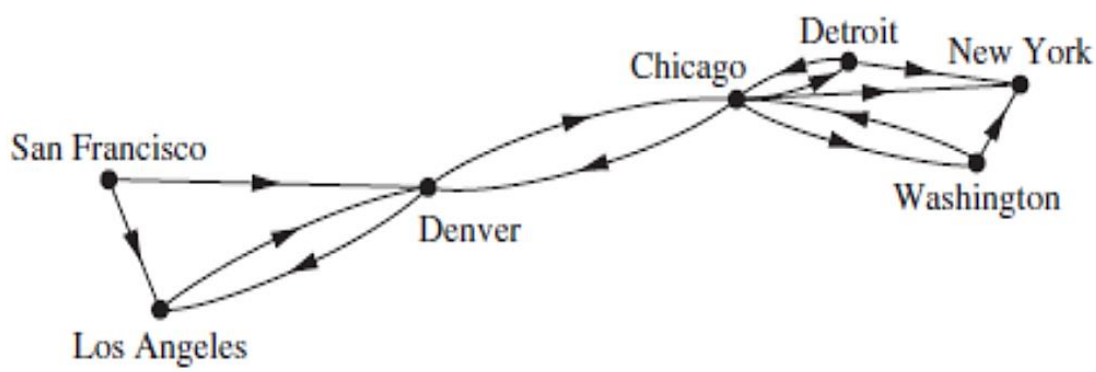




A computer network may contain multiple links between data centers, as shown in the following figure. We need graphs with more than one edge connecting the same pair of vertices to model such networks Graphs that may have multiple edges connecting the same vertices are called **multigraphs**. sometimes a communications link connects a data center with itself, perhaps a feedback loop for diagnostic purposes. Such a network is illustrated in the following Figure. To model this network, we need to include edges that connect a vertex to itself. Such edges are called **loops**.



Also, we can use the directed Graph to determine the direction of the sent data as shown in follows where each edge of a directed graph is associated with an ordered pair.



In summary, a **multigraph** is a graph which is permitted to have multiple edges (also called parallel edges), that is, edges that have the same end nodes. Thus, two vertices may be connected by more than one edge.

For example, we have in the Following the graph G(V, E) where: V consists of four vertices A, B, C, D; and, E consists of five edges

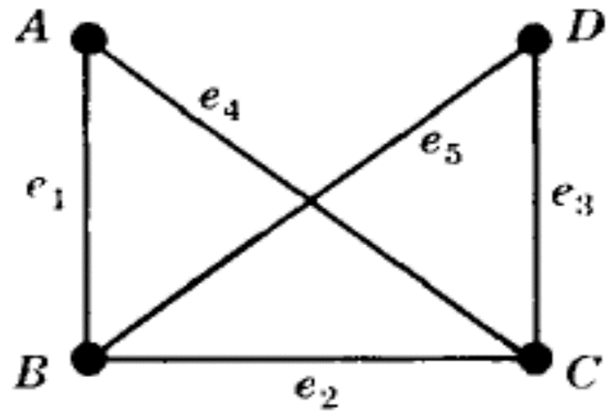
e1 ={A, B},

e2 = {B, C},

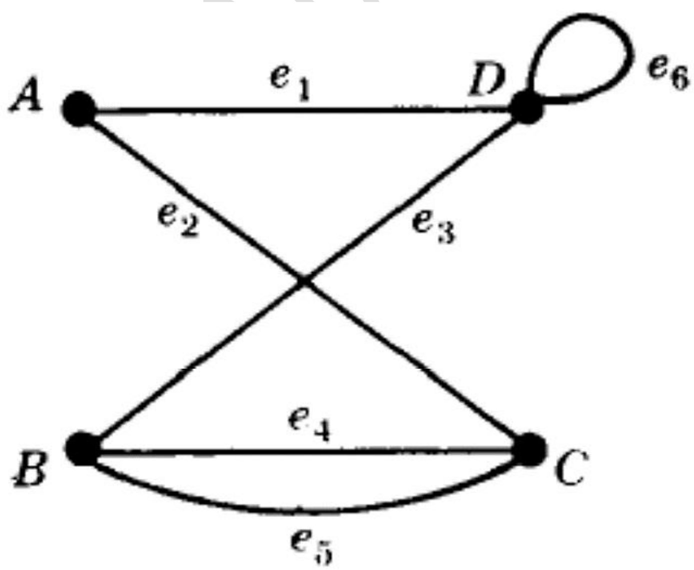
e3 = {C, D},

e4 = {A, C},

e5 = {B, D}.



Vertices u and v are said to be **adjacent** if there is an edge e={u,v}. In such a case, u and v are called the endpoints of e, and e is said to connect u and v. Also, the edge e is said to be **incident** on each of its endpoints u and v.



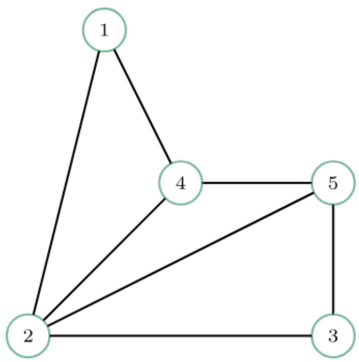
In this figure a multigraph with:

1) multiple edges e4 & e5

2) a loop e6

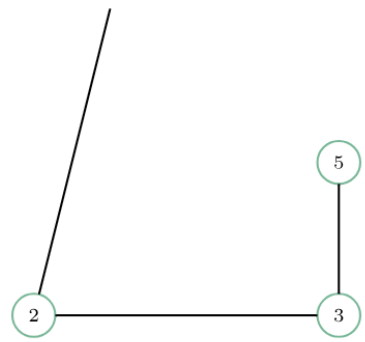
Example 1:

Examine the options below and identify which one qualifies as a graph. Explain your choice.



G1

G1



G3

The left-most graph G1 is a simple graph with 5 vertices and 7 edges: {1,2,3,4,5} and {{1,2}, {2,4}, {4,5}, {5,3}, {3,2}, {2,5}{1,4}}. The G2 is a proper subgraph of G1. It has vertices {2,3,5} and edges {{2,3}, {3,5}}. However, G3 is *not* a graph. It has an edge with no endpoint! Since G3 is not a valid graph, it cannot be a subgraph of G1.

Degree:

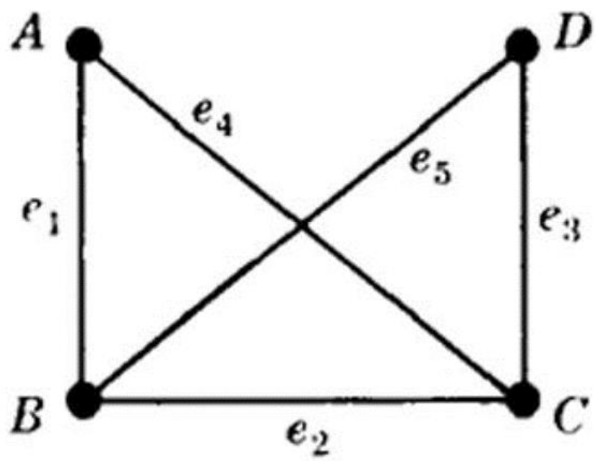
The degree of a vertex v [deg(v)], is equal to the number of edges which are incident on v. since each edge is counted twice in counting the degrees of the vertices of a graph.

Theorem: The sum of the degrees of the vertices of a graph is equal to twice the number of edges. Let $G = (V, E)$ be an undirected graph with m edges. Then

$2m = \Sigma\ deg(v).$

$v \in V$

For example, in the following figure, we have:



deg(A) = 2,

deg(B) = 3,

deg(C) = 3,

deg(D) = 2

The sum of the degrees = twice the number of edges = 2×5=10

EXAMPLE 2: How many edges are there in a graph with 10 vertices each of degree six?

Sol// Because the sum of the degrees of the vertices is 6 × 10 = 60, it follows that 2m = 60

where m is the number of edges. Therefore, m = 30.

A vertex is said to be even or odd according to its degree is an even or odd number. Thus, A and D are even vertices whereas B and C are odd vertices.

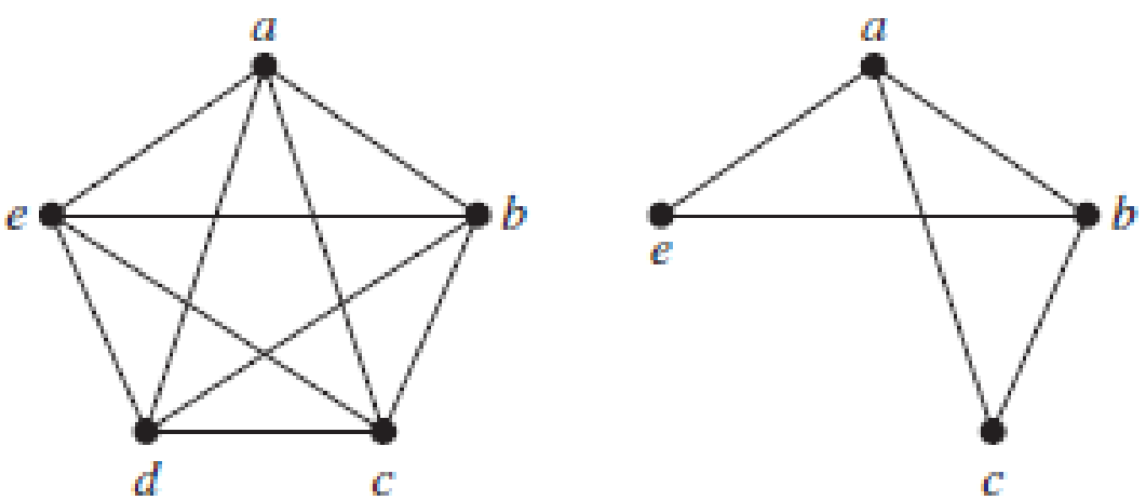
This theorem also holds for multigraphs where a loop is counted twice towards the degree of its endpoint. For example, in Figure of page 3 example, we have deg (D) = 4 since the edge e6 is counted twice; hence D is an even vertex.

A vertex of degree zero is called an isolated vertex.

Subgraphs

Consider a graph G = G(V,E) and a graph H = H(V', E') is called a subgraph of G if the vertices and edges of H are contained in the vertices and edges of G, that is, if V' ⊆ V and E' ⊆ E. Sometimes we need only part of a graph to solve a problem. For instance, we may care only about the part of a large computer the network that involves the computer centers in New York, Denver, Detroit, and Atlanta. Then we can ignore the other computer centers and all telephone lines not linking two of these specific four computer centers. In the graph model for the large network, we can remove the vertices corresponding to the computer centers other than the four of interest, and we can remove all edges incident with a vertex that was removed. When edges and vertices are removed from a graph, without removing endpoints of any remaining edges, a smaller graph is obtained. Such a graph is called a **subgraph** of the original graph.

EXAMPLE 3: The graph *G* shown in this example is a subgraph of *K*5. If we add the edge connecting a, b, *c* and *e* to *G*, we obtain the subgraph induced by $W = \{a, b, c, e\}$.



Connectivity

Arguably the most important thing about graphs is that they encode connections. Recall that the definition of a graph only requires vertices and connections (edges) between them. We don't care about the placement of vertices the length of edges, or even if edges overlap.

All we care about in a simple graph is whether two vertices are adjacent or not.

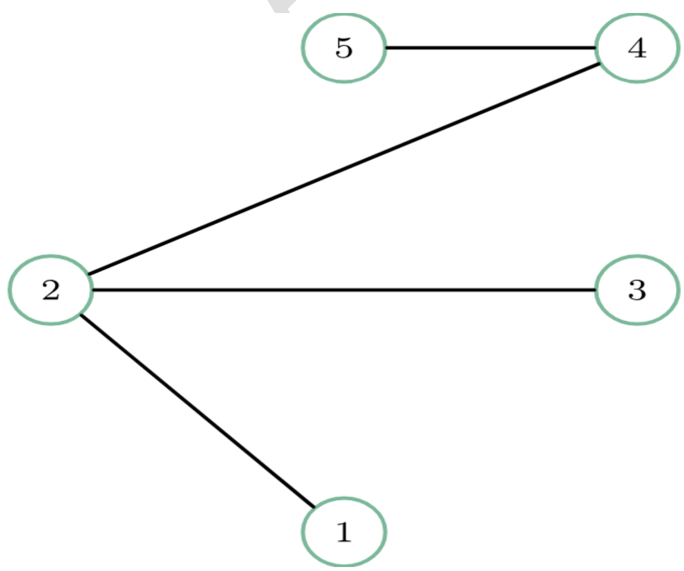
Understanding whether two vertices are connected is the study of connectivity.

Path: A *path* of a simple graph G= (V, E) is a sequence of vertices (V_n) where an edge exists between V_i and V_{i+1} for $1 \leq i < n$.

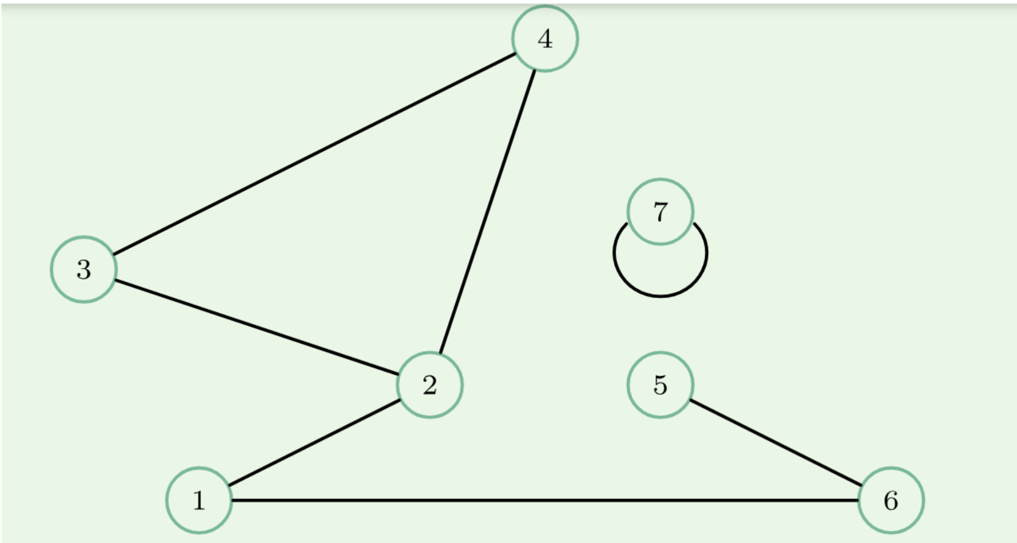
As a consequence of a path, we have an alternative definition for adjacent vertices. Two vertices are adjacent if they are connected with a path of length 1.

From paths, we get a formal definition of connectivity.

Connected: Two vertices u and, v in a graph are *connected* if there exists a path from u to v. Otherwise, u and v are said to be *disconnected*. A graph is *connected* if every pair of vertices in the graph is connected.



The above graph has many paths. And, in particular, *every* vertex is connected to every other vertex. Thus, the graph is connected. In contrast, the following graph is connected.



We also have useful terminology around paths in a graph:

1. When a path begins and ends at the same vertex it is called a **circuit**
2. A path (v1, v2, …, vn) is said to **pass through** the vertices v2, v3, …, vn−1 and **traverse** the edges {v1, v2}, {v2, v3}, …, {vn−1, vn}.
3. A path is **simple** if every edge in the path only appears *once*.
4. When a path exists between u and v, we say that u is **reachable** from v.

Connectivity (Terms and Examples):

A **walk** in a multigraph G consists of an alternating sequence of vertices and edges of the form:
v0, e1, v1, e2, v2,……., en-1,vn-1,en,vn
Where each edge e_i contains the vertices v_i-1 and v_i (which appear on the sides of ei in the sequence).
Length of walk: is the number n of edges. When there is no ambiguity, we denote a path by its sequence of vertices (v0, v1, . . . , vn).
Closed walk: the walk is said to be closed if v0 = vn . Otherwise, we say that the walk is from v0 to vn.
Trail: is a walk in which all edges are distinct.
Path: is a walk in which all vertices are distinct.
Cycle: is a closed walk such that all vertices are distinct except v1 = vn, A cycle of length k is called a k-cycle.

Example 1
In the simple graph shown in Figure 1:

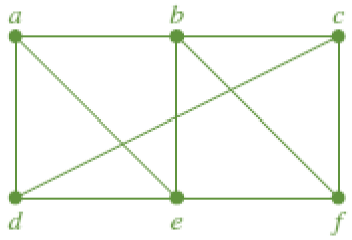


Figure 1

a, *d*, *c*, *f* , *e* is a path of length 4, because {*a*, *d*}, {*d*, *c*}, {*c*, *f* }, and {*f*, *e*} are all edges. However,

d, *e*, *c*, *a* is not a path, because {*e*, *c*} is not an edge. *Note* that *b*, *c*, *f* , *e*, *b* is a circuit of length 4 because {*b*, *c*}, {*c*, *f* }, {*f*, *e*}, and {*e*, *b*} are edges, and this path begins and ends at *b*.

The walk *a*, *b*, *e*, *d*, *a*, *b*, which is of length 5, is not a path because it contains the edge {*a*, *b*} twice.

Example 2: Consider the graph in Figure (2), then

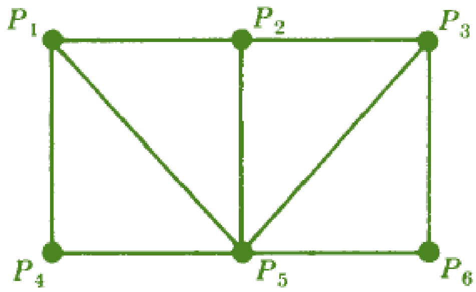


Figure (2)

The sequence: (P4, P1, P2, P5, P1, P2, P3, P6) is a walk from P4 to P6. It is not a trail since the edge {P1, P2} is used twice.

The sequence: (P4, P1, P5, P3, P2, P6) Is not a walk since there is no edge {P2, P6}.

The sequence: (P4, P1, P5, P3, P6) Is a path from P4 to P6.

The shortest path from P4 to P6 is (P4, P5, P6) which has length = 2 (2 edges only)
The distance between vertices u & v d(u,v) is the length of the shortest path d(P4,P6) = 2

Tree graph:

A graph T is called a *tree* if T is connected and T has no cycles. Consider a tree T . Clearly, there is only one simple path between two vertices of T ; otherwise, the two paths would form a cycle. Also:

Theorem: Let G be a graph with $n > 1$ vertices. Then the following are equivalent:

(i) G is a tree.

(ii) G is a cycle-free and has $n - 1$ edges.

(iii) G is connected and has $n - 1$ edges.

This theorem also tells us that a finite tree T with n vertices must have $n-1$ edges. For example, the tree in Fig. 3*(a)* has 9 vertices and 8 edges, and the tree in Fig. 3*(b)* has 13 vertices and 12 edges.

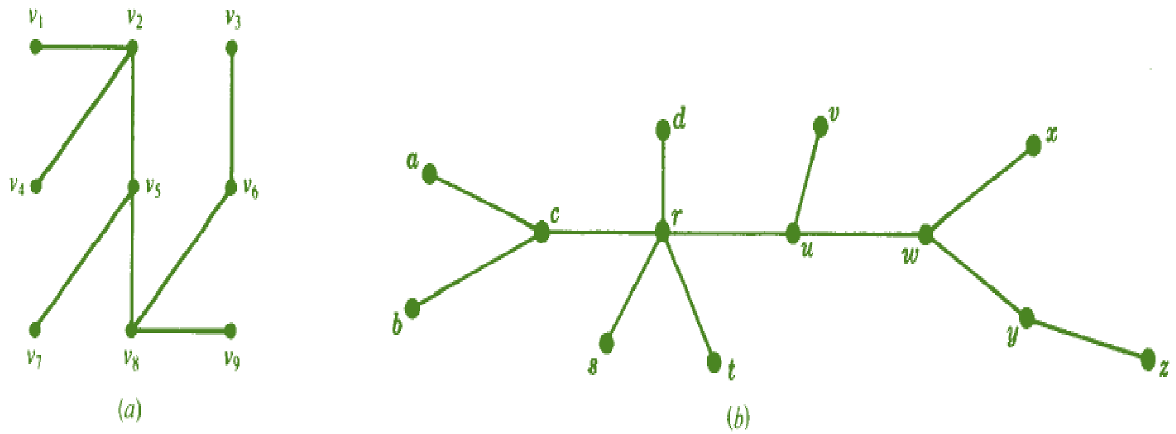


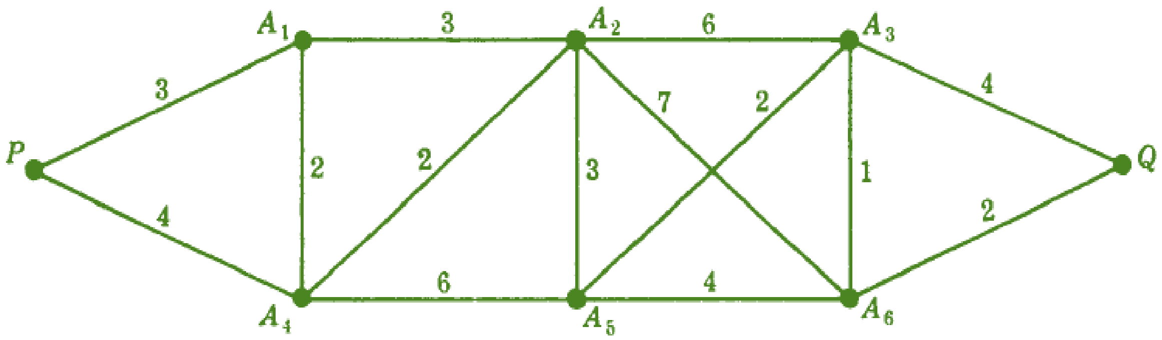
Figure 3

Labeled And weighted graphs:

Graph G is called a labelled graph if its edges and/or vertices are assigned data. If each edge (e) is assigned a non-negative number L(e). Then L(e) is called the weight or length of e. The weight of a path in such a weighted graph G is defined to be the sum of the weights of the edges in the path.

One important problem in graph theory is to find the shortest path, that is, a path of minimum weight (length), between any two given vertices.

Example 3: find the minimum path between P & Q:



(P, A1, A2, A5, A3, A6, Q)

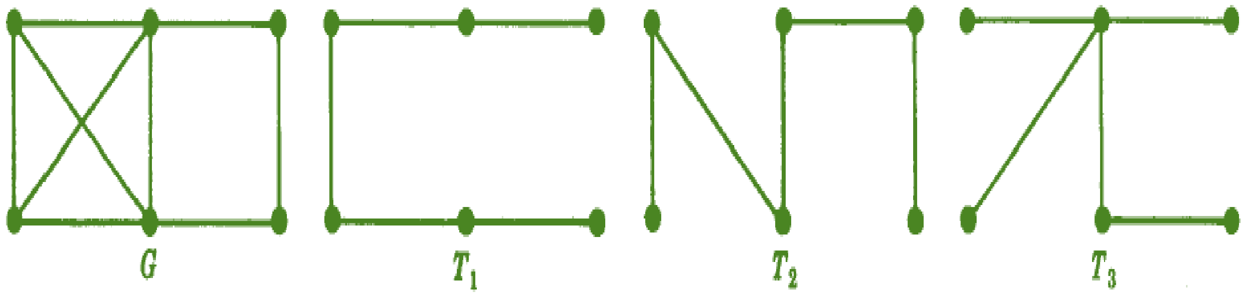
L (e) = 3 + 3 + 3 + 2 + 1 + 2 = 14

Another minimum path: (P, A4, A2, A5, A3, A6, Q)

L (e) = 4+ 2 + 3 + 2 + 1 + 2 = 14

Spanning Trees

A subgraph T of a connected graph G is called a spanning tree of G if T is a tree and T includes all the vertices of G.



Minimum Spanning Trees

Suppose G is a connected weighted graph. That is, each edge of G is assigned a nonnegative number called the weight of the edge. Then any spanning tree T of G is assigned a total weight obtained by adding the weights of the edges in T . A minimal spanning tree of G is a spanning tree whose total weight is as small as possible.

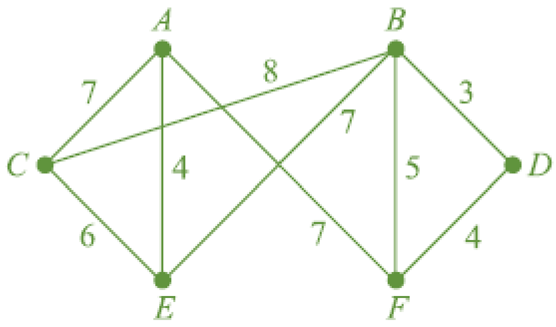
Algorithm 1 : The input is a connected weighted graph *G* with *n* vertices.

Step 1. Arrange the edges of *G* in the order of decreasing weights.

Step 2. Proceeding sequentially, delete each edge that does not disconnect the graph until $n - 1$ edges remain.

Step 3. Exit.

Example 4: Find a minimal spanning tree of the weighted graph Q, *note* that Q has six vertices, so a spanning tree will have five edges.

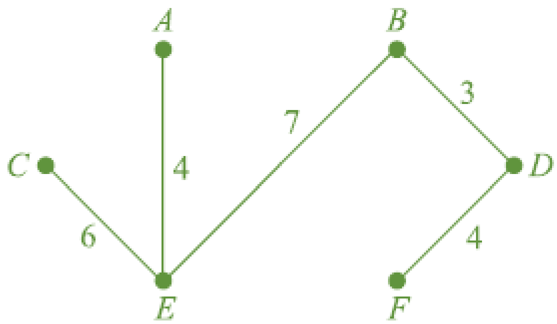


(*a*)

First, we order the edges by decreasing weights, and then we successively delete edges without disconnecting Q until five edges remain. This yields the following data:

Edges:	BC	AF	AC	BE	CE	BF	AE	DF	BD
Weight	8	7	7	7	6	5	4	4	3
Delete	Yes	Yes	Yes	No	No	Yes			

Thus, the minimal spanning tree of Q which is obtained contains the edges:
BE, CE, AE, DF, BD the spanning tree has a weight 24



(*b*)

Algorithm 2: (Kruskal): The input is a connected weighted graph *G* with *n* vertices.

Step 1. Arrange the edges of *G* in order of increasing weights.

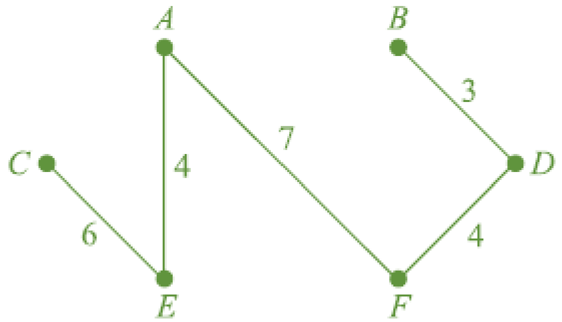
Step 2. Starting only with the vertices of *G* and proceeding sequentially, add each edge which does not result in a cycle until $n - 1$ edges are added.

Step 3. Exit.

First, we order the edges by increasing weights, and then we successively add edges without forming any cycles until five edges are included. This yields the following data:

Edges	BD	AE	DF	BF	CE	AC	AF	BE	BC
Weight	3	4	4	5	6	7	7	7	8
Add?	Yes	Yes	Yes	No	Yes	No	Yes		

Thus, the minimal spanning tree of Q which is obtained contains the edges:
BD, AE, DF, CE, AF
Observe that this spanning tree is not the same as the one obtained using Algorithm 1 as expected it also has a weight 24.



(c)

➢ REPRESENTING GRAPHS IN COMPUTER MEMORY:

There are many useful ways to represent graphs where in working with a graph it is helpful to be able to choose its most convenient representation.

(1) adjacency lists

EXAMPLE 5: Use adjacency lists to describe the simple graph given in Figure 4.

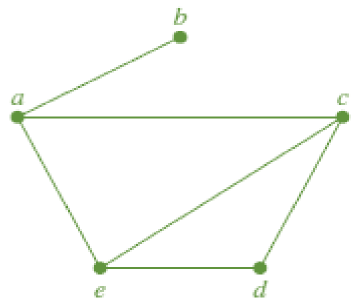


Figure 4

Solution*:* Table 1 lists those vertices adjacent to each of the vertices of the graph.

TABLE 1 An Adjacency List for a Simple Graph.	
Vertex	*Adjacent Vertices*
a	*b, c, e*
b	*a*
c	*a, d, e*
d	*c, e*
e	*a, c, d*

Example 6:
Represent the directed graph shown in Figure 5 by adjacency lists

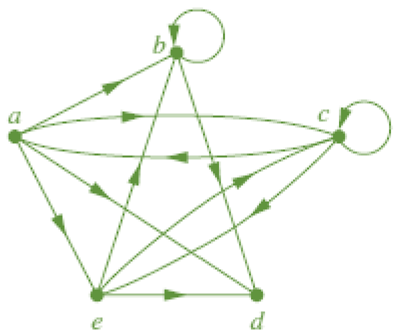


Figure 5

Solution: Table 2 represents the directed graph shown in Figure 5.

TABLE 2 An Adjacency List for a Directed Graph.	
Initial Vertex	*Terminal Vertices*
a	*b, c, d, e*
b	*b, d*
c	*a, c, e*
d	
e	*b, c, d*

(2) Adjacency Matrices

Carrying out graph algorithms using the representation of graphs by adjacency lists, can be cumbersome if there are many edges in the graph. To simplify computation, graphs can be represented using matrices. Two types of matrices commonly used to represent graphs will be presented here. One is based on the adjacency of vertices, and the other is based on incidence of vertices and edges.

Suppose that $G = (V, E)$ is a simple graph where $|V| = n$. The **adjacency matrix A** of G, is the n x n zero–one matrix with 1 as its (i, j)th entry when v_i and v_j are adjacent, and 0 as its *(i, j)*th entry when they are not adjacent.

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

Example 7: Use an adjacency matrix to represent the graph shown in Figure 6.

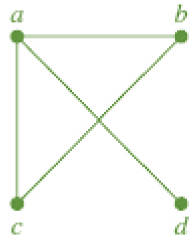


Figure 6

Solution:

We order the vertices as *a, b, c, d*. The matrix representing this graph is

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

Example 8: Draw a graph with the following adjacency matrix

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Solution: A graph with this adjacency matrix is shown in Figure 7.

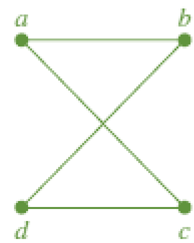


Figure 7

Adjacency matrices can also be used to represent undirected graphs with loops and with multiple edges. A loop at the vertex *vi* is represented by a 1 at the *(i, i)*th position of the adjacency matrix. When multiple edges connecting the same pair of vertices *vi* and *vj*, or multiple loops at the same vertex, are present, the adjacency matrix is no longer a zero–one matrix, because the *(i, j)*th entry of this matrix equals the number of edges that are associated to {*vi* , *vj* }.

Example 9: Use an adjacency matrix to represent the multigraph shown in Figure 8.

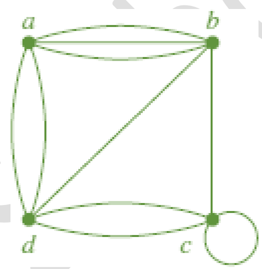


Figure 8

***Solution**:* The adjacency matrix using the ordering of vertices a, b, c, d is:

$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}.$$

The adjacency matrix for a directed graph does not have to be symmetric, because there may not be an edge from vj to vi when there is an edge from vi to vj .

Conditional

Many statements, especially in mathematics are of the form " $if\ p\ then\ q$ " such statements are called conditional statements and are denoted by $p \to q$. The truth value of the conditional statement $p \to q$ satisfies the following property:
- The conditional $p \to q$ is true unless p is true and q is false.
- The truth table of " $p \to q$ " can be written in the form.

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

Consider the conditional proposition $p \to q$ and other simple conditional propositions which contain p and q, i.e. $p \to q$, $\sim p \to \sim q\ and\ \sim q \to \sim p$, called respectively, the converse, inverse, and contra positive propositions. The truth table of these four propositions are as follows:

Example 10:

1. Let p: $Noor\ at\ home$.
2. q: $Noor\ answers\ the\ phone$.
3. $p \to q$: $If\ Noor\ at\ home\ then\ she\ will\ answer\ the\ phone$.
4. $q \to p$: $If\ Noor\ answered\ to\ the\ phone\ then\ she\ is\ at\ home$.
5. $\sim p \to \sim q$: $If\ Noor\ is\ not\ at\ home\ then\ she\ is\ not\ answer\ the\ phone$.
6. $\sim q \to \sim p$: $If\ Noor\ is\ not\ answer\ the\ phone\ then\ she\ is\ not\ at\ home$.

Biconditional

Another common statement is of the form " $p\ if\ and\ only\ if\ q$ " or simply, " $p\ iff\ q$ ". Such statements are called biconditional statements and are denoted by $p \leftrightarrow q$.

- ✓ The truth value of the biconditional statement $p \leftrightarrow q$ satisfies the property:
- ✓ If $p\ and\ q$ have the same truth value, then " $p \leftrightarrow q$ " is true,

- ✓ If p *and* q have an opposite truth value, then " $p \leftrightarrow q$ " is false.

Example 11:

Consider the following statements

1. Paris is in France iff 2 + 2 = 5.
2. Paris is in England iff 2 + 2 = 4.
3. Paris is in France iff 2 + 2 = 4.
4. Paris is in England iff 2 + 2 = 5.

According, (3) and (4) are true while (1) and (2) are false.
The truth table is written as follows

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Logical Equivalence

Two statements are said to be logically equivalent if their truth table are identical. We denote the logical equivalent of p *and* q by " ≡ ".

Example 12:

The truth tables of $(p \to q) \wedge (q \to p)$ *and* $p \leftrightarrow q$ are as follows:

p	q	$p \to q$	$q \to p$	$p \to q \wedge (q \to p)$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

p	q	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Hence, $(p \to q) \wedge (q \to p) \equiv p \leftrightarrow q$

Example 12:
The truth tables below show that $p \to q$ *and* $\sim p \vee q$ are logically equivalent, i.e. $p \to q \equiv \sim p \vee q$

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

p	q	$\sim p$	$\sim p \vee q$
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

1. Show that $\sim(p \wedge q) \equiv \sim p \wedge \sim q$

Solution:

p	q	~(p ∧ q)	~p	~q	~(p ∧ q) ∧ (~p ∧ ~q)
T	T	F	F	F	F
T	F	T	F	T	T
F	T	T	T	F	T

F	F	T	T	T	T

The truth tables match, so $\neg(p \wedge q) \equiv \neg p \vee \neg q$.

2. **S**how that $\sim(p \rightarrow q) \equiv p \wedge \sim q$

Solution:

p	q	p→q	~(p→q)	~q	p∧~q
T	T	T	F	F	F
T	F	F	T	T	T
F	T	T	F	F	F
F	F	T	F	T	F

The truth tables match, so $\neg(p \rightarrow q) \equiv p \wedge \neg q$.