Discrete Structures
1st Stage

Dr. Alaa Jarah

Lec_ 1

Propositional Logic

Introduction

Logic is the basis of all mathematical reasoning. It has practical applications in areas of computer science as well as to many other fields of study. In mathematics, we must understand what makes up a correct mathematical argument, that is, a proof. Once we prove that a mathematical statement is true, we call it a theorem. A collection of theorems on a topic organize what we know about this topic. To learn a mathematical topic, a person needs to actively construct mathematical arguments on this topic. Moreover, knowing the proof of a theorem often makes it possible to modify the result to fit new situations. Everyone knows that proofs are important throughout mathematics. The rules of logic give precise meaning to mathematical statements. These rules are used to distinguish between valid and invalid mathematical arguments. In this chapter, we will explain what makes up a correct mathematical argument and introduce tools to construct these arguments. These basic tools will help us to develop different proof methods that will enable us to prove many different types of results in the later chapters.

1.1 Basic Concepts in Logic

Our discussion begins with an introduction to the basic building blocks of logic viz., propositions.

Definition 1.1. *A* ***proposition*** *is a declarative sentence (that is, a sentence that declares a fact) that is either true or false, but not both.*

All the following declarative sentences are propositions.

1. New Delhi, is the capital of India.
2. $2+1=3$.
3. $2+1=2$.

Here propositions 1 and 2 are true, whereas 3 is false.
Some sentences that are not propositions are:

1. How are you?
2. Read this carefully.
3. $x+1=2$.
4. $x+y=z$.

Sentences 1 and 2 are not propositions because they are not declarative sentences. Sentences 3 and 4 are not propositions because they are neither true nor false. Note that each of the sentences 3 and 4 can be turned into a proposition if we assign values to the variables.

We use letters to denote **propositional variables (or statement variables)**, that is, variables that represent propositions, just as letters are used to denote numerical variables. The conventional letters used for propositional variables are $p, q, r, s, \ldots$. The **truth value** of a proposition is true, denoted by **T**, if it is a true proposition, and the truth value of a proposition is false, denoted by **F**, if it is a false proposition.

Definition 1.2. *The area of logic that deals with propositions is called the **propositional calculus** or **propositional logic**.*

Propositional calculus was first developed systematically by the greek philosopher Aristotle more than 2300 years ago.

Definition 1.3. ***Compound propositions** are new propositions formed from existing propositions using logical operators.*

Definition 1.4. *Let p be a proposition. The **negation** of p, denoted by $\neg p$, is the statement "It is not the case that p." The proposition $\neg p$ is read "not p." The truth value of the negation of $p, \neg p$, is the opposite of the truth value of p.*

The negation operator constructs a new proposition from a single existing proposition. We will now introduce the logical operators that are used to form new propositions from two or more existing propositions. These logical operators are also called connectives.

p	$\neg p$
T	F
F	T

Table 1.1: Negation

Definition 1.5. *Let p and q be propositions. The **conjunction** of p and q, denoted by $p \wedge q$, is the proposition "p and q." The conjunction $p \wedge q$ is true when both p and q are true and is false otherwise.*

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

AND

Table 1.2: Conjunction

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

OR

Table 1.3: Disjunction

Table 1.2 displays the truth table of $p \wedge q$. This table has a row for each of the four possible combinations of truth values of p and q. The four rows correspond to the pairs of truth values TT, TF, FT, and FF, where the first truth value in the pair is the truth value of p and the second truth value is the truth value of q. Note that in logic the word "but" sometimes is used instead of "and" in a conjunction. For example, the statement "The sun is shining, but it is raining" is another way of saying "The sun is shining and it is raining."

Definition 1.6. *Let p and q be propositions. The **disjunction** of p and q, denoted by $p \vee q$, is the proposition "p or q." The disjunction $p \vee q$ is false when both p and q are false and is true otherwise.*

Table 1.3 displays the truth table for $p \vee q$. The use of the connective "or" in a disjunction corresponds to one of the two ways the word "or" is used in English, namely, in an inclusive way. Thus, a disjunction is true when at least one of the two propositions in it is true. Sometimes, we use "or" in an exclusive sense. When the "exclusive or" is used to connect the propositions p and q, the proposition "p or q (but not both)" is obtained.

Definition 1.7. *Let p and q be propositions. The **exclusive or** of p and q, denoted by $p \oplus q$, is the proposition that is true when exactly one of p and q is true and is false otherwise.*

The truth table for the exclusive or of two propositions is displayed in Table 1.4.

Definition 1.8. *Let p and q be propositions. The **conditional statement** $p \to q$ is the proposition "if p, then q." The conditional statement $p \to q$ is false when p is true and q is false, and true otherwise.*

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Table 1.4: Exclusive or

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

Table 1.5: Conditional Statement

In the conditional statement $p \to q$, p is called the hypothesis (or antecedent or premise) and q is called the conclusion (or consequence). The statement $p \to q$ is called a conditional statement because $p \to q$ asserts that q is true on the condition that p holds. A conditional statement is also called an **implication**. The truth table for the conditional statement $p \to q$ is shown in Table 1.5. Note that the statement $p \to q$ is true when both p and q are true and when p is false (no matter what truth value q has).
Because conditional statements play such an essential role in mathematical reasoning, a variety of terminology is used to express $p \to q$. A useful way to understand the truth value of a conditional statement is to think of an obligation or a contract. For example, a pledge many politicians make when running for office is "If I am elected, then I will lower taxes." If the politician is elected but does not lowers taxes, then and then only the voters can say that the politician has broken the campaign pledge. This scenario corresponds to the case when p is true but q is false in $p \to q$.

You will encounter most if not all of the following ways to express this conditional statement:

"if p, then q"	"p implies q"
"if p, q"	"p only if q"
"p is sufficient for q"	"a sufficient condition for q is p"
"q if p"	"q whenever p"
"q when p"	"q is necessary for p"
"q unless $\neg p$"	"q follows from p"
"a necessary condition for p is q".	

Example 1.1. Let p be the statement "Ali learns discrete mathematics" and q the statement "Ali will find a good job." Express the statement $p \to q$ as a statement in English.
Solution: From the definition of conditional statements, we see that when
p is the statement "Ali learns discrete mathematics"
q is the statement "Ali will find a good job,"
$p \to q$ represents the statement "If Ali learns discrete mathematics, then he will find a good job."

There are many other ways to express this conditional statement in English. Among the most natural of these are:
"Ali will find a good job when Ali learns discrete mathematics."
"For Ali to get a good job, it is sufficient for him to learn discrete mathematics." and
"Ali will find a good job unless Ali does not learn discrete mathematics." and so on.

1.1.1 Converse, Contrapositive, and Inverse

We can form some new conditional statements starting with a conditional statement $p \to q$. In particular, there are three related conditional statements that occur so often that they have special names.

Definition 1.9. *The proposition $q \to p$ is called the **converse** of $p \to q$. The **contrapositive** of $p \to q$ is the proposition $\neg q \to \neg p$. The proposition $\neg p \to \neg q$ is called the **inverse** of $p \to q$.*

From the truth table we can easily check that the truth values of $p \to q$ and $\neg q \to \neg p$ are same. This leads us to the next definition.

Definition 1.10. *When two compound propositions always have the same truth value we call them **equivalent**.*

The converse and the inverse of a conditional statement are also equivalent.

Example 1.2. What are the contrapositive, the converse, and the inverse of the conditional statement "The home team wins whenever it is raining?"
Solution: Because "q whenever p" is one of the ways to express the conditional statement $p \to q$, the original statement can be rewritten as "If it is raining, then the

home team wins." Consequently, the contrapositive of this conditional statement is "If the home team does not win, then it is not raining." The converse is "If the home team wins, then it is raining." The inverse is "If it is not raining, then the home team does not win." Only the contrapositive is equivalent to the original statement. It can be easily verified by the truth table.

We now introduce another way to combine propositions that expresses that two propositions have the same truth value.

Definition 1.11. *Let p and q be propositions. The **biconditional statement** $p \leftrightarrow q$ is the proposition "p if and only if q." The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise. Biconditional statements are also called **bi-implications**.*

The truth table for $p \leftrightarrow q$ is shown in Table 1.6. There are some other common ways to express $p \leftrightarrow q$: "p is necessary and sufficient for q", "if p then q, and conversely", "p iff q." The last way of expressing the biconditional statement $p \leftrightarrow q$ uses the abbreviation "iff" for "if and only if." Note that $p \leftrightarrow q$ has exactly the same truth value as $(p \to q) \wedge (q \to p)$. We have now introduced four important

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

X-NOR

Table 1.6: Biconditional Statement

logical connectives–conjunctions, disjunctions, conditional statements, and biconditional statements–as well as negations. We can use these connectives to build up complicated compound propositions involving any number of propositional variables. We can use truth tables to determine the truth values of these compound propositions. We use a separate column to find the truth value of each compound expression that occurs in the compound proposition as it is built up. The truth values of the compound proposition for each combination of truth values of the propositional variables in it is found in the final column of the table.

Example 1.3. Construct the truth table of the compound proposition $(p \vee \neg q) \to (p \wedge q)$.

Solution: Because this truth table involves two propositional variables p and q, there are four rows in this truth table, one for each of the pairs of truth values TT, TF, FT, and FF. The first two columns are used for the truth values of p and q, respectively. In the third column we find the truth value of $\neg q$, needed to find the truth value of $p \vee \neg q$, found in the fourth column. The fifth column gives the truth value of $p \wedge q$. Finally, the truth value of $(p \vee \neg q) \rightarrow (p \wedge q)$ is found in the last column. The resulting truth table is shown in Table 1.7.

p	q	$\neg q$	$p \vee \neg q$	$p \wedge q$	$(p \vee \neg q) \rightarrow (p \wedge q)$
T	T	F	T	T	T
T	F	T	T	F	F
F	T	F	F	F	T
F	F	T	T	F	F

Table 1.7: The Truth table

1.1.2 Precedence of Logical Operators

We can construct compound propositions using the negation operator and the logical operators defined so far. We will generally use parentheses to specify the order in which logical operators in a compound proposition are to be applied. For instance, $(p \vee q) \wedge (\neg r)$ is the conjunction of $p \vee q$ and $\neg r$. However, to reduce the number of parentheses, we specify that the negation operator is applied before all other logical operators. This means that $\neg p \wedge q$ is the conjunction of $\neg p$ and q, namely, $(\neg p) \wedge q$, not the negation of the conjunction of p and q, namely $\neg(p \wedge q)$.
Another general rule of precedence is that the conjunction operator takes precedence over the disjunction operator, so that $p \vee q \wedge r$ means $p \vee (q \wedge r)$ rather than $(p \vee q) \wedge r$. Because this rule may be difficult to remember, we will continue to use parentheses so that the order of the disjunction and conjunction operators is clear. Table 1.8 displays the precedence levels of the logical operators.

Operator	**Precedence**
$\neg$	1
$\wedge$	2
$\vee$	3
$\rightarrow$	4
$\leftrightarrow$	5

Table 1.8: Precedence of Logical Operators

Discrete Structures
1st Stage

Dr_Alaa Jarah

Lec_ 2

1.2 Propositional Equivalences

An important type of step used in a mathematical argument is the replacement of a statement with another statement with the same truth value. Because of this, methods that produce propositions with the same truth value as a given compound proposition are used extensively in the construction of mathematical arguments. Note that we will use the term "compound proposition" to refer to an expression formed from propositional variables using logical operators, such as $p \wedge q$. We begin our discussion with a classification of compound propositions according to their possible truth values.

Definition 1.12. *A compound proposition that is always true, no matter what the truth values of the propositional variables that occur in it, is called a* ***tautology****. A compound proposition that is always false is called a* ***contradiction****. A compound proposition that is neither a tautology nor a contradiction is called a* ***contingency****.*

Tautologies and contradictions are often important in mathematical reasoning. An example of a tautology is $p \vee \neg p$ whereas an example for a contradiction is $p \wedge \neg p$. The following truth table illustrates this.

p	$\neg p$	$p \vee \neg p$	$p \wedge \neg p$
T	F	T	F
F	T	T	F

Table 1.9: Examples of Tautology and Contradiction

1.2.1 Logical Equivalences

Compound propositions that have the same truth values in all possible cases are called logically equivalent. We can also define the notion as follows.

Definition 1.13. *The compound propositions p and q are called* ***logically equivalent*** *if $p \leftrightarrow q$ is a tautology. The notation $p \equiv q$ denotes that p and q are logically equivalent.*

One way to determine whether two compound propositions are equivalent is to use a truth table.

Example 1.4. Show that $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are logically equivalent.
Solution: The truth tables for these compound propositions are displayed in Table 1.10. Because the truth values of the compound propositions $\neg(p \vee q)$ and $\neg p \wedge \neg q$ agree for all possible combinations of the truth values of p and q, it follows that $\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$ is a tautology and that these compound propositions are logically equivalent.

p	q	$p \vee q$	$\neg(p \vee q)$	$\neg p$	$\neg q$	$\neg p \wedge \neg q$	$\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q)$
T	T	T	F	F	F	F	T
T	F	T	F	F	T	F	T
F	T	T	F	T	F	F	T
F	F	F	T	T	T	T	T

Table 1.10: The truth table

Example 1.5. Show that $\boxed{p \to q \text{ and } \neg p \vee q}$ are logically equivalent.
Solution: We construct the truth table for these compound propositions in Table 1.11. Because the truth values of $\neg p \vee q$ and $p \to q$ agree, they are logically equivalent.

p	q	$\neg p$	$\neg p \vee q$	$p \to q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Table 1.11: The truth table

We will now establish a logical equivalence of two compound propositions involving three different propositional variables p, q, and r. To use a truth table to establish such a logical equivalence, we need eight rows, one for each possible combination of truth values of these three variables. In general, 2^n rows are required in the truth table to establish logical equivalence involving n propositional variables.

Table 1.12 demonstrates that $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ are logically equivalent.

Example 1.6. Show that $\boxed{p \vee (q \wedge r) \text{ and } (p \vee q) \wedge (p \vee r)}$ are logically equivalent. This is the distributive law of disjunction over conjunction.
Solution: We construct the truth table for these compound propositions in Table

$p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$

p	q	r	$q \wedge r$	$p \vee (q \wedge r)$	$p \vee q$	$p \vee r$	$(p \vee q) \wedge (p \vee r)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

Table 1.12: The truth table

1.12. Because the truth values of $p \vee (q \wedge r)$ and $(p \vee q) \wedge (p \vee r)$ agree, these compound propositions are logically equivalent.

Equivalence	Name	Equivalence	Name
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws	$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws	$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan's laws
$\neg(\neg p) \equiv p$	Double negation law	$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws	$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

Table 1.13: Logical Equivalences

Table 1.13 contains some important equivalences. In these equivalences, **T** denotes the compound proposition that is always true and **F** denotes the compound proposition that is always false. Note that $p_1 \vee p_2 \vee \ldots \vee p_n$ and $p_1 \wedge p_2 \wedge \ldots \wedge p_n$ are well defined whenever $p_1, p_2, \ldots, p_n$ are propositions. Also De Morgan's laws extend to

$$\neg(p_1 \vee p_2 \vee \ldots \vee p_n) \equiv (\neg p_1 \wedge \neg p_2 \wedge \ldots \wedge \neg p_n)$$

and

$$\neg(p_1 \wedge p_2 \wedge \ldots \wedge p_n) \equiv (\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n).$$

Example 1.7. Show that $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent by developing a series of logical equivalences.

Solution: We will use one of the equivalences in Table 1.13 at a time, starting with $\neg(p \vee (\neg p \wedge q))$ and ending with $\neg p \wedge \neg q$. We have the following equivalences.

Equivalence	
$p \to q \equiv \neg p \vee q$ $p \to q \equiv \neg q \to \neg p$ $p \vee q \equiv \neg p \to q$ $p \wedge q \equiv \neg(p \to \neg q)$ $\neg(p \to q) \equiv p \wedge \neg q$ $(p \to q) \wedge (p \to r) \equiv p \to (q \wedge r)$ $(p \to r) \wedge (q \to r) \equiv (p \vee q) \to r$ $(p \to q) \vee (p \to r) \equiv p \to (q \vee r)$ $(p \to r) \vee (q \to r) \equiv (p \wedge q) \to r$	Logical Equivalences Involving Conditional Statements
$p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$ $p \leftrightarrow q \equiv \neg p \leftrightarrow \neg q$ $p \leftrightarrow q \equiv (p \wedge q) \vee (\neg p \wedge \neg q)$ $\neg(p \leftrightarrow q) \equiv p \leftrightarrow \neg q$	Logical Equivalences Involving Biconditional Statements

Table 1.14: Logical Equivalences

$$\begin{aligned}
\neg(p \vee (\neg p \wedge q)) &\equiv \neg p \wedge \neg(\neg p \wedge q) && \text{by the second De Morgan law}\\
&\equiv \neg p \wedge [\neg(\neg p) \vee \neg q] && \text{by the first De Morgan law}\\
&\equiv \neg p \wedge (p \vee \neg q) && \text{by the double negation law}\\
&\equiv (\neg p \wedge p) \vee (\neg p \wedge \neg q) && \text{by the second distributive law}\\
&\equiv \mathbf{F} \vee (\neg p \wedge \neg q) && \text{by the second negation law}\\
&\equiv (\neg p \wedge \neg q) \vee \mathbf{F} && \text{by the commutative law}\\
& && \text{for disjunction}\\
&\equiv \neg p \wedge \neg q && \text{by the identity law for } \mathbf{F}
\end{aligned}$$

Consequently $\neg(p \vee (\neg p \wedge q))$ and $\neg p \wedge \neg q$ are logically equivalent.

Logical equivalences involving conditional statements and biconditional statements are given in the table 1.14. These equivalences are important as they form basic tools for proving theorems. Few theorems involve “if and only if” $p \leftrightarrow q$. To prove the theorem of this type we use the equivalence $p \leftrightarrow q \equiv (p \to q) \wedge (q \to p)$. So it is enough to prove the statements $p \to q$ and $q \to p$ separately.

Remark 1.1. *A logical equivalence can be proved by using either a truth table or by using a chain of known logical equivalences. Also a tautology can be proved by using either a truth table or by using logical equivalences.*

Example 1.8. Show that $(p \wedge q) \to (p \vee q)$ is a tautology.

Solution: To show that this statement is a tautology, we will use logical equivalences to demonstrate that it is logically equivalent to **T**.

$$\begin{aligned}
(p \wedge q) \rightarrow (p \vee q) &\equiv \neg(p \wedge q) \vee (p \vee q) && \text{since } p \rightarrow q \equiv \neg p \vee q \\
&\equiv (\neg p \vee \neg q) \vee (p \vee q) && \text{by the first De Morgan law} \\
&\equiv (\neg p \vee p) \vee (\neg q \vee q) && \text{by the associative and comm-} \\
& && \text{utative laws for disjunction} \\
&\equiv \mathbf{T} \vee \mathbf{T} && \text{by the commutative and} \\
& && \text{negation laws for disjunction.} \\
&\equiv \mathbf{T} && \text{by the domination law}
\end{aligned}$$

Thus we have shown that $(p \wedge q) \rightarrow (p \vee q)$ is a tautology.
(Note: This could also be done using a truth table.)

Logic has practical applications to the design of computing machines, to the specification of systems, to artificial intelligence, to computer programming, to programming languages, and to other areas of computer science, as well as to many other fields of study. In the next chapter we will introduce the concepts which will help us to express the meaning of statements in mathematics and natural language.

Discrete Structures
1st Stage

By Dr_Alaa Jarah

Lec_3

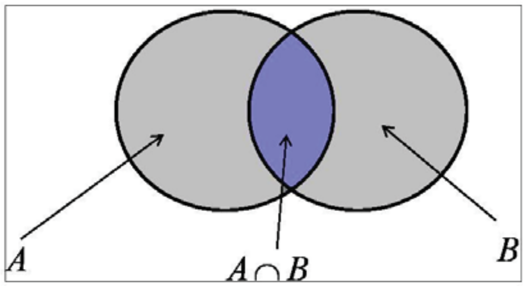
The Inclusion and Exclusion Principle

Introduction

The principle of Inclusion and Exclusion is doubtless very old; its origin is probably untraceable. The principle of Inclusion and Exclusion is sometimes referred to as "Poincare's Theorem". J. J. Sylvester and Danial da Silva are the two mathematicians associated with the combinatorial form of the principle.
The principle of Inclusion and Exclusion is a way of thinking about combining sets with overlapping elements.

3.1 The Subtraction Rule

If a task can be done in either n_1 ways or n_2 ways, then the number of ways to do the task is $n_1 + n_2$ minus the number of ways to do the task that are common to the two different ways.
The subtraction rule is also known as the principle of inclusion-exclusion, especially when it is used to count the number of elements in the union of two sets. Suppose that A_1 and A_2 are sets. Then, there are $|A_1|$ ways to select an element from A_1 and $|A_2|$ ways to select an element from A_2. The number of ways to select an element from A_1 or from A_2, that is, the number of ways to select an element from their union, is the sum of the number of ways to select an element from A_1 and the number of ways to select an element from A_2, minus the number of ways to select an element that is in both A_1 and A_2. Because there are $|A_1 \cup A_2|$ ways to select an element in either A_1 or in A_2, and $|A_1 \cap A_2|$ ways to select an element common to both sets, we have $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|$.

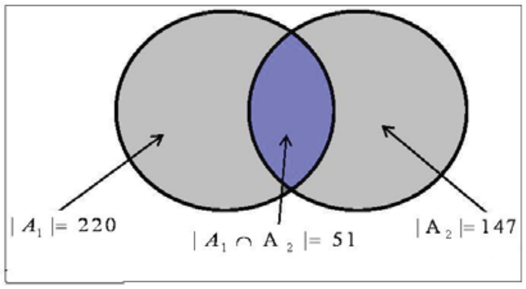


Example 3.1. How many bit strings of length eight either start with a 1 bit or end with the two bits 00?

Solution: We can construct a bit string of length eight that either starts with a 1 bit or ends with the two bits 00, by constructing a bit string of length eight beginning with a 1 bit or by constructing a bit string of length eight that ends with the two bits 00. We can construct a bit string of length eight that begins with a 1 in $2^7 = 128$ ways. This follows by the product rule, because the first bit can be chosen in only one way and each of the other seven bits can be chosen in two ways. Similarly, we can construct a bit string of length eight ending with the two bits 00, in $2^6 = 64$ ways. This follows by the product rule, because each of the first six bits can be chosen in two ways and the last two bits can be chosen in only one way. Some of the ways to construct a bit string of length eight starting with a 1 are the same as the ways to construct a bit string of length eight that ends with the two bits 00. There are $2^5 = 32$ ways to construct such a string. This follows by the product rule, because the first bit can be chosen in only one way, each of the second through the sixth bits can be chosen in two ways, and the last two bits can be chosen in one way. Consequently, the number of bit strings of length eight that begin with a 1 or end with a 00, which equals the number of ways to construct a bit string of length eight that begins with a 1 or that ends with 00, equals $128 + 64 - 32 = 160$.

Example 3.2. A computer company receives 350 applications from graduates for a job. Suppose that 220 of these applicants majored in computer science, 147 majored in business, and 51 majored both in computer science and in business. How many of these applicants majored neither in computer science nor in business?

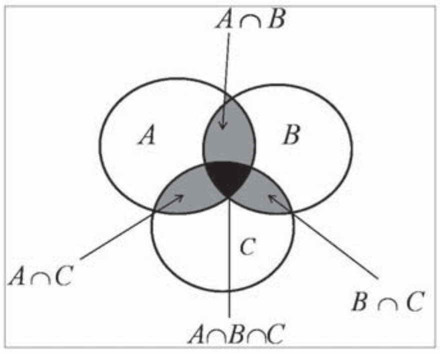
Solution: To find the number of these applicants who majored neither in computer science nor in business, we can subtract the number of students who majored either



in computer science or in business (or both) from the total number of applicants. Let A_1 be the set of students who majored in computer science and A_2 the set of students who majored in business. Then $A_1 \cup A_2$ is the set of students who majored in computer science or business (or both), and $A_1 \cap A_2$ is the set of students who majored both in computer science and in business. By the subtraction rule the number of students who majored either in computer science or in business (or both) equals $|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2| = 220 + 147 - 51 = 316$. We conclude that $350 - 316 = 34$ of the applicants majored neither in computer science nor in business.

3.2 The Principle of Inclusion and Exclusion

The principle of Inclusion and Exclusion, hereafter called PIE, gives a formula for the size of the union of n finite sets. We assume that the universe is finite. It is a generalization of the familiar formulas $|A \cup B| = |A| + |B| - |A \cap B|$ and $\boxed{|A \cup B \cup C|} = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$.



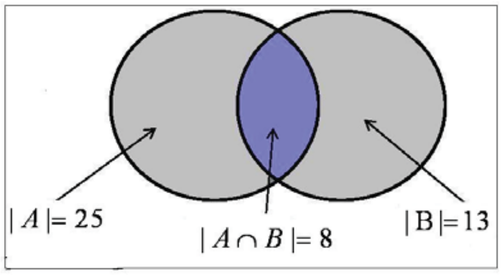
Theorem 3.1. *If $P_1, P_2, \ldots, P_n$ be finite sets, then*
$|P_1 \cup P_2 \cup \ldots \cup P_n| = |P_1| + |P_2| + \ldots + |P_n| - |P_1 \cap P_2| - |P_1 \cap P_3| - \ldots |P_{n-1} \cap P_n| +$
$|P_1 \cap P_2 \cap P_3| + |P_1 \cap P_2 \cap P_4| + \ldots + |P_{n-2} \cap P_{n-1} \cap P_n| - \ldots + (-1)^{n+1}|P_1 \cap P_2 \cap \ldots \cap P_n|$
That is

$$|P_1 \cup P_2 \cup \ldots \cup P_n| = \sum_{1 \le i \le n} |P_i| - \sum_{1 \le i < j \le n} |P_i \cap P_j|$$
$$+ \sum_{1 \le i < j < k \le n} |P_i \cap P_j \cap P_k| - \ldots + (-1)^{n+1}|P_1 \cap P_2 \cap ... \cap P_n|$$
.

That is, the cardinality of the union $P_1 \cup P_2 \cup ... \cup P_n$ can be calculated by including (adding) the sizes of all of the sets together, then excluding (subtracting) the sizes of the intersections of all pairs of sets, then including the sizes of the intersections of all triples, excluding the sizes of the intersections of all quadruples, and so on until, finally, the size of the intersection of all of the sets has been included or excluded, as appropriate. If n is odd it is included, and if n is even it is excluded.
It is important to remember that all sets involved must be finite.
We should try to use PIE when we are trying to count something described by a bunch of conditions, any number of which might hold at the same time. Often PIE is used in conjunction with counting the complement. That is, you use it to count the number of objects in the universe that you do not want, and subtract this from the size of the universe (which needs to be finite). In applying PIE, the setup is of great importance. You need to be clear about what the sets are (what it means to belong to one or more of them), what the universe is, and how the principle gives you what you want. Once you have done this, things often reduce to more or less straightforward counting problems.
Note: In PIE, for n sets there are $\binom{n}{1}$ sums of one sets, there are $\binom{n}{2}$ sums of intersection of two sets, there are $\binom{n}{3}$ sums of intersections of three sets,...,there are $\binom{n}{n}$ sums of intersections of n sets.If n=4 then there are $\binom{4}{1} = 4$ sums of one sets, $\binom{4}{2} = 6$ sums of intersections of two sets, $\binom{4}{3} = 4$ sums of intersection of three sets and $\binom{4}{4} = 1$ sum of intersections of three sets.

Example 3.3. In a discrete mathematics class every student is a major in computer science or mathematics, or both. The number of students having computer science as a major (possibly along with mathematics) is 25; the number of students having mathematics as a major (possibly along with computer science) is 13; and the number of students majoring in both computer science and mathematics is 8. How many students are in this class?

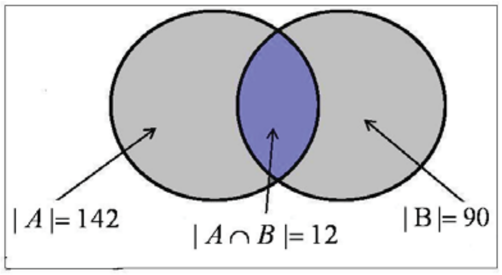
Solution: Let A be the set of students in the class majoring in computer science



and B be the set of students in the class majoring in mathematics. Then $A \cap B$ is the set of students in the class who are joint mathematics and computer science majors. Because every student in the class is majoring in either computer science or mathematics (or both), it follows that the number of students in the class is $|A \cup B|$. Therefore, $|A \cup B| = |A| + |B| - |A \cap B| = 25 + 13 - 8 = 30$. Therefore, there are 30 students in the class.

Example 3.4. How many positive integers not exceeding 1000 are divisible by 7 or 11?

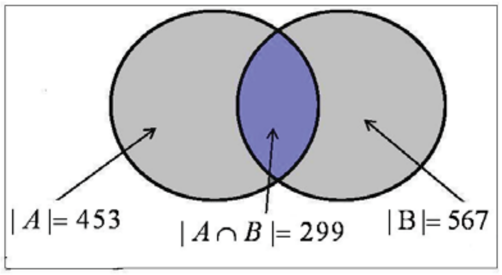
Solution: Let A be the set of positive integers not exceeding 1000 that are divisible



by 7, and let B be the set of positive integers not exceeding 1000 that are divisible by 11. Then $A \cup B$ is the set of integers not exceeding 1000 that are divisible by either 7 or 11, and $A \cap B$ is the set of integers not exceeding 1000 that are divisible by both 7 and 11. We know that among the positive integers not exceeding 1000 there are $\left\lfloor \frac{1000}{7} \right\rfloor$ integers divisible by 7 and $\left\lfloor \frac{1000}{11} \right\rfloor$ divisible by 11. Because 7 and 11 are relatively prime, the integers divisible by both 7 and 11 are those divisible by 7×11. Consequently, there are $\left\lfloor \frac{1000}{11 \times 7} \right\rfloor$ positive integers not exceeding 1000 that are divisible by both 7 and 11. It follows that there are $|A \cup B| = |A| + |B| - |A \cap B| = 142 + 90 - 12 = 220$ positive integers not exceeding 1000 that are divisible by either 7 or 11.

Example 3.5. Suppose that there are 1807 students in first year at your college. Of these, 453 are taking a course in computer science, 567 are taking a course in mathematics, and 299 are taking courses in both computer science and mathematics. How many are not taking a course either in computer science or in mathematics?

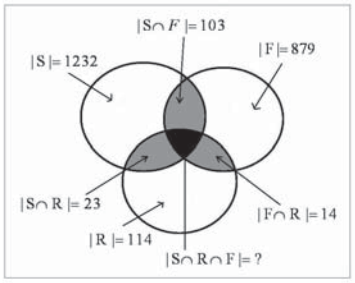
Solution: To find the number of first year students who are not taking a course



in either mathematics or computer science, subtract the number that are taking a course in either of these subjects from the total number of first year students. Let A be the set of all first year students taking a course in computer science, and let B be the set of all first year students taking a course in mathematics. It follows that $|A| = 453$, $|B| = 567$, and $|A \cap B| = 299$. The number of first year students taking a course in either computer science or mathematics is $|A \cup B| = |A| + |B| - |A \cap B| = 453 + 567 - 299 = 721$. Consequently, there are $1807 - 721 = 1086$ first year students who are not taking a course in computer science or mathematics.

Example 3.6. A total of 1232 students have taken a course in Spanish, 879 have taken a course in French, and 114 have taken a course in Russian. Further, 103 have taken courses in both Spanish and French, 23 have taken courses in both Spanish and Russian, and 14 have taken courses in both French and Russian. If 2092 students have taken at least one of Spanish, French, and Russian, how many students have taken a course in all three languages?

Solution: Let S be the set of students who have taken a course in Spanish, F the set of students who have taken a course in French, and R the set of students who have taken a course in Russian. Then $|S| = 1232$, $|F| = 879$, $|R| = 114$, $|S \cap F| = 103$, $|S \cap R| = 23$, $|F \cap R| = 14$, and $|S \cup F \cup R| = 2092$. When we insert these



quantities into the equation $|S \cup F \cup R| = |S| + |F| + |R| - |S \cap F| - |S \cap R| - |F \cap R| + |S \cap F \cap R|$ we obtain $2092 = 1232 + 879 + 114 - 103 - 23 - 14 + |S \cap F \cap R|$. We now solve for $|S \cap F \cap R|$. We find that $|S \cap F \cap R| = 7$. Therefore, there are seven students who have taken courses in Spanish, French, and Russian.

Example 3.7. At your college, there are $20, 30, 25$, and 43 students have taken languages Marathi, Hindi, English, Sanskrit respectively; there are 10 students who have taken Marathi and Hindi; 9 students who have taken Hindi and English; 13 students who have taken Marathi and Sanskrit; 18 students who have taken Marathi and English; 25 students who have taken Hindi and Sanskrit; 17 students who have taken English and Sanskrit; 5 students who have taken Marathi, Hindi, English; 3 students who have taken Marathi, Hindi, Sanskrit; 2 students who have taken Hindi, English, Sanskrit; 2 students who have taken Marathi, English, Sanskrit; 1 student who has taken all four languages. How many students are enrolled in languages either Marathi or Hindi or English or Sanskrit?

Solution: Let M be the set of students who have taken Marathi, H the set of students who have taken Hindi, E the set of students who have taken English, S the set of students who have taken Sanskrit. Then $|M| = 20$, $|H| = 30$, $|E| = 25$, $|S| = 43$, $|M \cap H| = 10$, $|H \cap E| = 9$, $|M \cap S| = 13$, $|M \cap E| = 18$, $|H \cap S| = 25$, $|E \cap S| = 17$, $|M \cap H \cap E| = 5$, $|M \cap H \cap S| = 3$, $|M \cap E \cap S| = 2$, $|H \cap E \cap S| = 2$, $|M \cap H \cap E \cap S| = 1$. When we insert these quantities into the equation $|M \cup H \cup E \cup S| = |M| + |H| + |E| + |S| - |M \cap H| - |H \cap E| - |M \cap S| - |M \cap E| - |H \cap S| - |E \cap S| + |M \cap H \cap E| + |M \cap H \cap S| + |M \cap E \cap S| + |H \cap E \cap S| - |M \cap H \cap E \cap S| = 20 + 30 + 25 + 43 - 10 - 9 - 13 - 18 - 25 - 17 + 5 + 3 + 2 + 2 - 1 = 37$ Therefore, there are 37 students who have taken atleast one language out of Marathi, Hindi, English, and Sanskrit.

Exercises

1. How many elements are in $A_1 \cup A_2$ if there are 12 elements in A_1, 18 elements in A_2, and a) $A_1 \cap A_2 = \phi$? b) $|A_1 \cap A_2| = 1$? c) $|A_1 \cap A_2| = 6$? d) $A_1 \subseteq A_2$?

2. A survey of households in the United States reveals that 96 percent have at least one television set, 98 percent have telephone service, and 95 percent have telephone service and at least one television set. What percentage of households in the United States have neither telephone service nor a television set?

3. Find the number of elements in $A_1 \cup A_2 \cup A_3$ if there are 100 elements in each set and if

   a) the sets are pairwise disjoint.

   b) there are 50 common elements in each pair of sets and no elements in all three sets.

   c) there are 50 common elements in each pair of sets and 25 elements in all three sets.

   d) the sets are equal.

4. There are 2504 computer science students at a college. Of these, 1876 have taken a course in Java, 999 have taken a course in Linux, and 345 have taken a course in C. Further, 876 have taken courses in both Java and Linux, 231 have taken courses in both Linux and C, and 290 have taken courses in both Java and C. If 189 of these students have taken courses in Linux, Java, and C, how many of these 2504 students have not taken a course in any of these three programming languages?

5. How many students are enrolled in a course either in calculus, discrete mathematics, data structures, or programming languages at a college if there are $507, 292, 312$, and 344 students in these courses, respectively; 14 in both calculus and data structures; 213 in both calculus and programming languages; 211 in both discrete mathematics and data structures; 43 in both discrete mathematics and programming languages; and no student may take calculus and discrete mathematics, or data structures and programming languages, concurrently?

6. How many elements are in the union of four sets if the sets have $50, 60, 70$, and

80 elements, respectively, each pair of the sets has 5 elements in common, each triple of the sets has 2 common elements, and 1 element is common in all four sets?

7. In a survey of 270 college students, it is found that 64 like brussels sprouts, 94 like broccoli, 58 like cauliflower, 26 like both brussels sprouts and broccoli, 28 like both brussels sprouts and cauliflower, 22 like both broccoli and cauliflower, and 14 like all three vegetables. How many of 270 students do not like any of these vegetables?

Discrete Structures
1st Stage
By Dr. Alaa Jarah

Lec_4

Counting

Introduction

The origin of combinatorics goes far back in history. Magic squares (arrays where columns, rows and diagonals all sum to the same number) were popular subjects of mathematical study in medieval times. Jewish and Arab mathematicians in the early middle ages focused on combinatorial problems that counted the number of possibilities in a situation and evaluated their probability. This subject was studied in the seventeenth century, when combinatorial questions arose in the study of gambling games. Combinatorial approach to problem solving appears in the works of leading mathematicians such as Fibonacci, Pascal, Fermat and Euler. In modern times, the works of J. J. Sylvester (late 19th century) and Percy MacMahon (early 20th century) laid the foundation for enumerative and algebraic combinatorics. In the second half of 20th century, combinatorics enjoyed a rapid growth. The growth was spurred by new connections and applications to other fields, ranging from algebra to probability, from functional analysis to number theory, etc. These connections shed the boundaries between combinatorics and parts of mathematics and theoretical computer science, but at the same time led to a partial fragmentation of the field.

Combinatorics is that part of mathematics which deals with counting and enumeration of specified objects, patterns or designs. Counting is also required to determine whether there are enough telephone numbers or Internet protocol addresses to meet demand. Recently, it has played a key role in mathematical biology, especially in sequencing DNA.

4.1 Product and Sum Rule

Suppose that a password on a computer system consists of six, seven or eight characters. Each of these characters must be a digit or a letter of the alphabet. Each password must contain at least one digit. How many such passwords are there?

The techniques needed to answer this question and a wide variety of other counting problems will be introduced in this section. Here we study two basic counting principles, the product rule and the sum rule.

4.1.1 The Product Rule

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

Note: The way to perform the second task does not depend on the way in which the first task is performed.

Example 4.1. Chairs of an auditorium are to be labeled with an uppercase English letter followed by a positive integer not exceeding 100. What is the largest number of chairs that can be labeled differently?

Solution: The procedure of labellings a chair consists of two tasks, namely, assigning to the seat one of the 26 uppercase English letters, and then assigning to it one of the 100 possible integers. The product rule shows that there are $26\times100 = 2600$ different ways that a chair can be labeled. Therefore, the largest number of chairs that can be labeled differently is 2600.

Example 4.2. There are 32 microcomputers in a computer center. Each microcomputer has 24 ports. How many different ports to a microcomputer in the center are there?

Solution: The procedure of choosing a port consists of two tasks, first picking a microcomputer and then picking a port on this microcomputer. Because there are 32 ways to choose the microcomputer and 24 ways to choose the port no matter which microcomputer has been selected, the product rule shows that there are $32 \times 24 = 768$ ports.

Example 4.3. A new company with just two employees, Anil and Neel, rents a floor of a building with 12 offices. How many ways are there to assign different offices to these two employees?

Solution: The procedure of assigning offices to these two employees consists of assigning an office to Anil, which can be done in 12 ways, then assigning an office to Neel different from the office assigned to Anil and which can be done in 11 ways. By the product rule, there are 12 × 11 = 132 ways to assign offices to these two employees.

An extended version of the product rule is often useful.

Generalized Product Rule : Suppose that a procedure is carried out by performing the tasks $T_1, T_2, ..., T_m$ in sequence. If each task T_i, $i = 1,2,...,m$, can be done in n_i ways, regardless of how the previous tasks were done, then there are $n_1 \times n_2 \times \cdots \times n_m$ ways to carry out the procedure.

Example 4.4. A certain type of car can be purchased in any of five colors, with a manual or automatic transmission, and with any of three engine sizes. How many different car packages are available?

Solution: We can select colour in 5 ways, we can select transmission type in 2 ways, we can select engine type in 3 ways. Therefore by generalized product rule there are 5 × 2 × 3 = 30 car packages available.

Example 4.5. Let L be the set of Washington state license plates, three numbers followed by three capital letters. How many license plates are in the set?

Solution: Each letter on license plate can be selected in 26 ways, each digit on license plate can be selected in 10 ways. Therefore by multiplication principle there are 26 × 26 × 26 × 10 × 10 × 10 = 17,576,000 Washington state license plates.

Example 4.6. In the above example if letters and digits on license plate can not be repeated, then find the number of possible license plates.

Solution: First letter on license plate can be selected in 26 ways. Since there is no repetition, second letter on license plate can be selected in 25 ways, third letter on license plate can be selected in 24 ways. First digit on license plate can be selected in 10 ways, second digit on license plate can be selected in 9 ways, third digit on license plate can be selected in 8 ways. Therefore by multiplication principle there are 26 × 25 × 24 × 10 × 9 × 8 = 11,232,000 required license plates.

Example 4.7. How many different 4-letter radio station call letters (upper case) can be made

a) if the first letter must be a K or W and no letter may be repeated?

b) if repeats are allowed (but the first letter is a K or W).

c) How many of the 4-letter call letters (starting with K or W) with no repeats endin R?

Solution: a) Since first letter is K or W, there are 2 ways to select first letter. Since there is no repetition, there are 25 ways to select second letter, 24 ways to select third letter, 23 ways to select fourth letter. By multiplication principle, there are 2 × 25 × 24 × 23 = 27,600 radio station call letters.

b) Since first letter is K or W, there are 2 ways to select first letter. Since repetition is allowed, there are 26 ways to select second letter, 26 ways to select third letter, 26 ways to select fourth letter. By multiplication principle, there are 2 × 26 × 26 × 26 = 35,152 radio station call letters.

$$\underline{2} \times \underline{24} \times \underline{23} \times \underline{1}$$

c) The last place can be filled in 1 way(with R). Since the first letter is K or W, thereare 2 ways to select first letter. Since repetition is not allowed, there are 24 ways to select the second letter and 23 ways to select the third letter. By multiplication principle, there are 2 × 24 × 23 × 1 radio station call letters that can be made.

Example 4.8. How many different bit (each bit is either 0 or 1) strings of length seven are there?

Solution: Each of the seven bits can be chosen in two ways, because each bit is either 0 or 1. Therefore, the product rule shows there are a total of 2^7 = 128 different bit strings of length seven.

Theorem 4.1. (Counting Functions) *The number of functions from a set with r elements to a set with n elements is n^r.*

Proof. A function corresponds to a choice of one of the *n* elements in the codomain for each of the *r* elements in the domain. Hence, by the product rule there are $\underbrace{n \times n \times \ldots \times n}_{r\ times} = n^r$ functions from a set with *r* elements to one with *n* elements.

□

Theorem 4.2. (Counting One-to-One Functions) *The number of one-to-one functions from a set with r elements to a set with n elements is* $n \times (n-1) \times (n-2) \times \ldots \times (n-r+1)$

Proof. First note that when $r > n$ there are no one-to-one functions from a set with r elements to a set with n elements.
Now let $r \leq n$. Suppose the elements in the domain are $a_1, a_2, \ldots, a_r$. There are n ways to choose the value of the function at a_1. Because the function is one-to-one, the value of the function at a_2 can be chosen in $n-1$ ways (because the value used for a_1 cannot be used again). In general, the value of the function at a_k can be chosen in $n-(k-1)$ ways. By the product rule, there are $n\times(n-1)\times(n-2)\times\ldots\times(n-r+1)$ one-to-one functions from a set with r elements to one with n elements. □

Theorem 4.3. (Counting Subsets of a Finite Set) *The number of different subsets of a finite set X with n elements is* 2^n.

Proof. Let $X = \{a_1, a_2, \ldots, a_n\}$ be a finite set. For any subset A of X we define bit-string $S_A = b_1 b_2 \ldots b_n$, where $b_i = 0$ if $a_i \notin A$ and $b_i = 1$ if $a_i \in A$. define function φ from power set of X to set of all bit strings of length n as below.

$\varphi(A) = S_A$ for all $A \subseteq X$. Note that φ is one-one and onto function. Therefore number of subsets of X is number of bit strings of length n. By the product rule, there are 2^n bit strings of length n. Hence total number of subsets of $X = 2^n$. □

Note: The product rule is often phrased in terms of sets in the following way.
If $A_1, A_2, \ldots, A_m$ are finite sets, then the number of elements in the Cartesian product of these sets is the product of the number of elements in each set. To relate this to the product rule, note that the task of choosing an element in the Cartesian product $A_1 \times A_2 \times \ldots \times A_m$ is done by choosing an element in A_1, an element in $A_2, \ldots,$ and an element in A_m. By the product rule it follows that $|A_1 \times A_2 \times \ldots \times A_m| = |A_1| \times |A_2| \times \ldots \times |A_m|$.

4.1.2 The Sum Rule

The Sum Rule : If a task can be done either in one of n_1 ways or in one of n_2 ways, where none of the set of n_1 ways is the same as any of the set of n_2 ways, then there are $n_1 + n_2$ ways to do the task.

Example 4.9. Suppose there are 5 different types of burgers and 8 different types of pizzas. How many selections does a customer have ?
Solution: There are 5 choices for the burgers and 8 choices for the pizzas. We have to select one burger or one pizza. By addition principle there are 5+8 = 13 possible selections.

We can extend the sum rule to more than two tasks.
Generalized Sum Rule :
Suppose that a task can be done in one of n_1 ways, in one of n_2 ways,..., or in one of n_m ways, where 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 $1 \leq i < j \leq m$. Then the number of ways to do the task is $n_1 + n_2 + \ldots + n_m$.

Example 4.10. 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 choices are there for this representative if there are 37 members of the mathematics faculty and 83 mathematics majors and no one is both a faculty member and a student?
Solution: There are 37 ways to choose a member of the mathematics faculty and there are 83 ways to choose a student who is a mathematics major. Choosing a member of mathematics faculty is never same as choosing a student who is a mathematics major because no one is both a faculty member and a student. By the sum rule it follows that there are 37 + 83 = 120 possible ways to pick this representative.

Example 4.11. A student can choose a project from one of three lists. The three lists contain 23, 15, and 19 possible projects, respectively. No project is in more than one list. How many possible projects are there to choose from?
Solution: The student can choose a project by selecting a project from the first list, the second list, or the third list. Because no project is in more than one list, by the sum rule there are 23 + 15 + 19 = 57 ways to choose a project.

Example 4.12. Each user on a computer system has a password, which is six to eight characters long, where each character is an uppercase letter or a digit. Each password must contain at least one digit. How many possible passwords are there?
Solution: Let P be the total number of possible passwords, and let P_6, P_7, and P_8 denote the number of possible passwords of length 6,7, and 8, respectively. By the sum rule, $P = P_6 + P_7 + P_8$. We will now find P_6, P_7, and P_8. Finding P_6 directly is difficult. To find P_6 it is easier to find the number of strings of uppercase letters and digits that are six characters long, including those with no digits, and subtract from this the number of strings with no digits. By the product rule, the number of strings of six characters is 36^6, and the number of strings with no digits is 26^6. Hence,
$P_6 = 36^6 - 26^6 = 2{,}176{,}782{,}336 - 308{,}915{,}776 = 1{,}867{,}866{,}560$. Similarly, we have P_7

$= 36^7 - 26^7 = 78{,}364{,}164{,}096 - 8{,}031{,}810{,}176 = 70{,}332{,}353{,}920$ and

$P_8 =$ $- 26 \;=\; 2,821,109,907,456 - 208,827,064$ 36^8 ^{8} ,576 =
$P = P_6 + P_7 + P_8 = 2,684,483,063,360.$ 2,612,282,842, 880.

Consequently,

4.2 The Division Rule

We have introduced the product and sum rules for counting. You may wonder whether there is also a division rule for counting. In fact, there is such a rule, which can be useful when solving certain types of enumeration problems.

The Division Rule : There are $\frac{n}{d}$ ways to do a task if it can be done using a procedure that can be carried out in n ways, and for every way w, exactly d of the n ways correspond to way w.
We can restate the division rule in terms of sets: If the finite set A is the union of n pairwise disjoint subsets each with d elements, then $n = \frac{|A|}{d}$.

We can also formulate the division rule in terms of functions: If f is a function from A to B where A and B are finite sets, and that for every value $y \in B$ there are exactly

d values $x \in A$ such that $f(x) = y$ (in which case, we say that f is d -to-one), then

$|B| = \frac{|A|}{d}$.

Example 4.13. How many different ways are there to seat four people around a circular table, where two seatings are considered the same when each person has the same left neighbor and the same right neighbor?

Solution: We arbitrarily select a seat at the table and label it seat 1. We number the rest of the seats in numerical order, proceeding clockwise around the table. Note that there are four ways to select the person for seat 1, three ways to select the person for seat 2, two ways to select the person for seat 3, and one way to select the person for seat 4. Thus, there are 4! = 24 ways to order the given four people for these seats. However, each of the four choices for seat 1 leads to the same arrangement, as we distinguish two arrangements only when one of the people has a different immediate left or immediate right neighbor. Because there are four ways to choose the person for seat 1, by the division rule there are $\frac{24}{4}$ = 6 different seating arrangements of four people around the circular table.

Exercises

1. There are 18 mathematics majors and 325 computer science majors at a college.
   a) In how many ways can two representatives be picked so that one is a mathematics major and the other is a computer science major?
   b) In how many ways can one representative be picked who is either a mathematics major or a computer science major?

2. A multiple-choice test contains 10 questions. There are four possible answersfor each question.
   a) In how many ways can a student answer the questions on the test if thestudent answers every question?
   b) In how many ways can a student answer the questions on the test if thestudent can leave answers blank?

3. Six different airlines fly from Chennai to Mumbai and seven fly from Mumbaito Delhi. How many different pairs of airlines can you choose on which to book a trip from Chennai to Delhi via Mumbai, when you pick an airline for the flight to Mumbai and an airline for the continuation flight to Delhi?

4. How many different three capital letter initials can people have?

5. How many different three capital letter initials are there that begin with an A?

6. How many bit strings with length not exceeding n, where n is a positive integer, consist entirely of $1s$, not counting the empty string?

7. How many strings of five *ASCII* characters contain the character 'a' at least once? [Note: There are 128 different *ASCII* characters.]

8. How many 6-element RNA sequences(A,C,G,U sequences)
   a) do not contain U?
   b) end with GU?
   c) start with C?
   d) contain only A or U?

9. How many positive integers between 100 and 999 inclusive
   a) are divisible by 7?
   b) are odd?
   c) have the same three decimal digits?
   d) are not divisible by 4?

10. How many strings of three decimal digits
    a) do not contain the same digit three times?
    b) begin with an odd digit?
    c) have exactly two digits that are $4s$?

11. A committee is formed consisting of one representative from each of the 50 states in the United States, where the representative from a state is either the governor or one of the two senators from that state. How many ways are there to form this committee?

12. How many license plates can be made using either two uppercase English lettersfollowed by four digits or two digits followed by four uppercase English letters?

13. How many license plates can be made using either two or three uppercaseEnglish letters followed by either two or three digits?

14. How many strings of eight capital English letters are there

Discrete Structures

Basic Structures

Basic Structures

- Sets.
- Functions.
- Sequences.
- Summations.

Sets (1/24)

A **set** is an unordered collection of objects.

The objects in a set are called the *elements*, or *members*, of the set. A set is said to contain its elements.

Sets (2/24)

$S = \{a, b, c, d\}$

We write $a \in S$ to denote that a is an element of the set S. The notation $e \notin S$ denotes that e is not an element of the set S.

Sets (3/24)

The set O of odd positive integers less than 10 can be expressed by $O = \{1, 3, 5, 7, 9\}$.

The set of positive integers less than 100 can be denoted by $\{1, 2, 3, \ldots, 99\}$.

ellipses (…)

Sets (4/24)

Another way to describe a set is to use **set builder** notation.

The set O of odd positive integers less than 10 can be expressed by $O = \{1, 3, 5, 7, 9\}$.

$$\boxed{O = \{x \mid x \text{ is an odd positive integer less than } 10\},}$$

$$\boxed{O = \{x \in \mathbf{Z}^+ \mid x \text{ is odd and } x < 10\}.}$$

Sets (5/24)

$\mathbf{N} = \{0, 1, 2, 3, \ldots\}$, the set of all **natural numbers**

$\mathbf{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, the set of all **integers**

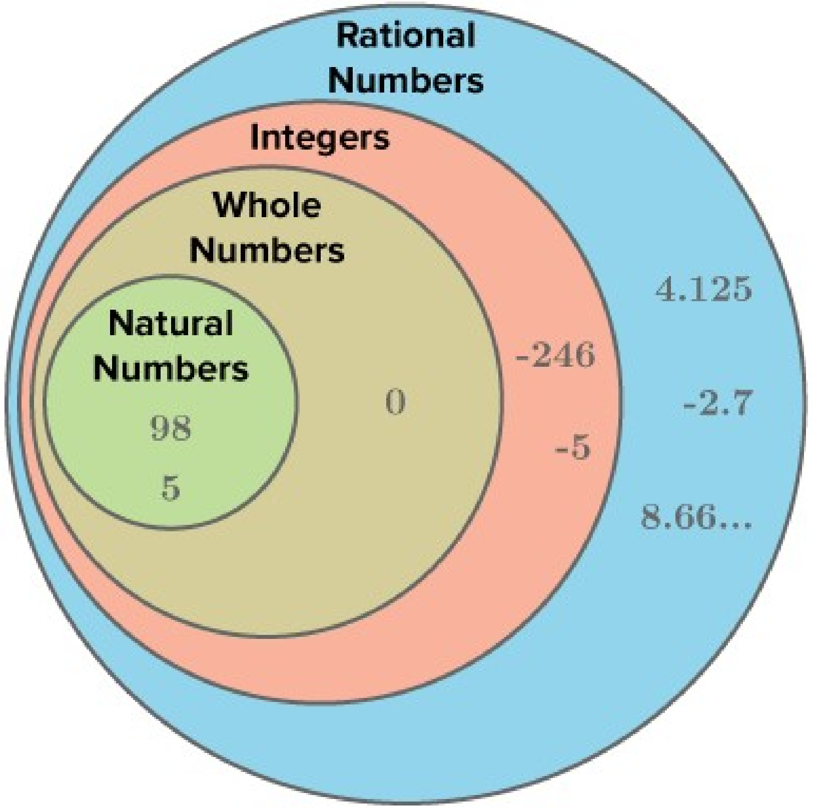
$\mathbf{Z}^+ = \{1, 2, 3, \ldots\}$, the set of all **positive integers**

$\mathbf{Q} = \{p/q \mid p \in \mathbf{Z}, q \in \mathbf{Z}, \text{and } q \neq 0\}$,
the set of all **rational numbers**

$\mathbf{R}$, the set of all **real numbers**

$\mathbf{R}^+$, the set of all **positive real numbers**

$\mathbf{C}$, the set of all **complex numbers**.



Sets (6/24)

Interval Notation

Closed interval $[a, b]$
Open interval (a, b)

$$[a, b] = \{x \mid a \le x \le b\}$$

$$[a, b) = \{x \mid a \le x < b\}$$

$$(a, b] = \{x \mid a < x \le b\}$$

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

Sets (7/24)

If A and B are sets, then A and B are equal if and only if $\forall x(x \in A \leftrightarrow x \in B)$. We write $A = B$, if A and B are equal sets.

- The sets $\{1, 3, 5\}$ and $\{3, 5, 1\}$ are equal, because they have the same elements.
- $\{1, 3, 3, 5, 5, 5\}$ is the same as the set $\{1, 3, 5\}$ because they have the same elements.

Sets (8/24)

Empty Set

There is a special set that has no elements. This set is called the empty set, or null set, and is denoted by $\emptyset$.

The empty set can also be denoted by $\{\ \}$

Sets (9/24)

Cardinality

The cardinality is the number of distinct elements in S. The cardinality of S is denoted by $|S|$.

Sets (10/24)

Example1

$S = \{a, b, c, d\}$
$|S| = 4$

$A = \{1, 2, 3, 7, 9\}$

$\emptyset = \{\ \}$

Sets (10/24)

Example1

$S = \{a, b, c, d\}$
$|S| = 4$

$A = \{1, 2, 3, 7, 9\}$
$|A| = 5$

$\emptyset = \{\ \}$
$|\emptyset| = 0$

Sets (11/24)

Example2

$S = \{a, b, c, d, \{2\}\}$

$|S| =$

$A = \{1, 2, 3, \{2,3\}, 9\}$

$|A| =$

$\{\emptyset\} = \{\{\ \}\}$

$|\{\emptyset\}| =$

Sets (11/24)

Example2

$S = \{a, b, c, d, \{2\}\}$
$|S| = 5$

$A = \{1, 2, 3, \{2,3\}, 9\}$
$|A| = 5$

$\{\emptyset\} = \{\{\ \}\}$
$|\{\emptyset\}| = 1$

Sets (12/24)

Infinite

A set is said to be **infinite** if it is not finite.
The set of positive integers is infinite.

$$Z^+ = \{1,2,3, \ldots\}$$

Sets (13/24)

Subset

The set A is said to be a subset of B if and only if every element of A is also an element of B .

We use the notation $A \subseteq B$ to indicate that A is a subset of the set B .

$$\boxed{A \subseteq B \leftrightarrow \forall x(x \in A \rightarrow x \in B)}$$

Sets (13/24)

Subset

The set A is said to be a subset of B if and only if every element of A is also an element of B .

We use the notation $A \subseteq B$ to indicate that A is a subset of the set B .

$$\boxed{(\boldsymbol{A} \subseteq \boldsymbol{B}) \equiv (\boldsymbol{B} \supseteq \boldsymbol{A})}$$

$$\boxed{A \subseteq B \leftrightarrow \forall x(x \in A \rightarrow x \in B)}$$

Sets (13/24)

Subset

For every set S,

$(i)\ \emptyset \subseteq S$ and $(ii)\ S \subseteq S$.

To show that two sets A and B are equal, show that $A \subseteq B$ and $B \subseteq A$.

Sets (14/24)

Proper Subset

The set A is a subset of the set B but that $A \neq B$,
we write $A \subset B$
and say that A is a **proper subset** of B.

$$\boxed{A \subset B \leftrightarrow \big(\forall x(x \in A \rightarrow x \in B) \wedge \exists x(x \in B \wedge x \notin A)\big)}$$

exist x

Sets (15/24)

Example

For each of the following sets,
determine whether 3 is an element of that set.

$\{1,2,3,4\}$

$\{\{1\},\{2\},\{3\},\{4\}\}$

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

Sets (16/24)

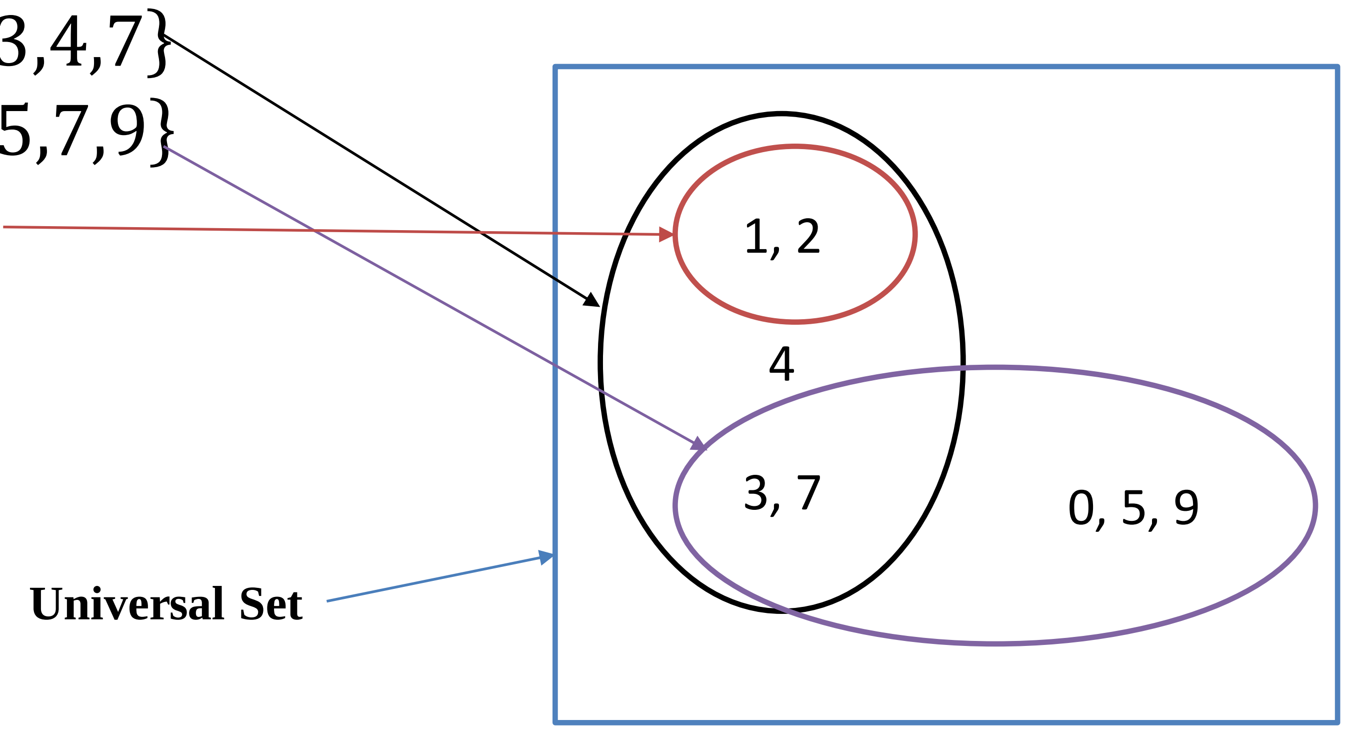
Venn Diagram

$A = \{1,2,3,4,7\}$
$B = \{0,3,5,7,9\}$
$C = \{1,2\}$

Sets (17/24)

Venn Diagram

$A = \{1,2,3,4,7\}$
$B = \{0,3,5,7,9\}$
$C = \{1,2\}$



Power Set

The set of all subsets.

If the set is S. The power set of S is denoted by $P(S)$.

The number of elements in the power set is $2^{|S|}$

Sets (18/24)

Power Set

The set of all subsets.

If the set is $S.$ The power set of S is denoted by $P(S)$.

The number of elements in the power set is $2^{|S|}$

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

$\boldsymbol{|P(S)| = 2^3 =}$ **8 elements**

$P(S) = 2^S$

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

Sets (19/24)

Example1

What is the power set of the empty set?

Sets (19/24)

Example1

What is the power set of the empty set?

$$\mathcal{P}(\emptyset) = \{\emptyset\}.$$

Sets (20/24)

Example2

What is the power set of the set {Ø}?

Sets (20/24)

Example2

What is the power set of the set $\{\emptyset\}$?

$$\mathcal{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}.$$

Sets (21/24)

The ordered n-tuple

The ordered n-tuple $(a_1, a_2, \dots, a_n)$ is the ordered collection that has a_1 as its first element, a_2 as its second element, … , and a_n as its nth element.

In particular, ordered 2-tuples are called ordered pairs (e.g., the ordered pairs (a, b))

Sets (22/24)

Cartesian Products

Let A and B be sets.

The **Cartesian product** of A and B, denoted by $A \times B$, is the set of all ordered pairs (a, b), where $a \in A$ and $b \in B$. Hence, $A \times B = \{(a, b) \mid a \in A \land b \in B\}$.

Cartesian Products - Example

Let $A = \{1,2\}$, and $B = \{a, b, c\}$

$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$.

$|A \times B| = |A| * |B| = 2 * 3 = 6$

Cartesian Products - Example

Let $A = \{1,2\}$, and $B = \{a, b, c\}$

$A \times B = \{(1, a), (1, b), (1, c), (2, a), (2, b), (2, c)\}$.

$|A \times B| = |A| * |B| = 2 * 3 = 6$

$\boxed{\text{Find } B \times A\,?}$

Sets (23/24)

The Cartesian product of more than two sets.

The Cartesian product of the sets $A_1, A_2, \ldots, A_n$, denoted by $A_1 \times A_2 \times \cdots \times A_n$, is the set of ordered n-tuples $(a_1, a_2, \ldots, a_n)$, where a_i belongs to A_i for $i = 1, 2, \ldots, n$. In other words,

$$\boxed{\begin{gathered} A_1 \times A_2 \times \cdots \times A_n = \\ \{(a_1, a_2, \ldots, a_n) \mid a_i \in A_i \text{ for } i = 1, 2, \ldots, n\}. \end{gathered}}$$

Sets (24/24)

Example:

$A \times B \times C$, where $A = \{0, 1\}$, $B = \{1, 2\}$, and $C = \{0, 1, 2\}$

$$\boxed{\begin{aligned} A \times B \times C = \{&(0, 1, 0), (0, 1, 1), (0, 1, 2), (0, 2, 0), (0, 2, 1), (0, 2, 2), \\ &(1, 1, 0), (1, 1, 1), (1, 1, 2), (1, 2, 0), (1, 2, 1), (1, 2, 2)\}.\end{aligned}}$$

Set Operations (1/7)

Union

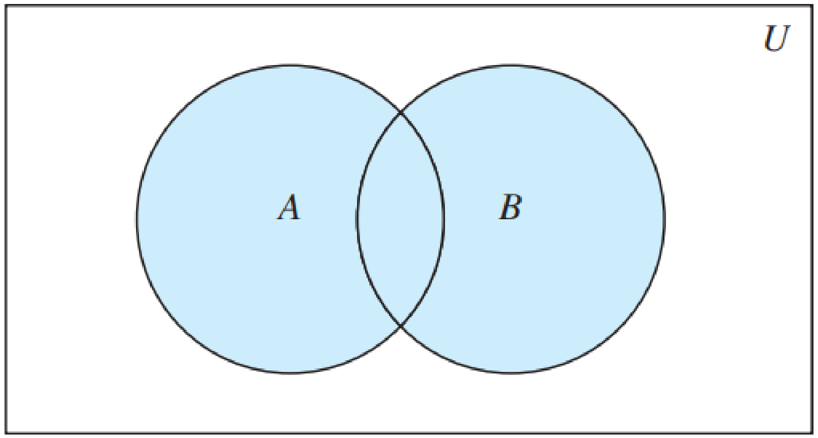
Let A and B be sets. The **union** of the sets A and B , denoted by $A \cup B$, is the set that contains those elements that are either in A or in B , or in both.

$$\boxed{A \cup B = \{x \mid x \in A \vee x \in B\}}$$

Set Operations (1/7)

Union

Let A and B be sets. The **union** of the sets A and B , denoted by $A \cup B$, is the set that contains those elements that are either in A or in B , or in both.



$A \cup B$ is shaded.

Set Operations (1/7)

Union

Let A and B be sets. The **union** of the sets A and B , denoted by $A \cup B$, is the set that contains those elements that are either in A or in B , or in both.

The union of the sets $\{1, 3, 5\}$ and $\{1, 2, 3\}$ is the set $\{1, 2, 3, 5\}$

Set Operations (2/7)

Intersection

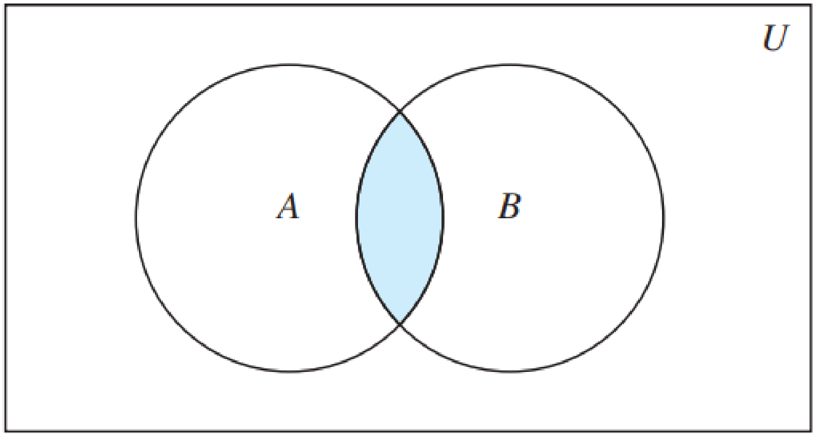
Let A and B be sets. The **intersection** of the sets A and B , denoted by $A \cap B$, is the set that contains those elements that are in both A and B.

$$\boxed{A \cap B = \{x \mid x \in A \land x \in B\}}$$

Set Operations (2/7)

Intersection

Let A and B be sets. The **intersection** of the sets A and B , denoted by $A \cap B$, is the set that contains those elements that are in both A and B.



$A \cap B$ is shaded.

Set Operations (2/7)

Intersection

Let A and B be sets. The **intersection** of the sets A and B , denoted by $A \cap B$, is the set that contains those elements that are in both A and B.

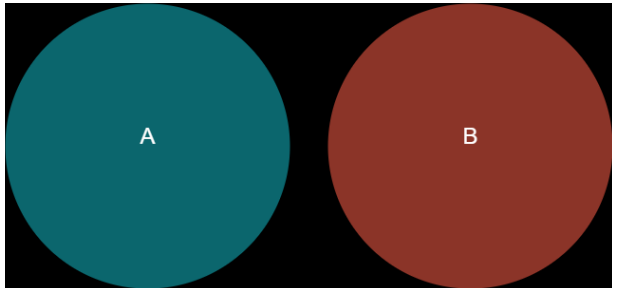
The intersection of the sets $\{1, 3, 5\}$ and $\{1, 2, 3\}$ is the set $\{1, 3\}$

Set Operations (3/7)

Disjoint

Two sets are called **disjoint** if their intersection is the empty set.

$$A \cap B = \emptyset$$



Set Operations (4/7)

Difference

Let A and B be sets. The difference of A and B , denoted by $A - B$, is the set containing those elements that are in A but not in B.

$$\boxed{A - B = \{x \mid x \in A \wedge x \notin B\}}$$

Set Operations (4/7)

Difference

Let A and B be sets. The difference of A and B , denoted by $A - B$, is the set containing those elements that are in A but not in B.

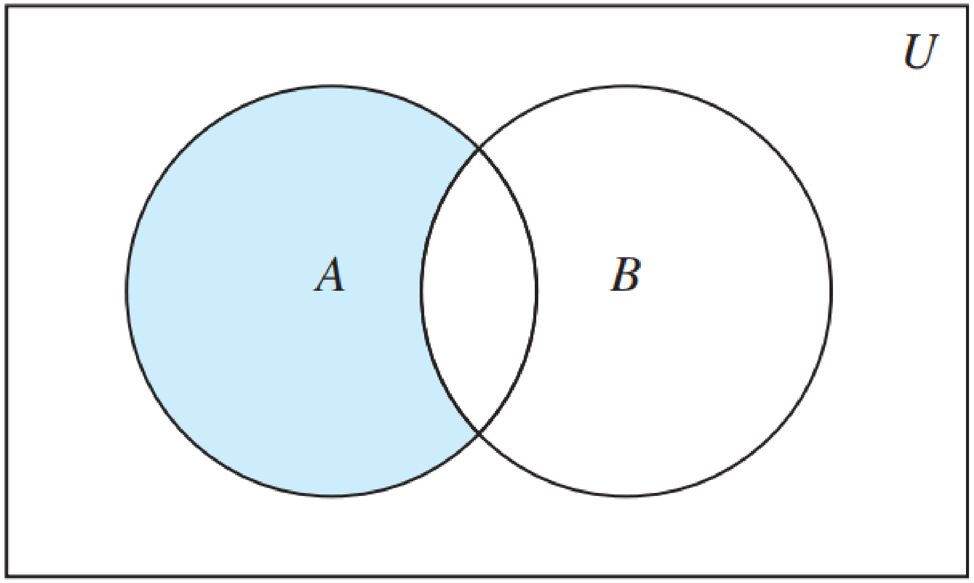
$$A = \{1,3,5\}, \qquad B = \{1,2,3\}$$

$$A - B = \{5\}$$

B-A ?

Set Operations (4/7)

Difference



$A - B$ is shaded.

Set Operations (5/7)

Complement

Let U be the universal set.

The complement of the set A , denoted by $\bar{A}$

An element x belongs to U if and only if $x \notin A$.

$$\boxed{\bar{A} = \{x \in U \mid x \notin A\}}$$

Set Operations (5/7)

Complement

Let U be the universal set.

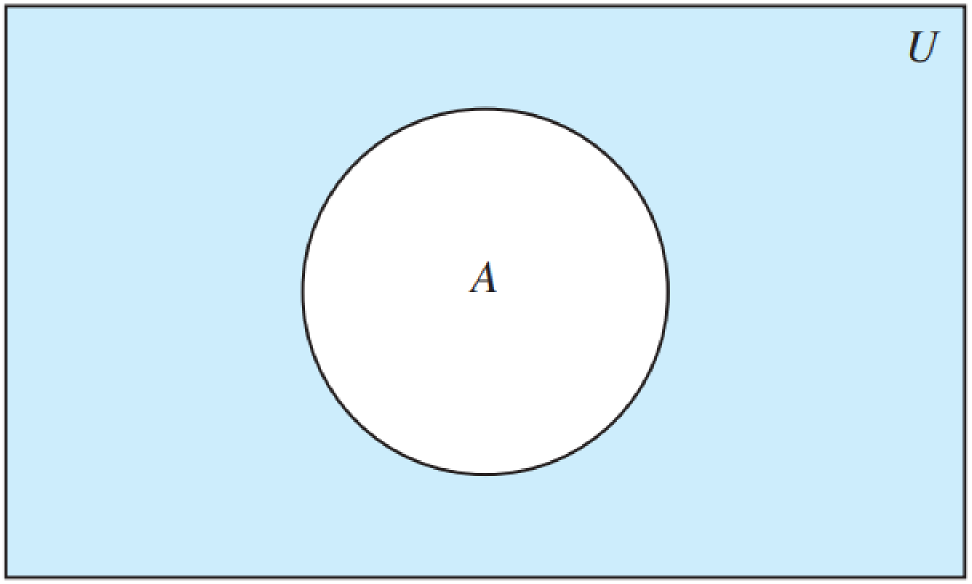
The complement of the set A , denoted by $\bar{A}$

An element x belongs to U if and only if $x \notin A$.

$$\boxed{\begin{array}{c} U = \{1,2,3,4,5\}, \qquad A = \{1,3\} \\ \bar{A} = \{2,4,5\} \end{array}}$$

Set Operations (5/7)

Complement



$\overline{A}$ is shaded.

Set Operations (6/7)

Generalized Unions

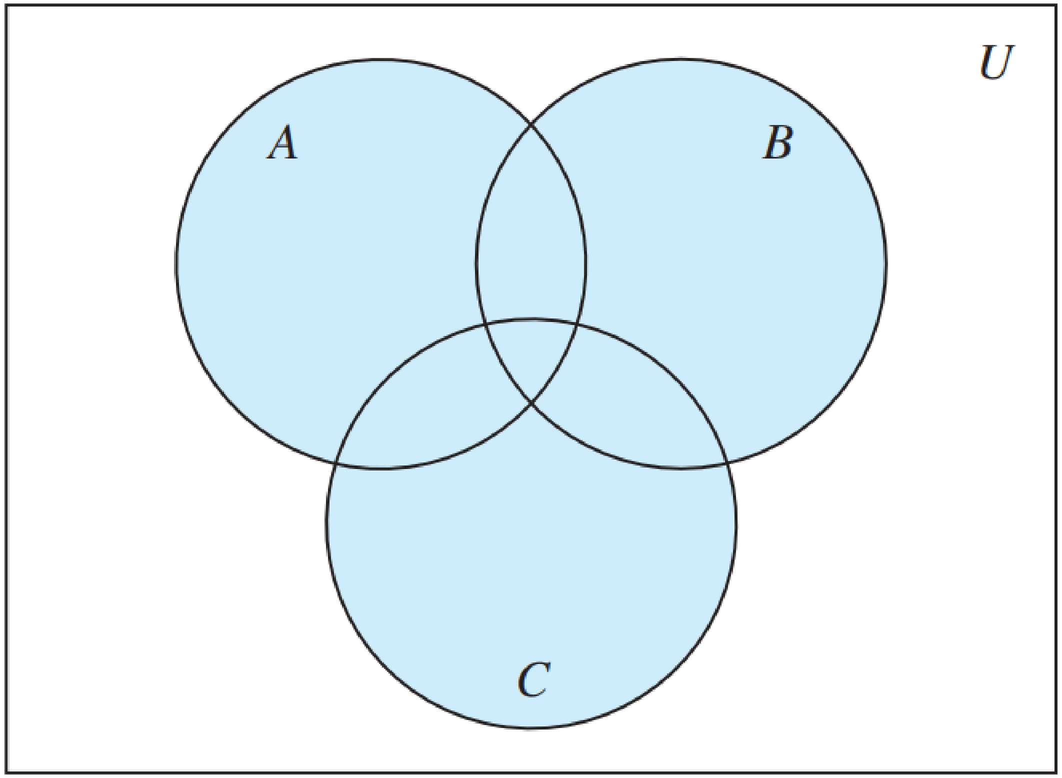
We use the notation

$$A_1 \cup A_2 \cup \cdots \cup A_n = \bigcup_{i=1}^{n} A_i$$

to denote the union of the sets $A_1, A_2, \ldots, A_n$.

Set Operations (6/7)

Generalized Unions



$A \cup B \cup C$ is shaded.

Set Operations (7/7)

Generalized Intersections

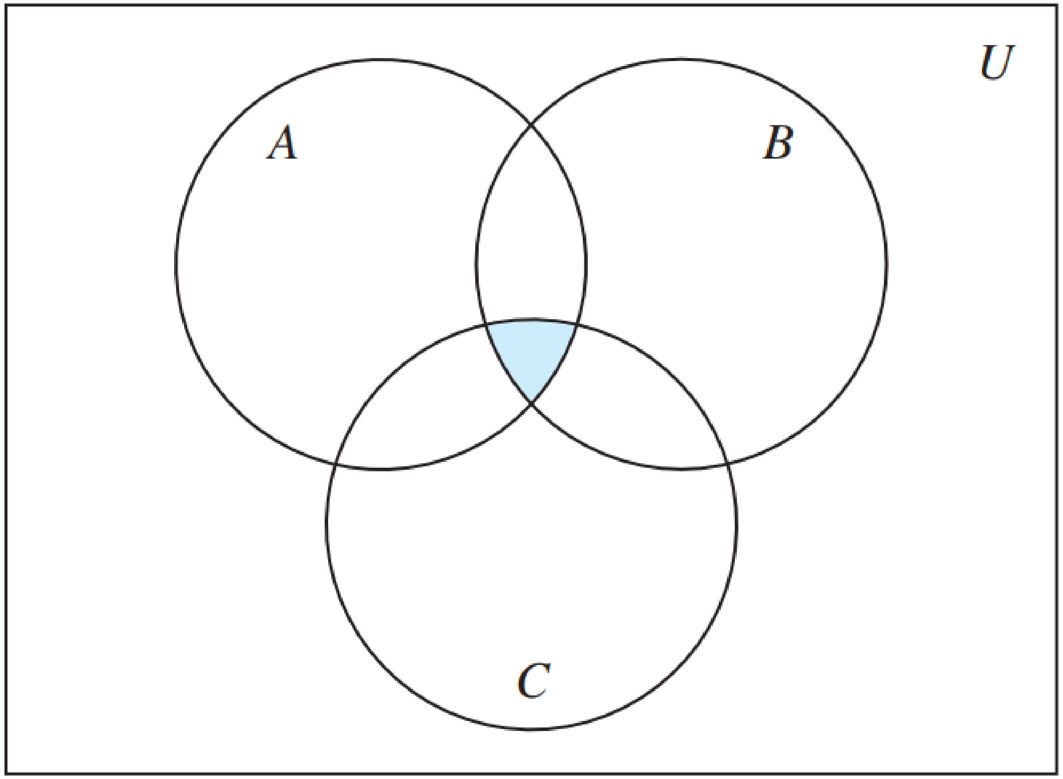
We use the notation

$$A_1 \cap A_2 \cap \cdots \cap A_n = \bigcap_{i=1}^{n} A_i$$

to denote the intersection of the sets $A_1, A_2, \ldots, A_n$.

Set Operations (7/7)

Generalized Intersections



$A \cap B \cap C$ is shaded.

Set Identities (1/8)

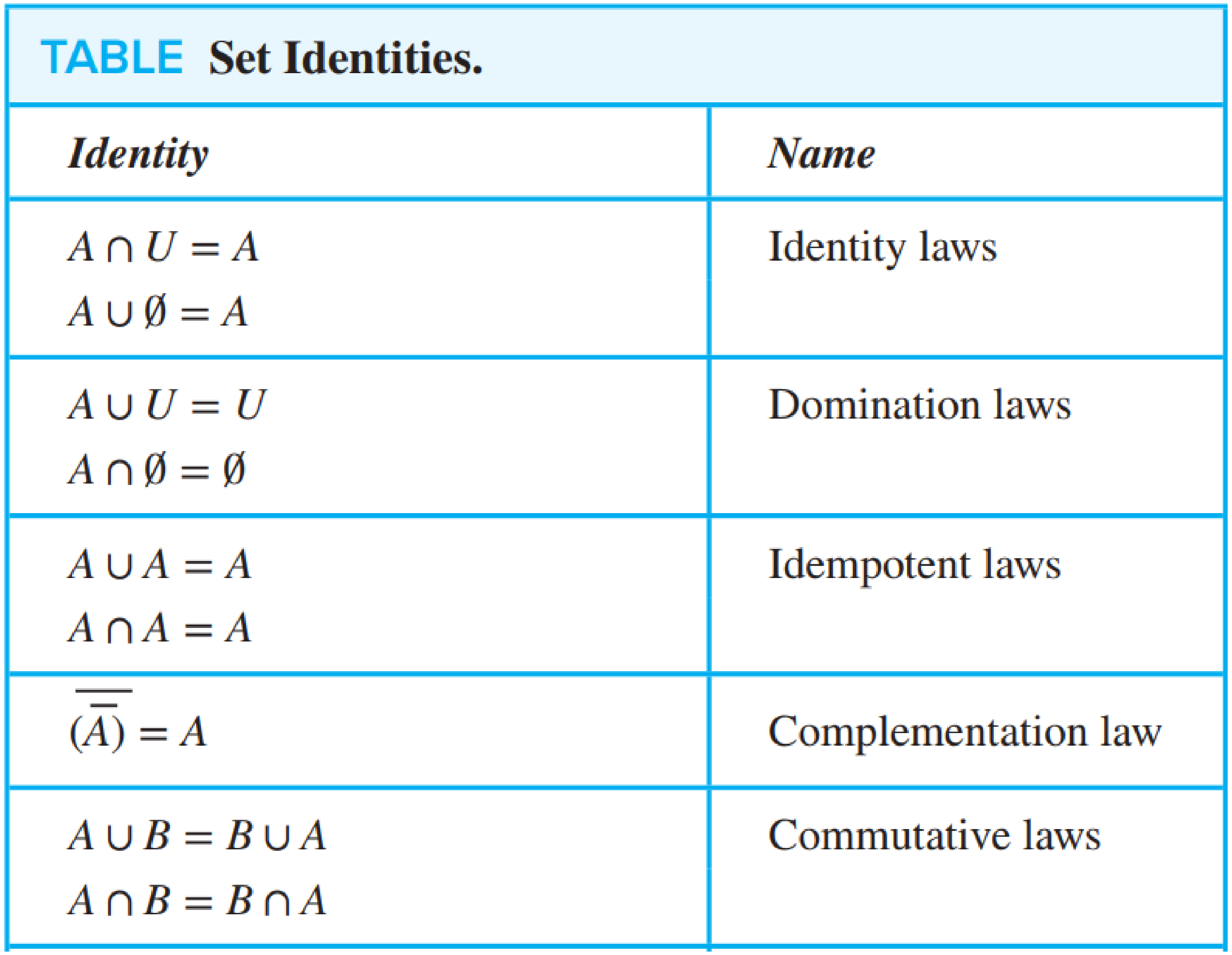


TABLE Set Identities.

Identity	*Name*
$A \cap U = A$ $A \cup \emptyset = A$	Identity laws
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Domination laws
$A \cup A = A$ $A \cap A = A$	Idempotent laws
$\overline{(\overline{A})} = A$	Complementation law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative laws

Set Identities (2/8)

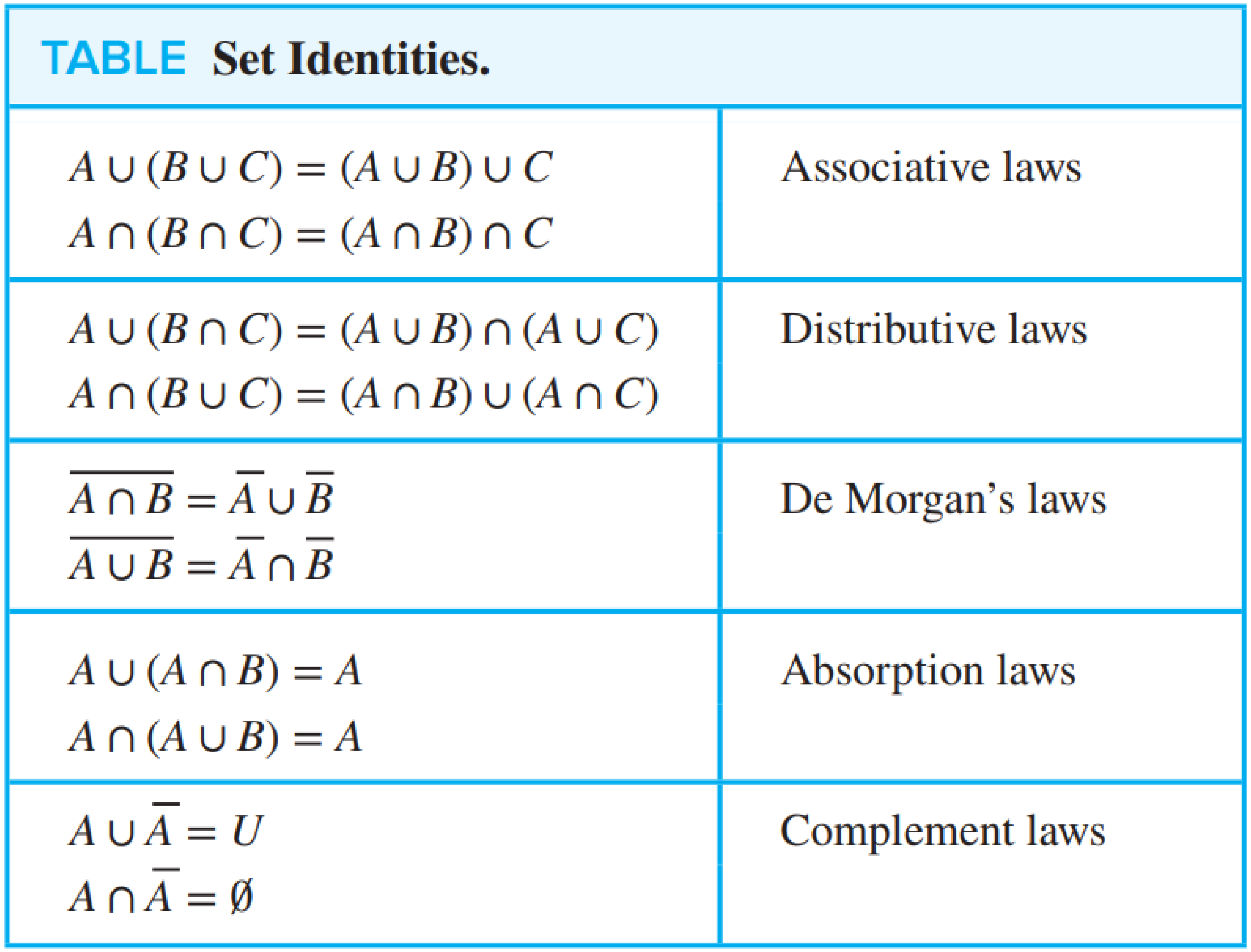


TABLE Set Identities.

$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative laws
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributive laws
$\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	De Morgan's laws
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Absorption laws
$A \cup \overline{A} = U$ $A \cap \overline{A} = \emptyset$	Complement laws

Set Identities (3/8)

Example1

Prove that $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

Set Identities (4/8)

Example1 – Answer

Prove that $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

First, we will show that $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$.

Next, we will show that $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$.

Set Identities (5/8)

First, we will show that $\overline{A \cap B} \subseteq \overline{A} \cup \overline{B}$.

$x \in \overline{A \cap B}$	by assumption
$x \notin A \cap B$	defn. of complement
$\neg((x \in A) \wedge (x \in B))$	defn. of intersection
$\neg(x \in A) \vee \neg(x \in B)$	1st De Morgan Law for Prop Logic
$x \notin A \vee x \notin B$	defn. of negation
$x \in \overline{A} \vee x \in \overline{B}$	defn. of complement
$x \in \overline{A} \cup \overline{B}$	defn. of union

Set Identities (6/8)

Next, we will show that $\overline{A} \cup \overline{B} \subseteq \overline{A \cap B}$.

$x \in \overline{A} \cup \overline{B}$	by assumption
$(x \in \overline{A}) \vee (x \in \overline{B})$	defn. of union
$(x \notin A) \vee (x \notin B)$	defn. of complement
$\neg(x \in A) \vee \neg(x \in B)$	defn. of negation
$\neg((x \in A) \wedge (x \in B))$	by 1st De Morgan Law for Prop Logic
$\neg(x \in A \cap B)$	defn. of intersection
$x \in \overline{A \cap B}$	defn. of complement

Set Identities (7/8)

Example2

Use set builder notation and logical equivalences to establish the first De Morgan law $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

Set Identities (8/8)

Example2 – Answer

Use set builder notation and logical equivalences to establish the first De Morgan law $\overline{A \cap B} = \overline{A} \cup \overline{B}$.

$\overline{A \cap B} = \{x \mid x \notin A \cap B\}$	by definition of complement
$= \{x \mid \neg(x \in (A \cap B))\}$	by definition of does not belong symbol
$= \{x \mid \neg(x \in A \wedge x \in B)\}$	by definition of intersection
$= \{x \mid \neg(x \in A) \vee \neg(x \in B)\}$	by the first De Morgan law for logical equivalences
$= \{x \mid x \notin A \vee x \notin B\}$	by definition of does not belong symbol
$= \{x \mid x \in \overline{A} \vee x \in \overline{B}\}$	by definition of complement
$= \{x \mid x \in \overline{A} \cup \overline{B}\}$	by definition of union
$= \overline{A} \cup \overline{B}$	by meaning of set builder notation

Functions (1/21)

Function

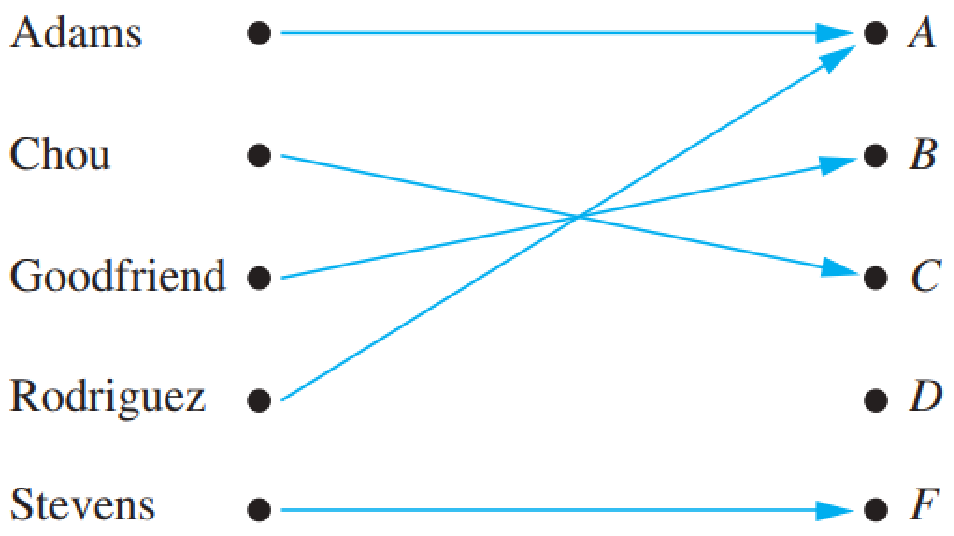
Let A and B be nonempty sets. A function f from A to B is an assignment of exactly one element of B to each element of A.

We write $f(a) = b$ if b is the unique element of B assigned by the function f to the element a of A.

If f is a function from A to B, we write $f: A \to B$.

Functions (2/21)

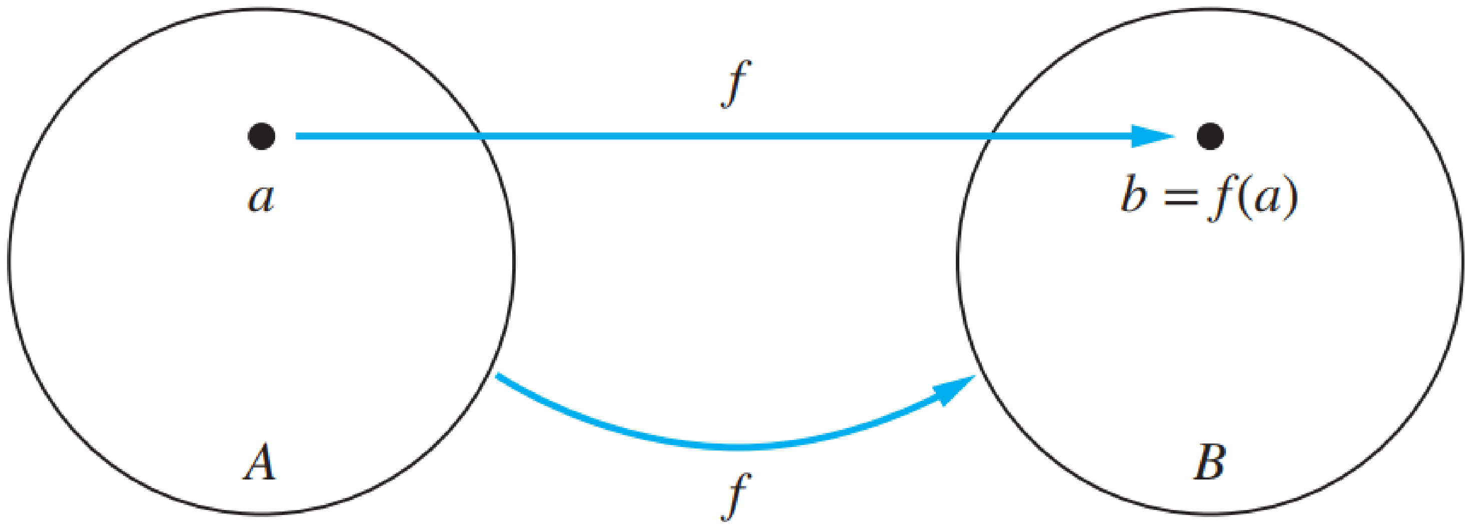
Function



Assignment of grades in a discrete mathematics class.

Functions (3/21)

The Function $f: A \to B$



The function f maps A to B.

Functions (3/21)

The Function $f: A \to B$

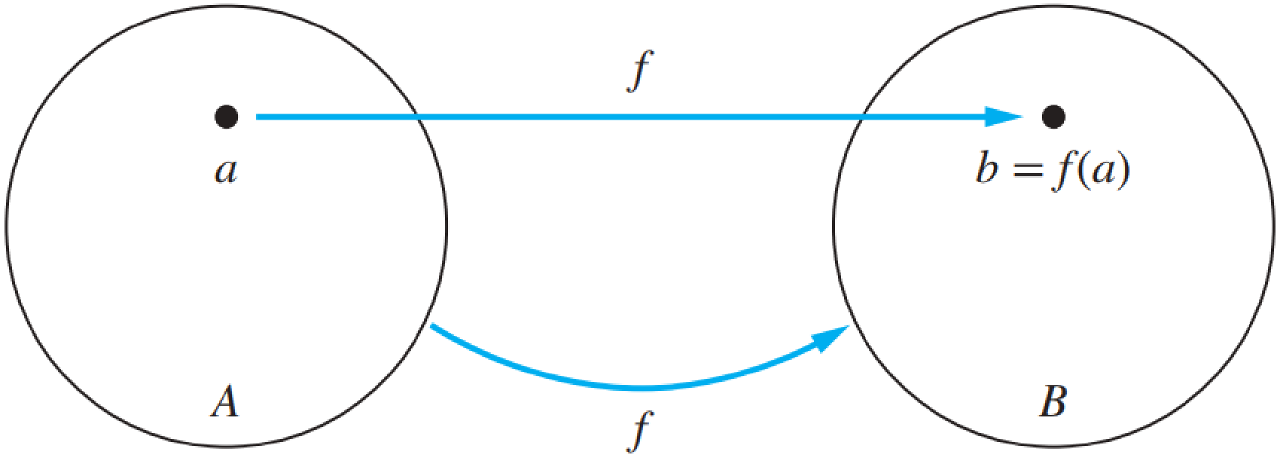
Domain: A

Co-Domain: B

$f(a) = b$

b is the ***image*** of a
a is a ***preimage*** of b

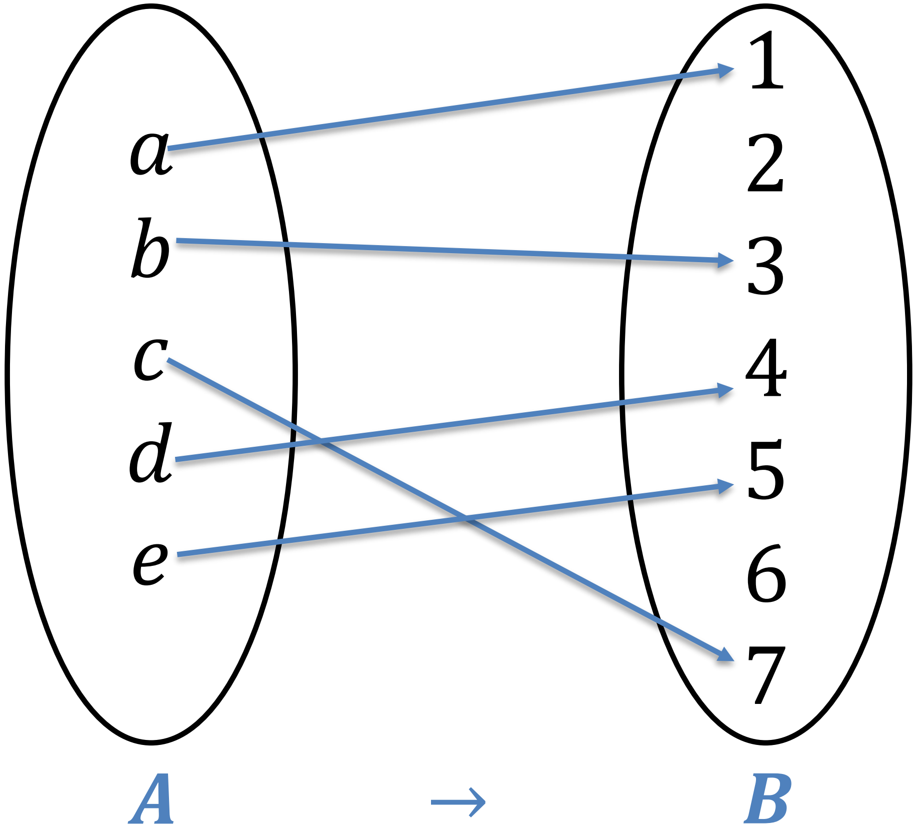
The **range**, or image, of f is the ***set of all images*** of elements of A.



The function f maps A to B.

Functions (4/21)

The Function $f: A \to B$



$\text{Domain} = \{a, b, c, d, e\}$

$\text{Co-Domain} = \{1,2,3,4,5,6,7\}$

$\text{Range} = \{1,3,4,5,7\}$

Functions (5/21)

Definition

Let f_1 and f_2 be functions from A to **R**. Then $f_1 + f_2$ and $f_1\ f_2$ are also functions from A to **R** defined for all $x \in A$ by

$$\boxed{\begin{aligned}&(f_1 + f_2)(x) = f_1(x) + f_2(x),\\ &(f_1 f_2)(x) = f_1(x) f_2(x).\end{aligned}}$$

Functions (6/21)

Example

Let f_1 and f_2 be functions from **R** to **R** such that $f_1(x) = x^2$ and $f_2(x) = x - x^2$. What are the functions $f_1 + f_2$ and $f_1 f_2$?

$$(f_1 + f_2)(x) = f_1(x) + f_2(x) = x^2 + (x - x^2) = x,$$

$$(f_1 f_2)(x) = f_1(x) f_2(x) = x^2(x - x^2) = x^3 - x^4.$$

Functions (7/21)

Definition

Let f be a function from A to B and let S be a subset of A.

The image of S under the function f is the subset of B that consists of the images of the elements of S.

We denote the image of S by $f(S)$, so

$$f(S) = \left\{ t \mid \exists s \in S \left(t = f(s) \right) \right\}.$$

or shortly $\{f(s) \mid s \in S\}$.

Functions (8/21)

Example

Let $A = \{a, b, c, d, e\}$ and $B = \{1, 2, 3, 4\}$ with $f(a) = 2$, $f(b) = 1$, $f(c) = 4$, $f(d) = 1$, and $f(e) = 1$.

$S = \{b, c, d\} \subseteq A$

The image of the subset $S = \{b, c, d\}$ is the set $f(S) = \{1, 4\}$

Functions (9/21)

One-to-One function (injective)

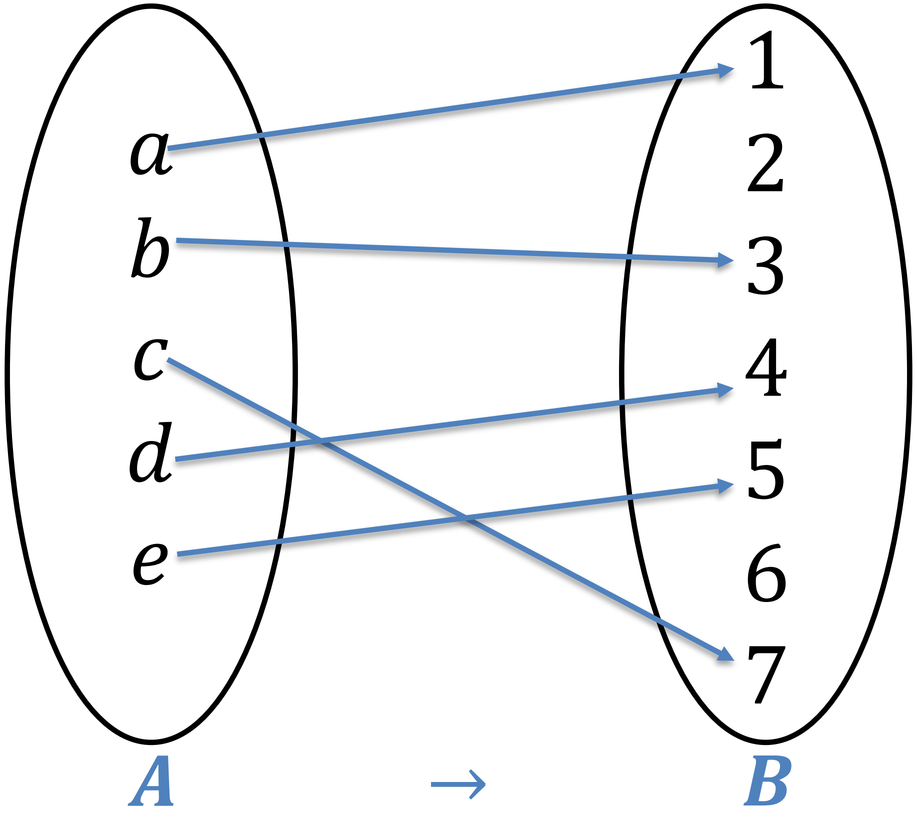
A function f is said to be **one-to-one**, or **injective**,

if and only if $f(a) = f(b)$ implies that $a = b$ for all a and b in the domain

of f.

Functions (9/21)

One-to-One function (injective)



$f(a) = 1$

$f(b) = 3$

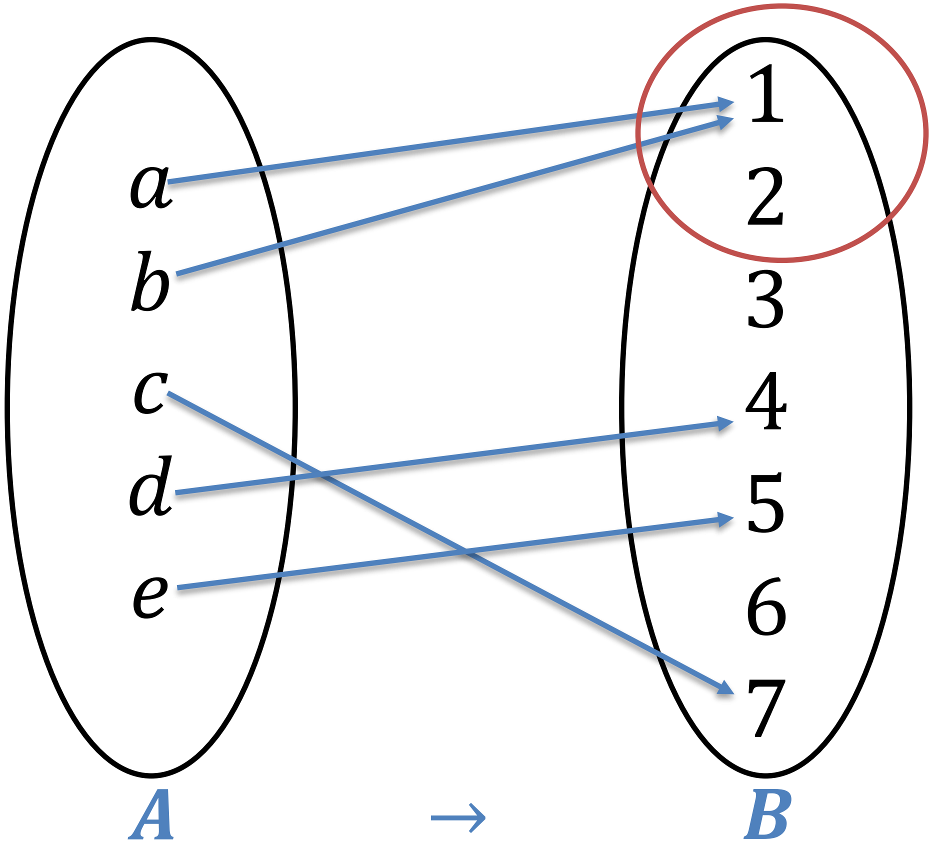
$f(c) = 7$

$f(d) = 4$

$f(e) = 5$

Functions (9/21)

NOT *One-to-One* function (Not injective)



$f(a) = 1$

$f(b) = 1$

$f(c) = 7$

$f(d) = 4$

$f(e) = 5$

Functions (10/21)

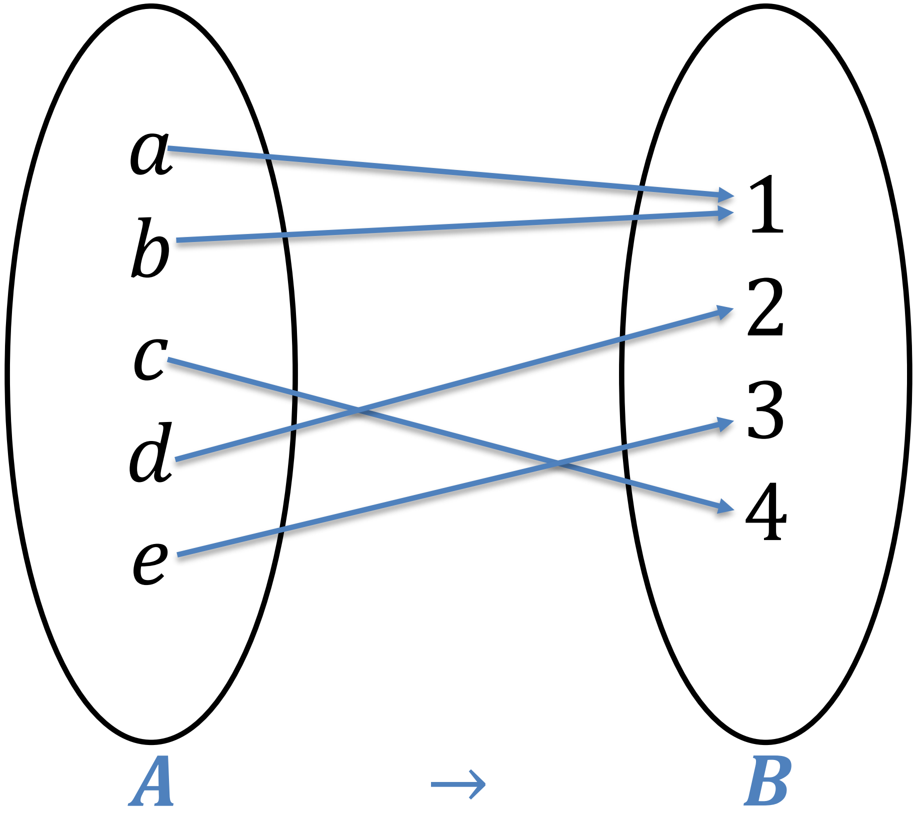
onto function (surjective)

A function *f* from *A* to *B* is called **onto**, or **surjective**, if and only if for every element $b \in B$ there is an element $a \in A$ with $f(a) = b$.

$$\boxed{\text{Co-Domain} = \text{Range}}$$

Functions (10/21)

onto function (surjective)



$f(a) = 1$

$f(b) = 1$

$f(c) = 4$

$f(d) = 2$

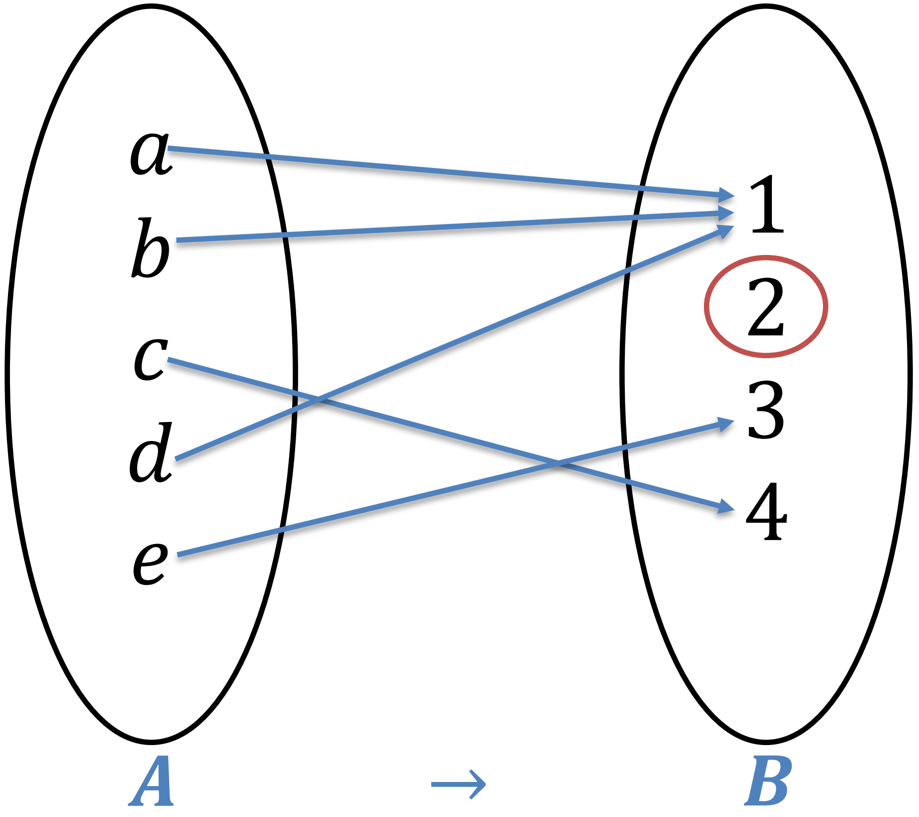
$f(e) = 3$

Co-Domain = $\{1,2,3,4\}$

Range = $\{1,2,3,4\}$

Functions (10/21)

NOT *onto* function (Not surjective)



$f(a) = 1$

$f(b) = 1$

$f(c) = 4$

$f(d) = 1$

$f(e) = 3$

Co-Domain = {1,2,3,4}

Range = {1,3,4}

Functions (11/21)

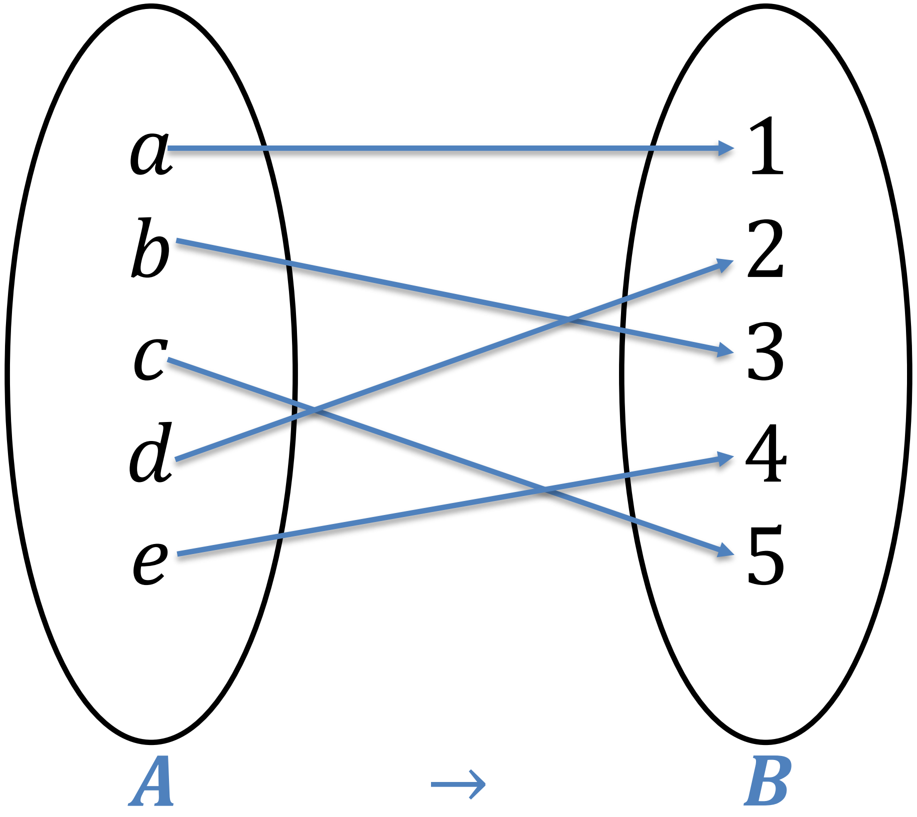
One-to-one correspondence (bijection)

The function *f* is a **one-to-one correspondence**, or a **bijection**, if it is $\boxed{\text{both one-to-one and onto.}}$

Functions (11/21)

***One-to-one correspondence* (bijection)**

$|A| = |B|$



$f(a) = 1$

$f(b) = 3$

$f(c) = 5$

$f(d) = 2$

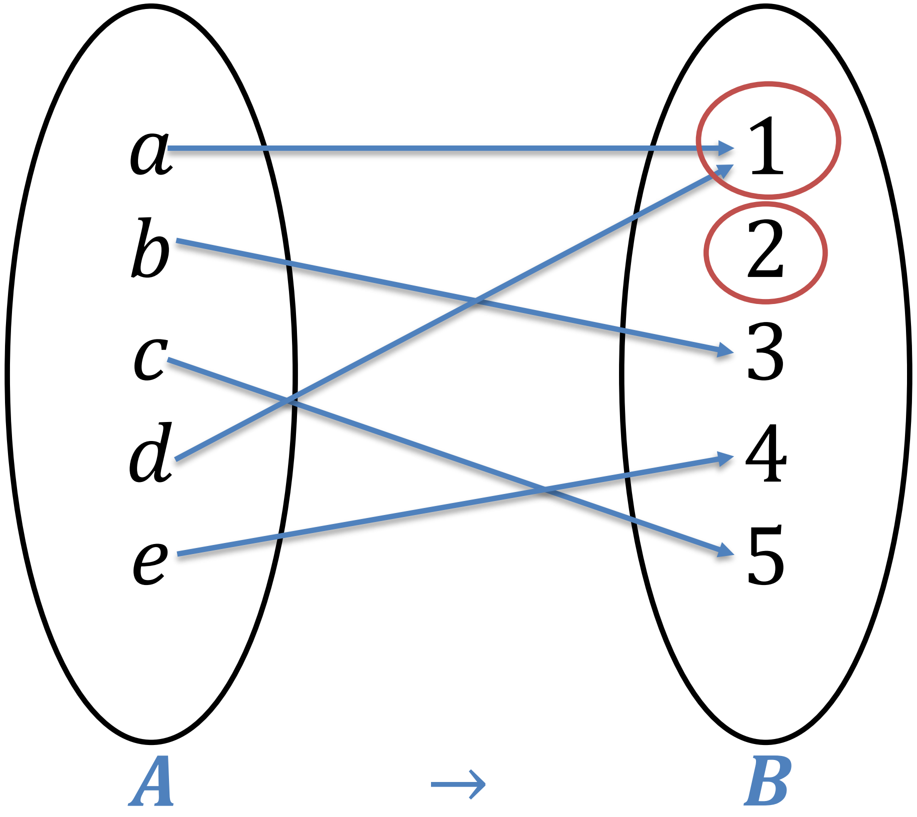
$f(e) = 4$

Co-Domain = {1,2,3,4,5}

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

Functions (11/21)

NOT *One-to-one correspondence* (Not bijection)



$f(a) = 1$

$f(b) = 3$ **NOT one-to-one**

$f(c) = 5$ **NOT onto**

$f(d) = 1$

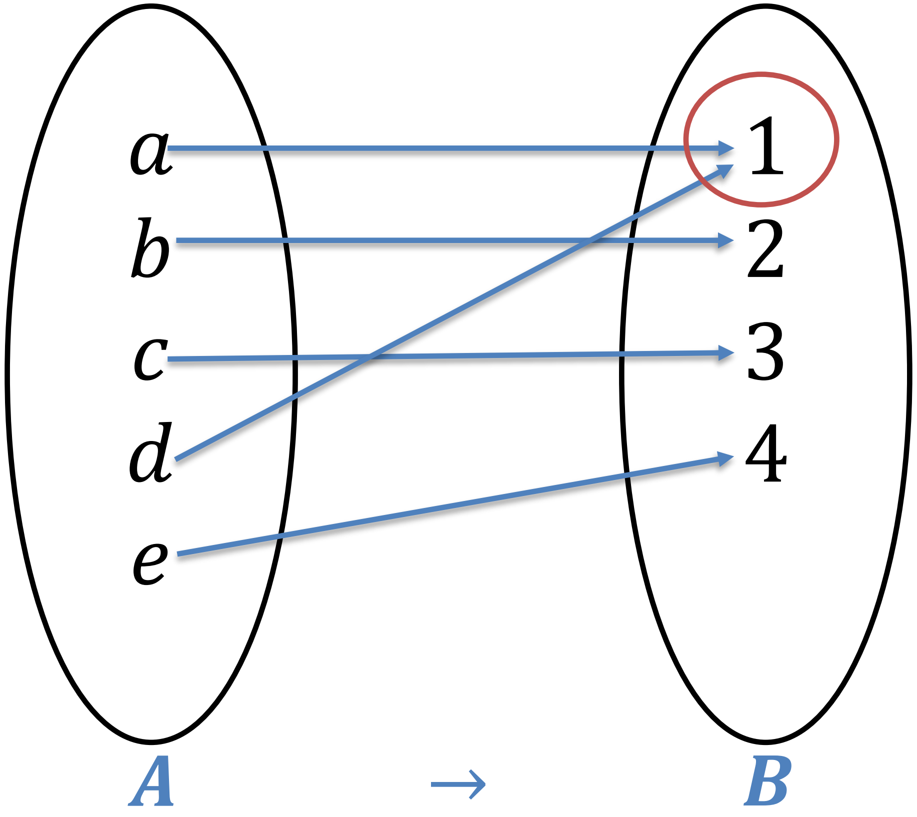
$f(e) = 4$

Co-Domain = {1,2,3,4,5}

Range = {1,3,4,5}

Functions (11/21)

NOT *One-to-one correspondence* (Not bijection)



$f(a) = 1$

$f(b) = 2$

$f(c) = 3$

$f(d) = 1$

$f(e) = 4$

Onto

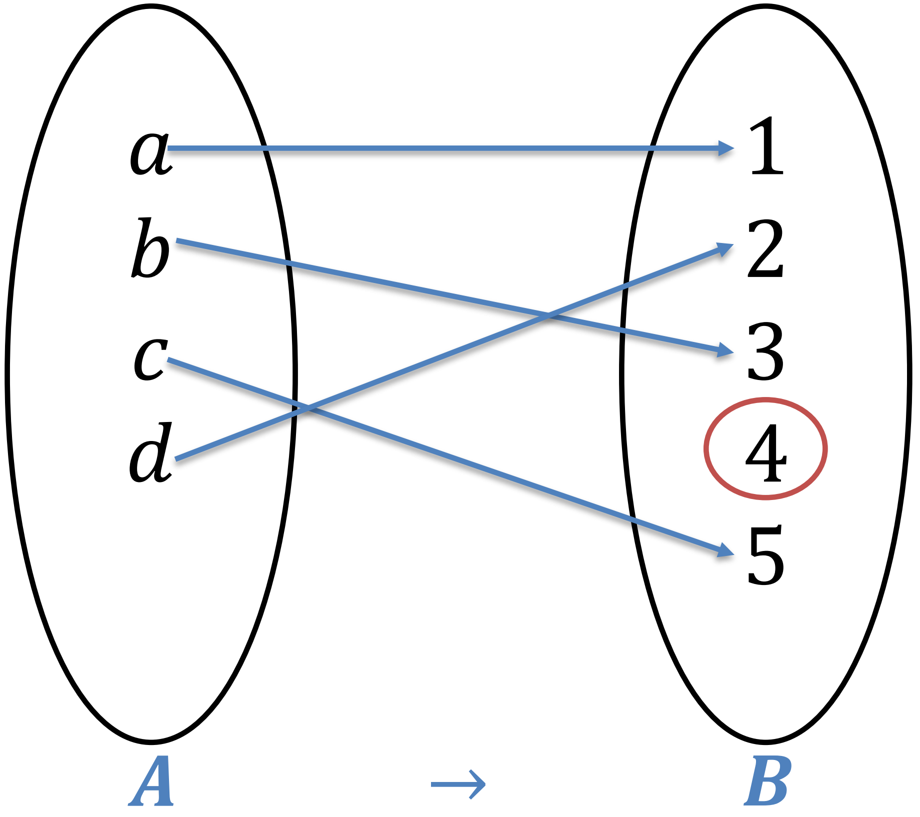
NOT one-to-one

Co-Domain = {1,2,3,4}

Range = {1,2,3,4}

Functions (11/21)

NOT *One-to-one correspondence* (Not bijection)



$f(a) = 1$

$f(b) = 3$

$f(c) = 5$

$f(d) = 2$

One-to-one

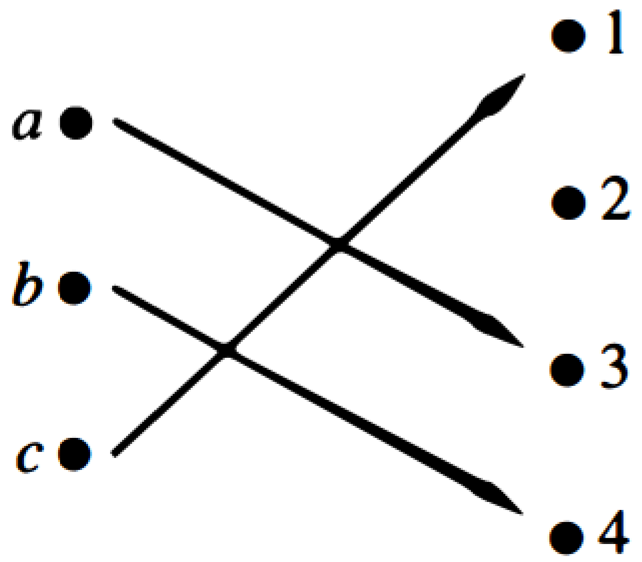
NOT onto

Co-Domain = {1,2,3,4,5}

Range = {1,2,3,5}

Functions (12/21)

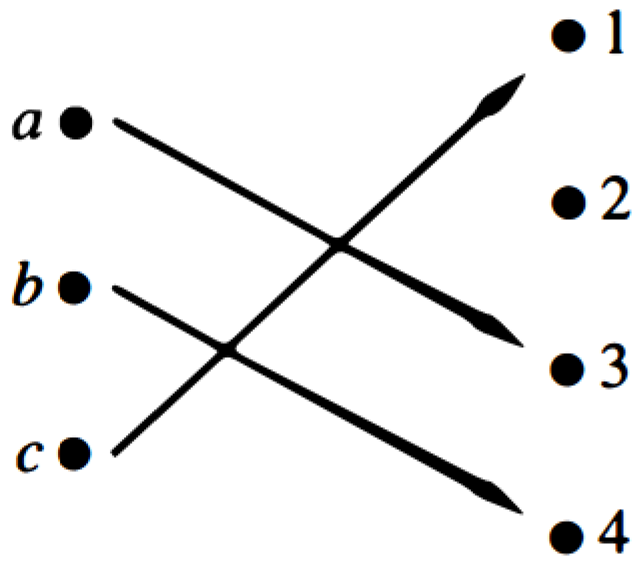
Examples



$A \rightarrow B$

Functions (12/21)

Examples



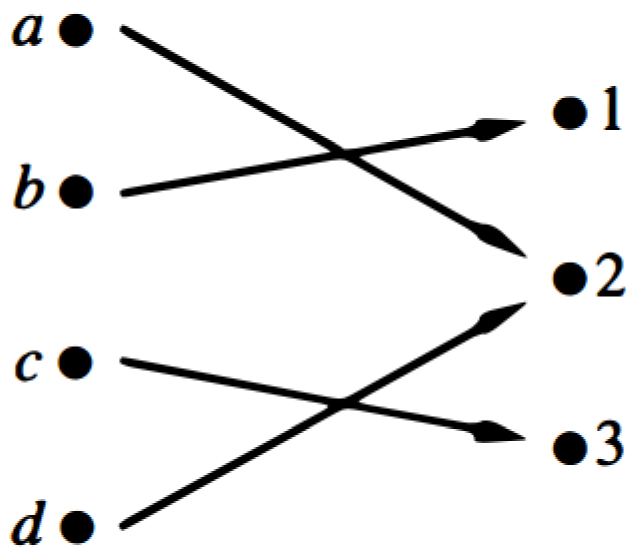
$A \rightarrow B$

One-to-one

NOT onto

Functions (12/21)

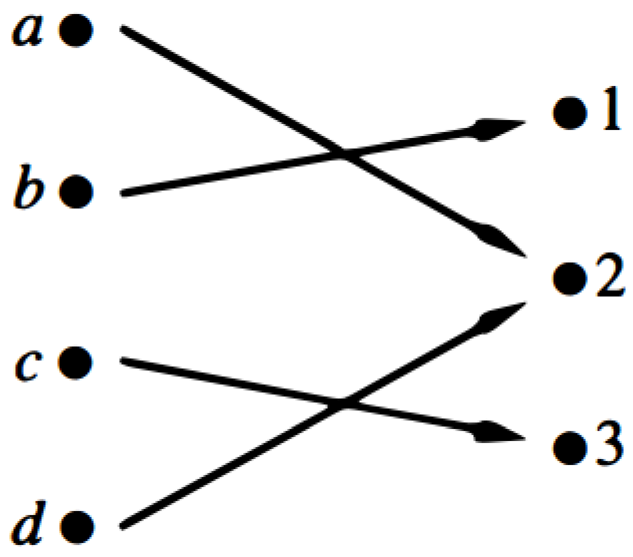
Examples



$A \quad \rightarrow \quad B$

Functions (12/21)

Examples



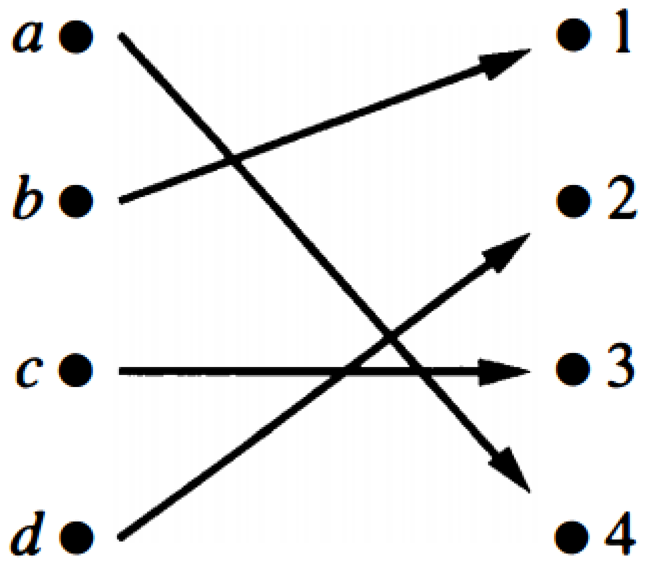
NOT One-to-one

Onto

$A \rightarrow B$

Functions (12/21)

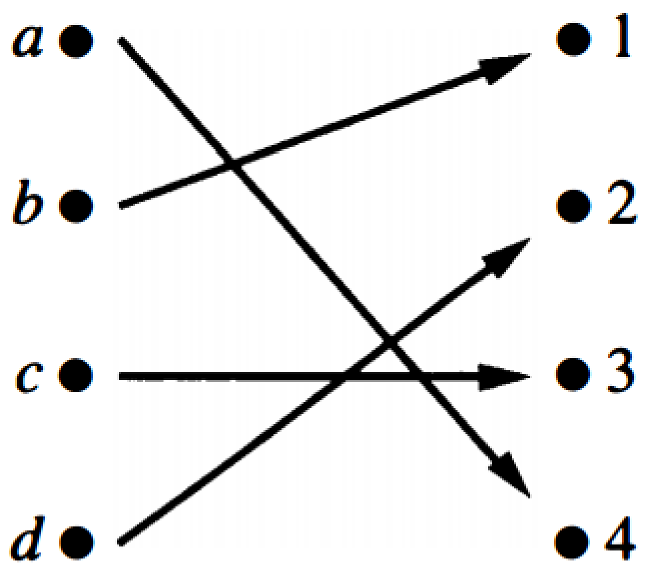
Examples



$A \rightarrow B$

Functions (12/21)

Examples



A → *B*

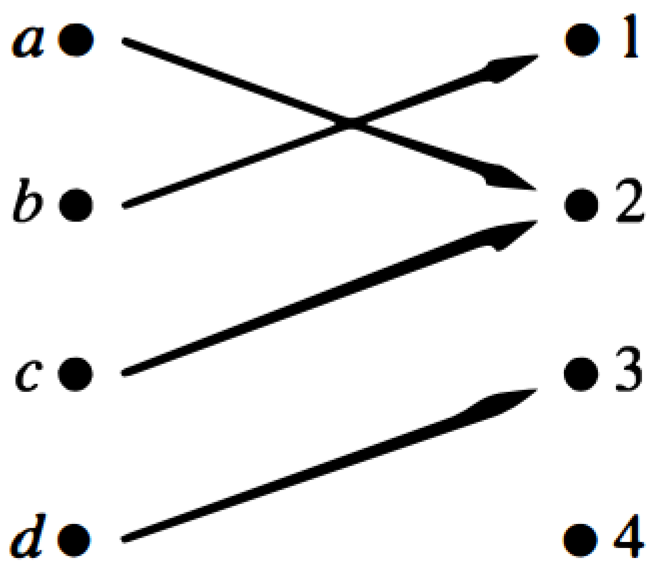
One-to-one

Onto

∴ bijection

Functions (12/21)

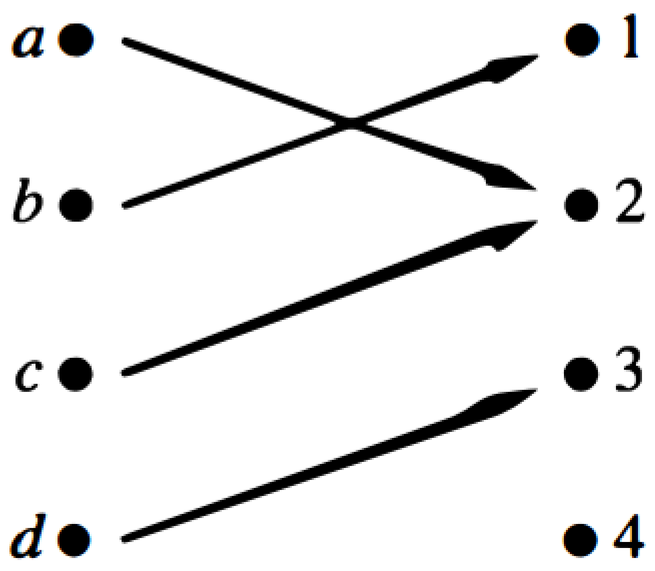
Examples



$$A \rightarrow B$$

Functions (12/21)

Examples



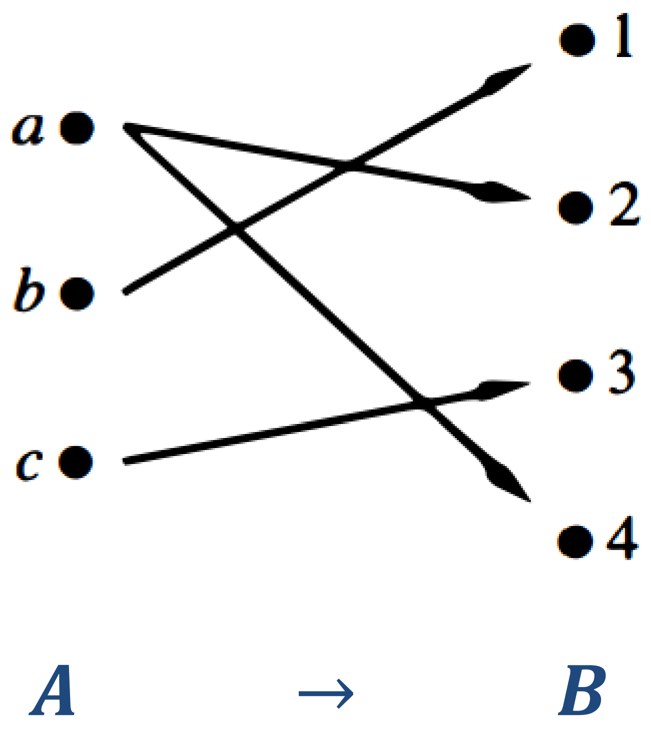
NOT One-to-one

NOT Onto

$A \rightarrow B$

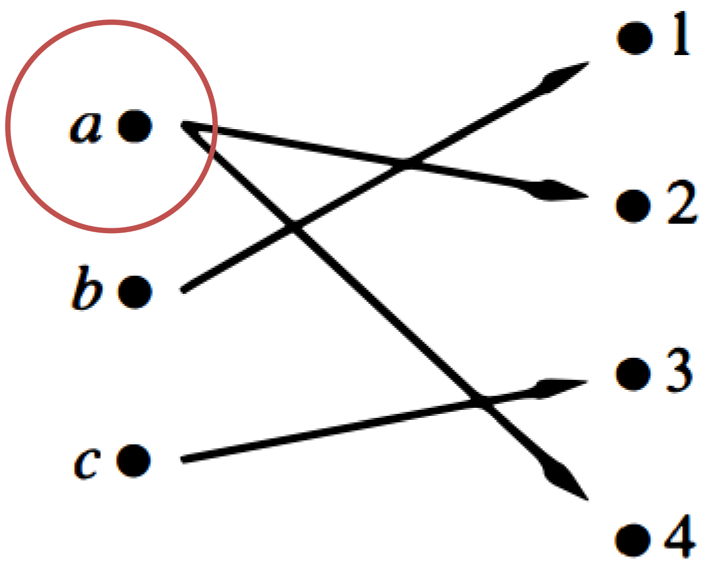
Functions (12/21)

Examples



Functions (12/21)

Examples



$A \quad \rightarrow \quad B$

NOT a function from *A* to *B*

Functions (13/21)

Examples

Determine whether the function $f(x) = x + 1$ from the set of integers to the set of integers is one-to-one.

Functions (13/21)

Examples (Answer)

Determine whether the function $f(x) = x + 1$ from the set of integers to the set of integers is one-to-one.

$f(a) = a + 1$ and $f(b) = b + 1$

$f(x)$is one–to–one (if $f(a) = f(b)$ and a equal b then).

$$a + 1 = b + 1$$

$$a = b$$

$$\boxed{\therefore f(x) \text{ is one–to–one}}$$

Functions (14/21)

Examples

Determine whether the function $f(x) = x^2$ from the set of integers to the set of integers is one-to-one.

Functions (14/21)

Examples (Answer)

Determine whether the function $f(x) = x^2$ from the set of integers to the set of integers is one-to-one.

$f(a) = a^2$ and $f(b) = b^2$

$f(x)$is one–to–one (if $f(a) = f(b)$ and a equal b then).

$$a^2 = b^2$$

$$\pm a = \pm b$$

$$a \text{ may be not equal } b$$

$$\therefore f(x) \text{ is NOT one–to–one}$$

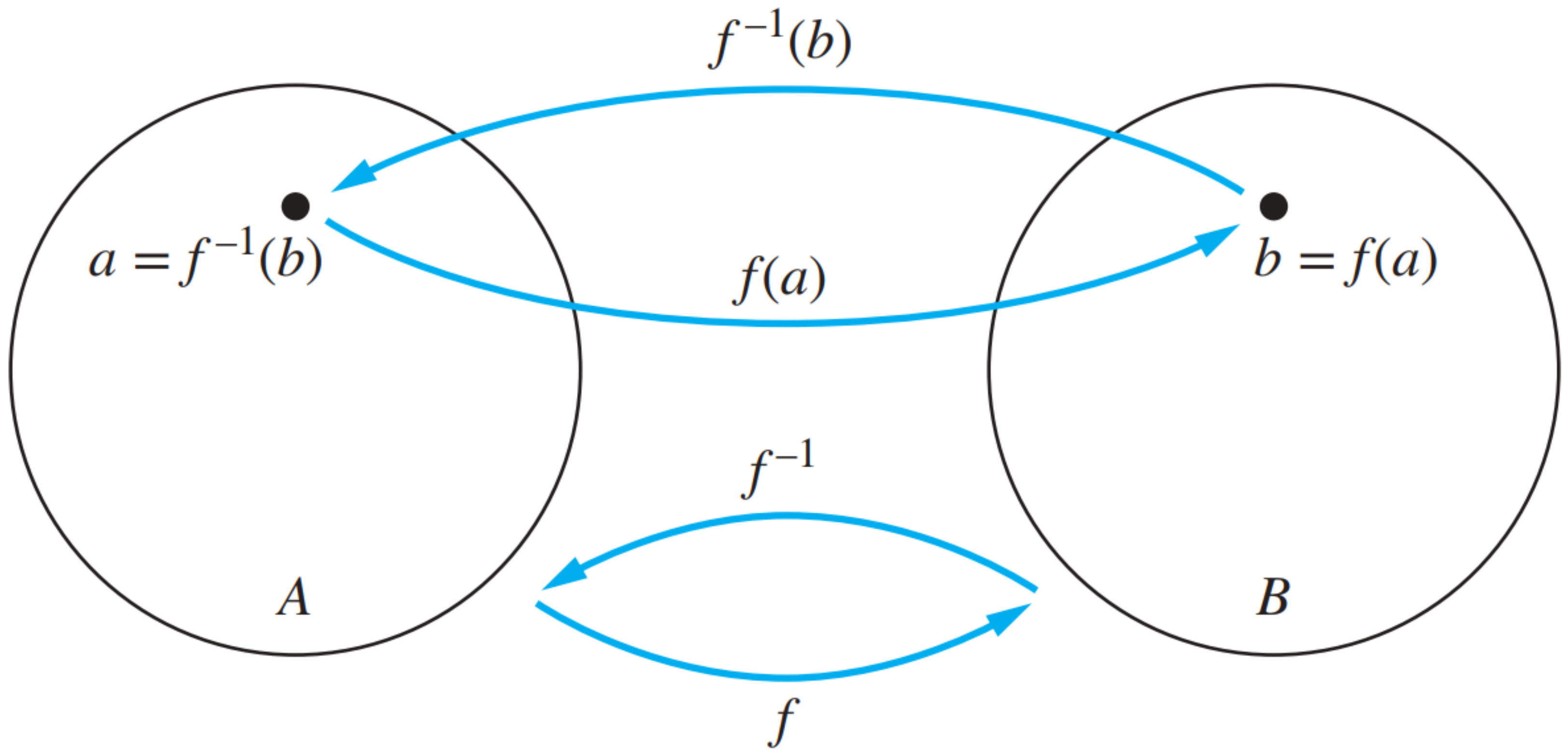
Functions (15/21)

Inverse Functions

Let f be a $\boxed{\textit{one-to-one correspondence}}$ from the set A to the set B. The **inverse** function of f is the function that assigns to an element b belonging to B the unique element a in A such that $f(a) = b$. The inverse function of f is denoted by $\boldsymbol{f^{-1}}$. Hence, $f^{-1}(b) = a$ when $f(a) = b$.

Functions (15/21)

Inverse Functions



Functions (16/21)

Invertible

A one-to-one correspondence is called **invertible** because we can define an inverse of this function. A function is **not invertible** if it is not a one-to-one correspondence, because the inverse of such a function does not exist.

Functions (17/21)

Invertible – Example

Let f be the function from $\{a, b, c\}$ to $\{1, 2, 3\}$ such that $f(a) = 2$, $f(b) = 3$, and $f(c) = 1$. Is f invertible, and if it is, what is its inverse?

Functions (17/21)

Invertible – Example

Let f be the function from $\{a, b, c\}$ to $\{1, 2, 3\}$ such that $f(a) = 2$, $f(b) = 3$, and $f(c) = 1$. Is f invertible, and if it is, what is its inverse?

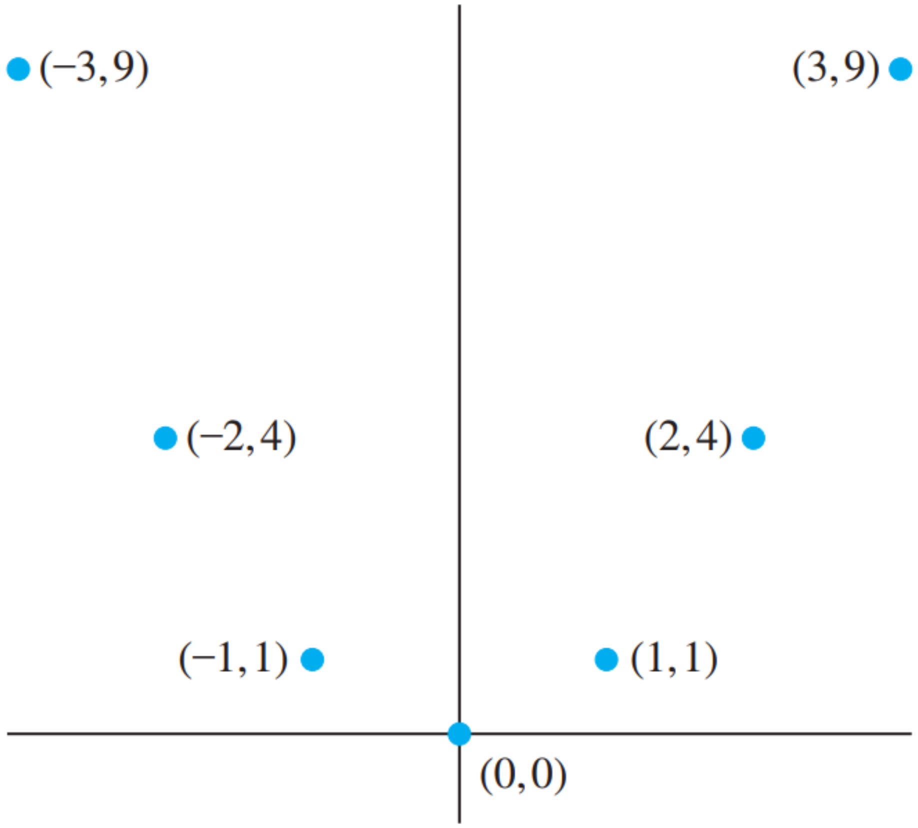
Answer:

The function f is invertible because it is a one-to-one correspondence. The inverse function f^{-1} reverses the correspondence given by f, so $f^{-1}(1) = c$, $f^{-1}(2) = a$, and $f^{-1}(3) = b$.

Functions (21/21)

The Graphs of Functions

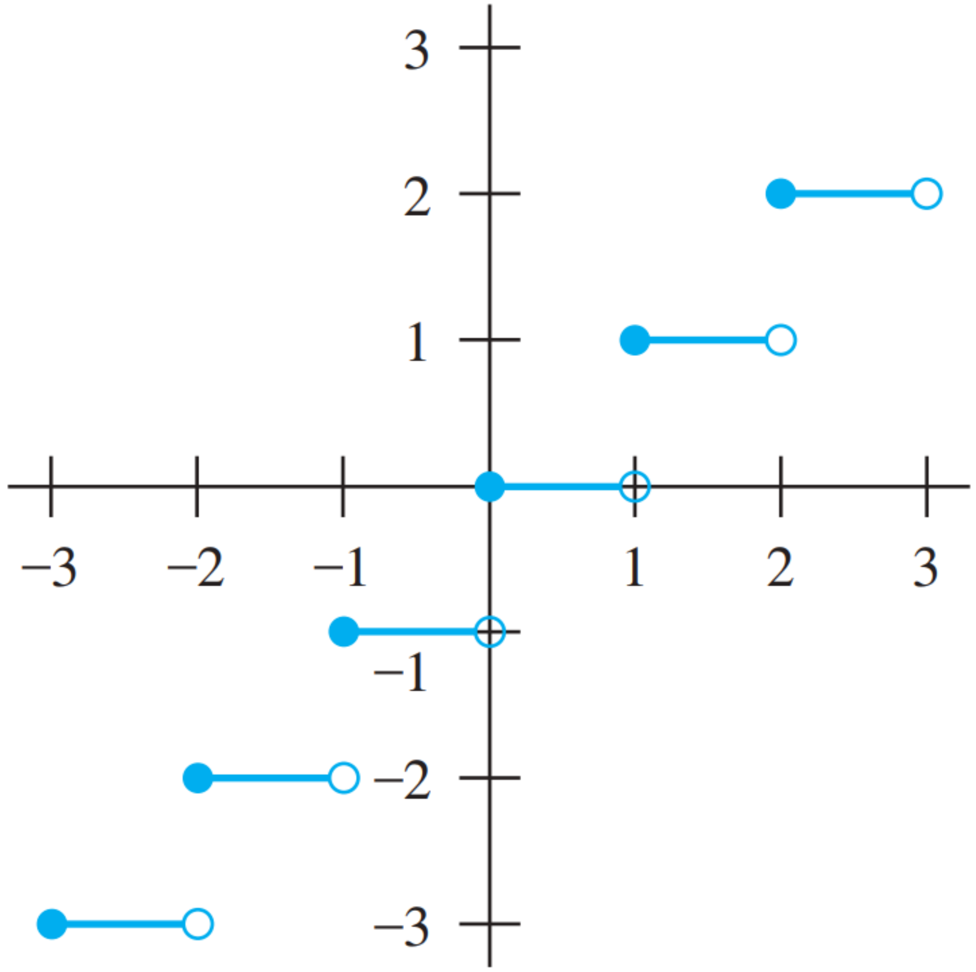
Let f be a function from A to B. The graph of the function f is the set of ordered pairs $\{(a, b) | \ a \in A \text{ and } b \in B\}$.



The graph of $f(x) = x^2$ from Z to Z.

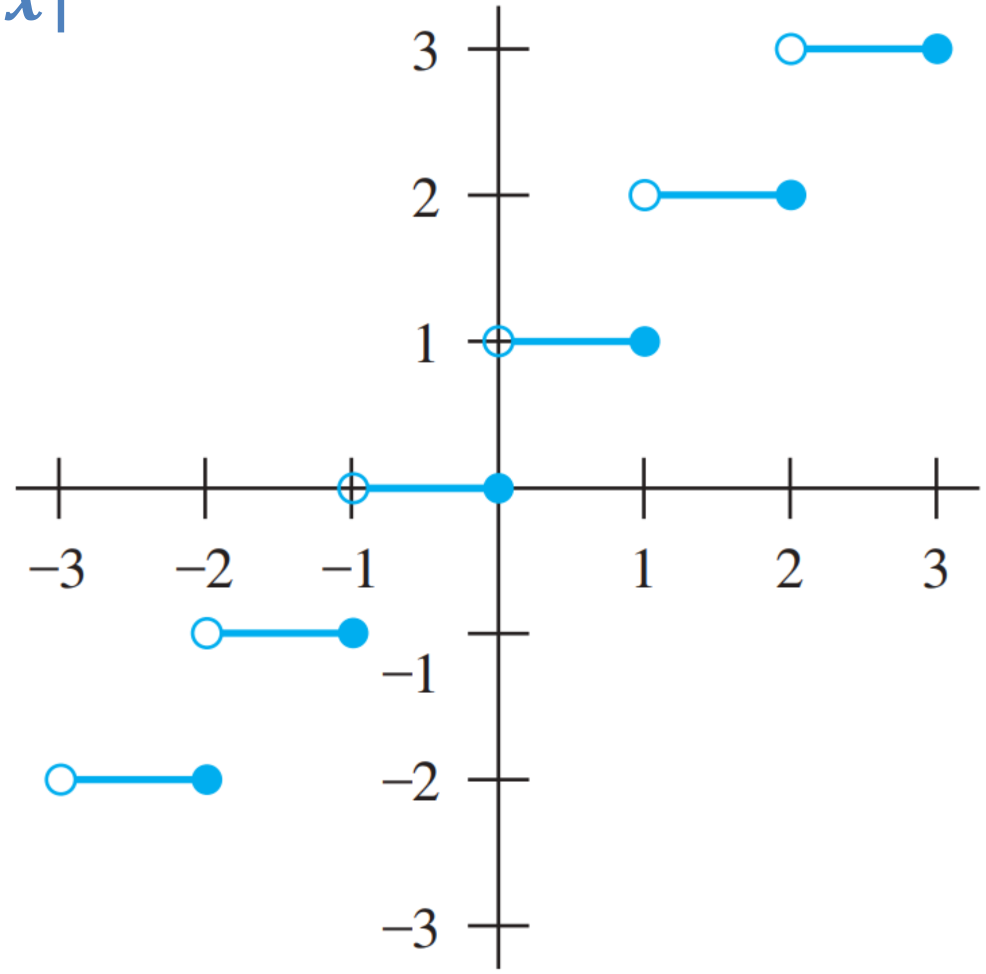
Some Important Functions (1/4)

Floor function $y = \lfloor x \rfloor$



Some Important Functions (2/4)

Ceiling function $y = \lceil x \rceil$



Some Important Functions (3/4)

Useful Properties

$$\lfloor -x \rfloor = -\lceil x \rceil$$
$$\lceil -x \rceil = -\lfloor x \rfloor$$

$$\lfloor x + n \rfloor = \lfloor x \rfloor + n$$
$$\lceil x + n \rceil = \lceil x \rceil + n$$

Some Important Functions (4/4)

Examples

$\lfloor 0.5 \rfloor =$

$\lceil 0.5 \rceil =$

$\lceil 3 \rceil =$

$\lfloor -0.5 \rfloor =$

$\lceil -1.2 \rceil =$

$\lfloor 1.1 \rfloor =$

$\lfloor 0.3 + 2 \rfloor =$

$\left\lceil 1.1 + \lceil 0.5 \rceil \right\rceil =$

Some Important Functions (4/4)

Examples-Answer

$\lfloor 0.5 \rfloor = 0$

$\lceil 0.5 \rceil = 1$

$\lceil 3 \rceil = 3$

$\lfloor -0.5 \rfloor = -\lceil 0.5 \rceil = -1$

$\lceil -1.2 \rceil = -1$

$\lfloor 1.1 \rfloor = 1$

$\lfloor 0.3 + 2 \rfloor = 2$

$\left\lceil 1.1 + \lceil 0.5 \rceil \right\rceil = 3$

Thank You

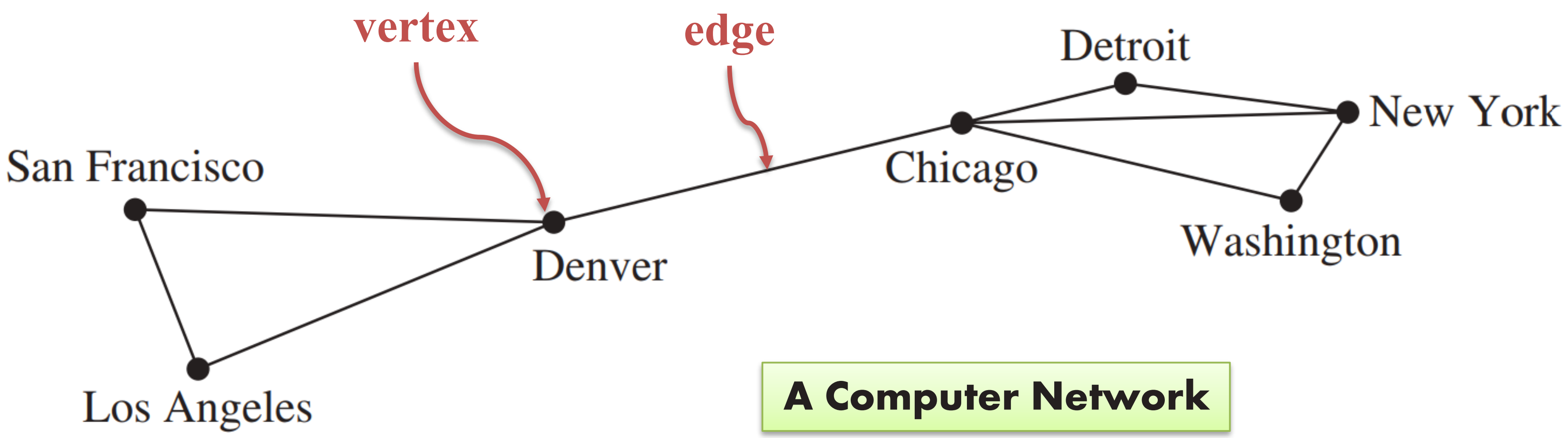
Dr. Alaa H. Jarah

Discrete Structures

Lec_8 Graphs

Definition:

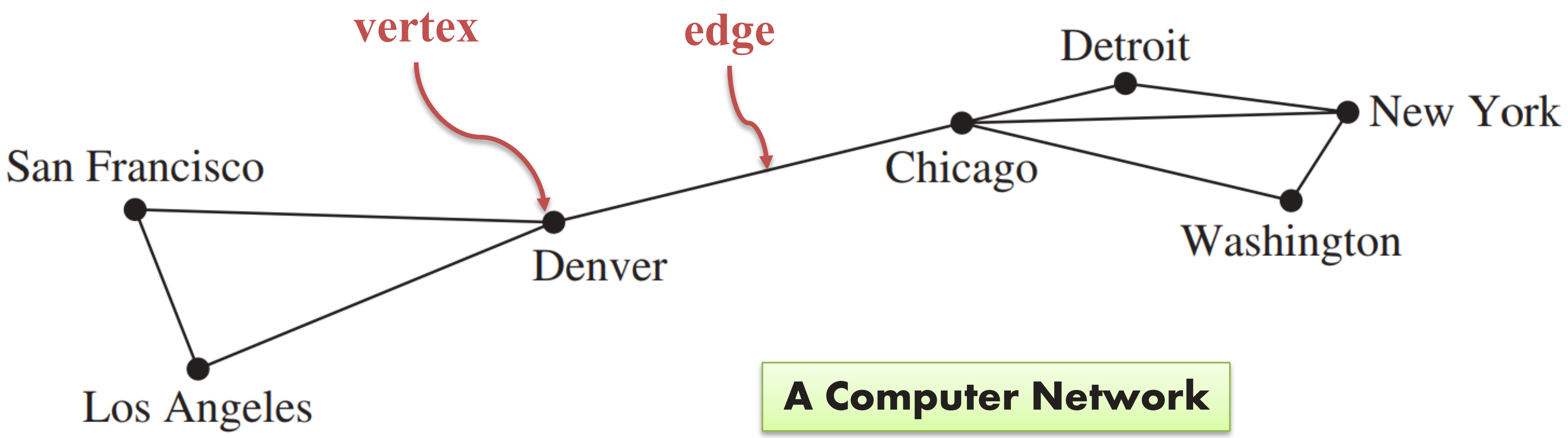
A graph $G = (V, E)$ consists of V, a nonempty set of **vertices** (or **nodes**) and E, a set of **edges**. Each edge has either one or two vertices associated with it, called its **endpoints**. An edge is said to **connect** its endpoints.



A Computer Network

Remark:

The set of vertices V of a graph G may be infinite. A graph with an *infinite vertex set* or an *infinite number of edges* is called an **infinite graph**, and in comparison, a graph with a *finite vertex* set and a *finite edge* set is called a **finite graph**. In this chapter, we will usually consider only finite graphs.

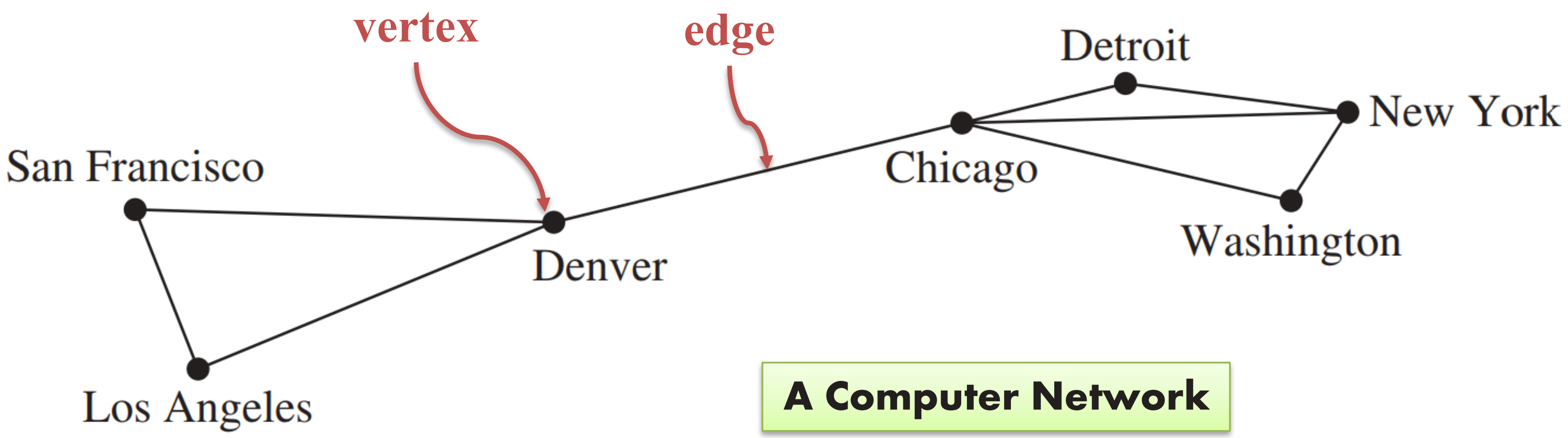


A Computer Network

Graph Terminology

Simple Graph:

Note that each edge of the graph representing this computer network connects two different vertices. 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***.

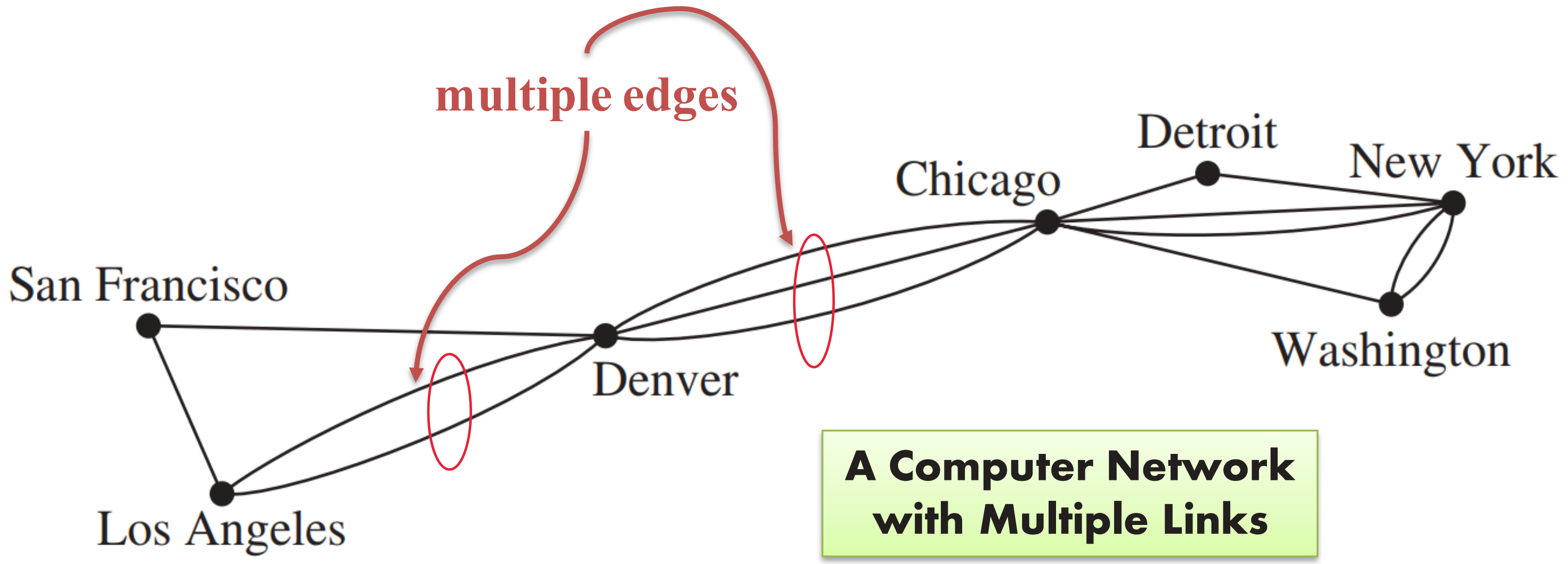


A Computer Network

Graph Terminology

Multigraphs:

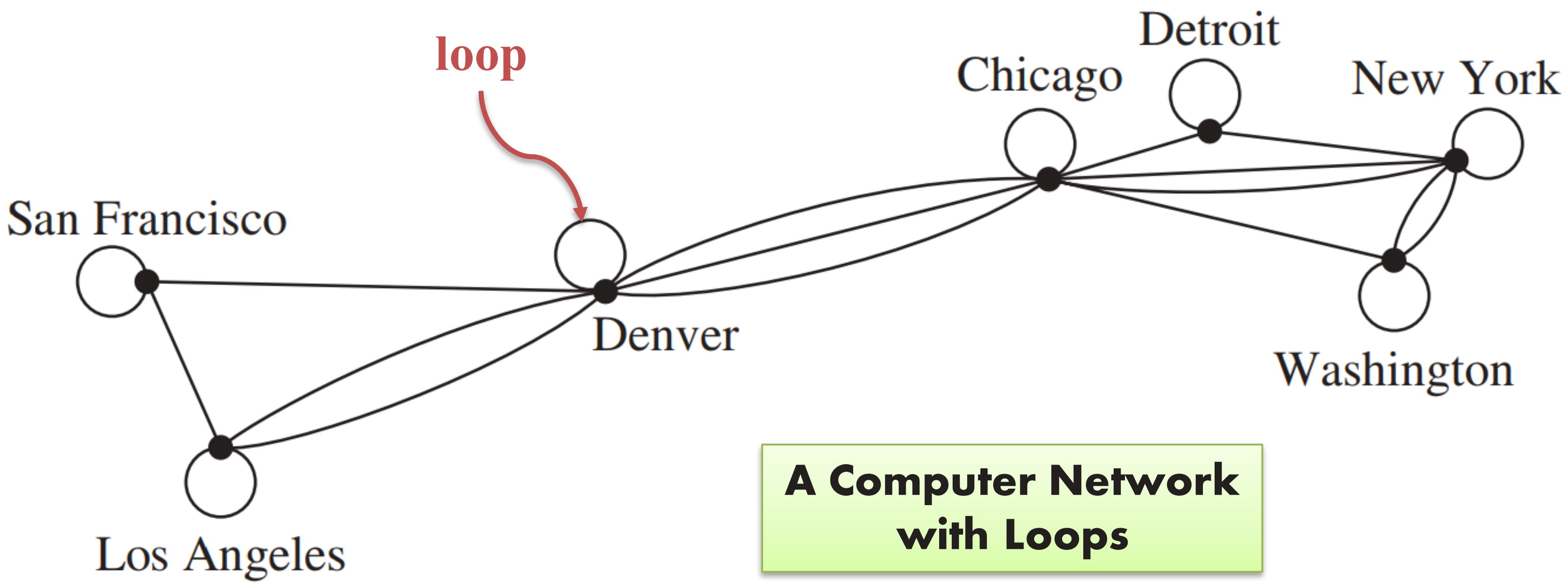
Graphs that may have multiple edges connecting the same vertices are called ***multigraphs***.



Graph Terminology

Loop:

Edges that connect a vertex to itself are called ***loops***.

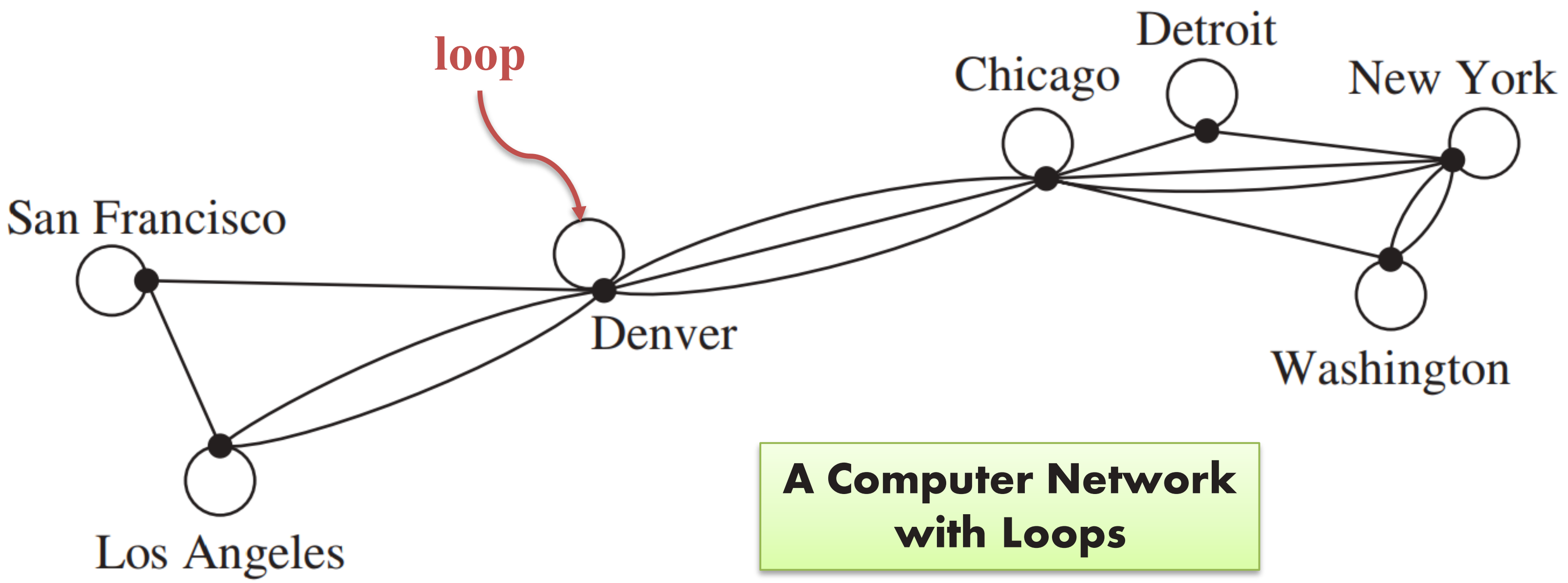


A Computer Network with Loops

Graph Terminology

Pseudographs:

Graphs that may include loops, and possibly multiple edges connecting the same pair of vertices or a vertex to itself, are sometimes called ***pseudographs***.

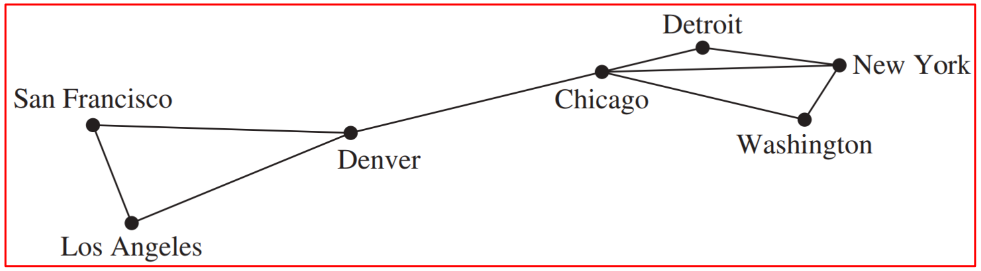


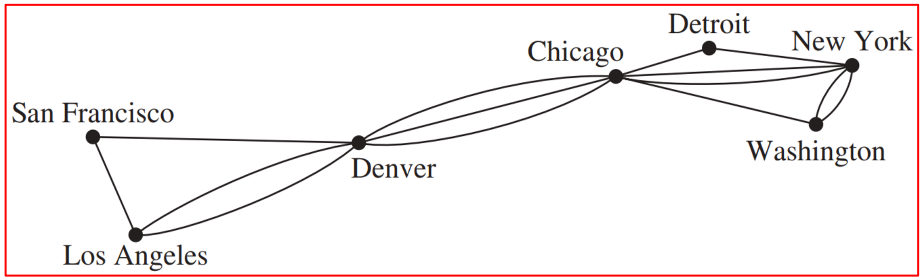
A Computer Network with Loops

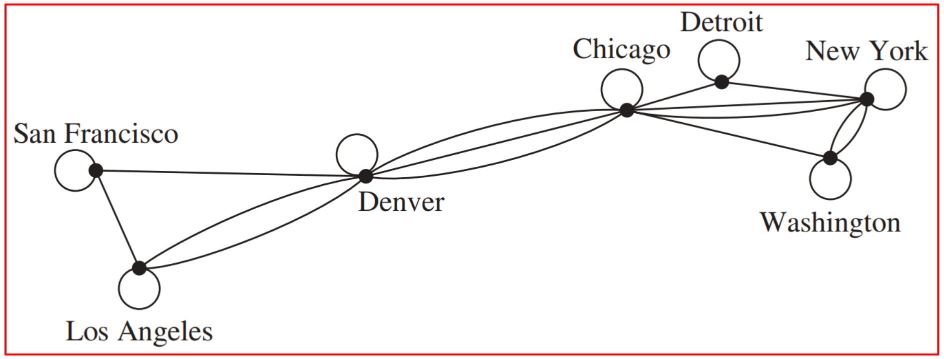
Basic Types of Graphs

Undirected Graphs:

Undirected edges



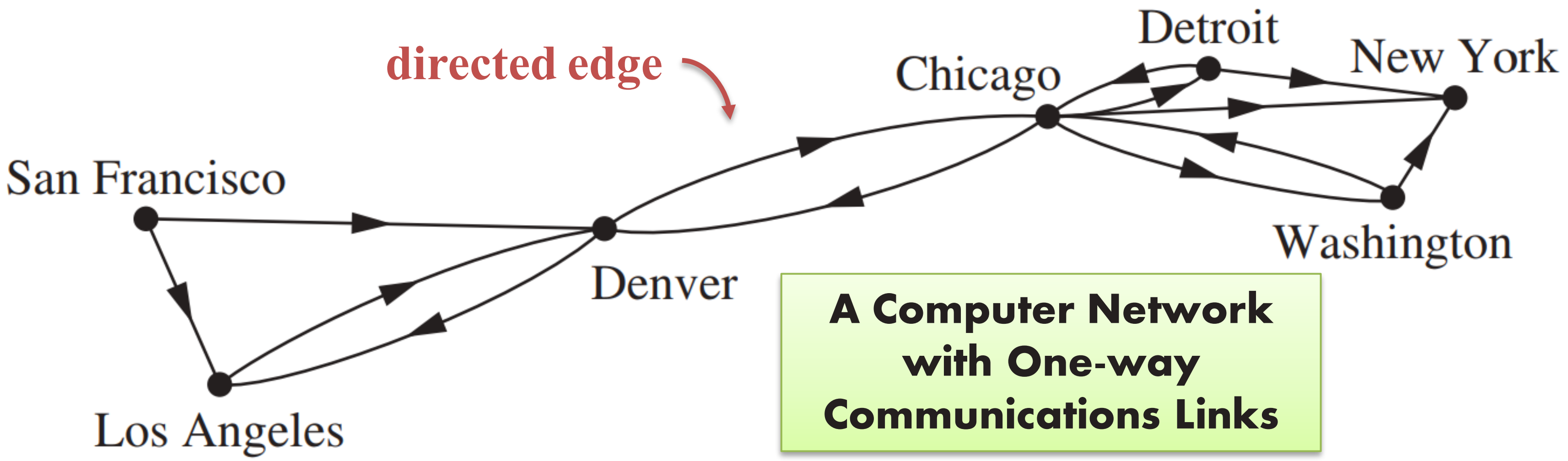




Basic Types of Graphs

Directed Graphs:

A ***directed graph*** (or ***digraph***) (V, E) consists of a nonempty set of vertices V and a set of *directed edges* (or *arcs*) E. Each directed edge is associated with an ordered pair of vertices. The directed edge associated with the ordered pair (u, v) is said to *start* at u and *end* at v.

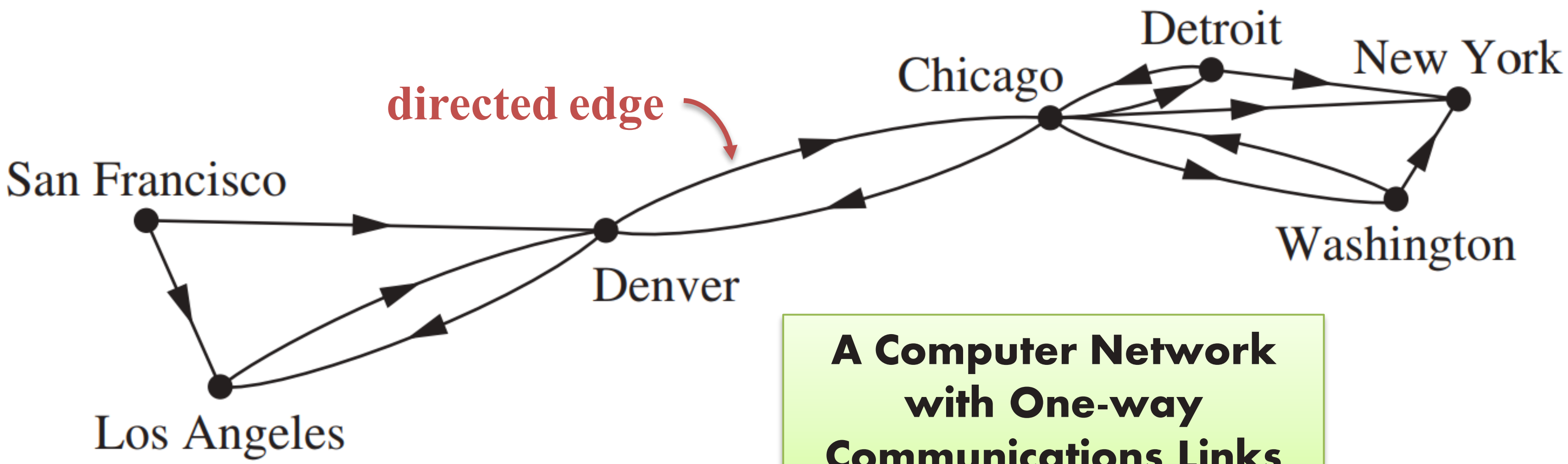


A Computer Network with One-way Communications Links

Basic Types of Graphs

Simple Directed Graph:

When a directed graph has ***no*** *loops* and has ***no*** *multiple directed edges*, it is called a ***simple directed graph***.

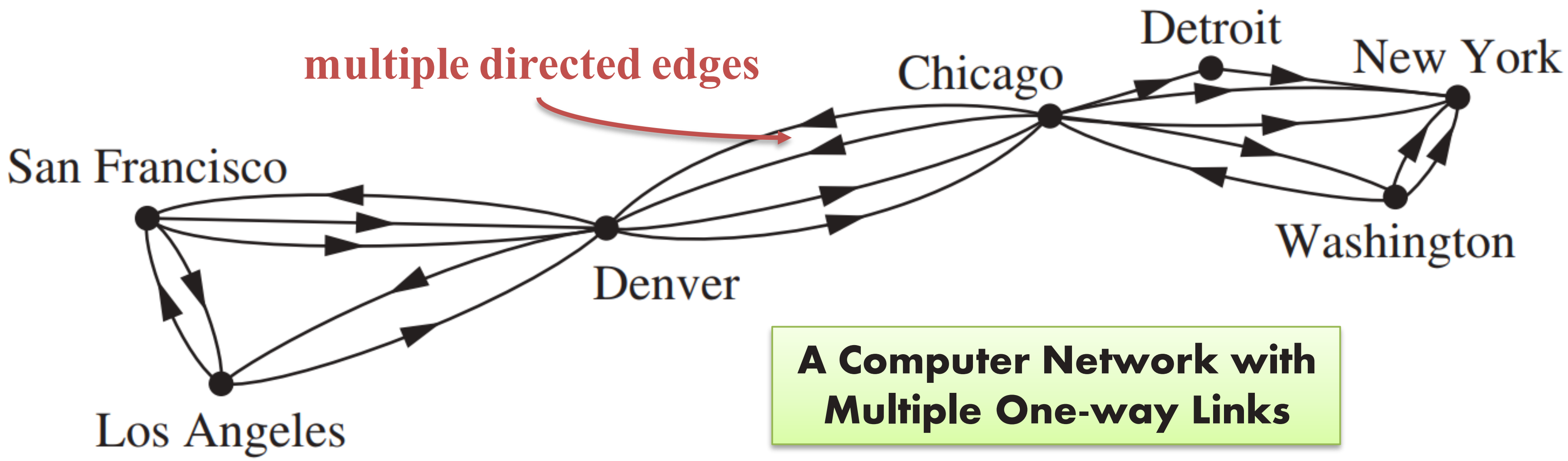


A Computer Network with One-way Communications Links

Basic Types of Graphs

Directed Multigraphs:

Directed graphs that may have *multiple directed edges* from a *vertex* to a *second* (possibly the same) *vertex* are used to model such networks. We called such graphs ***directed multigraphs***.

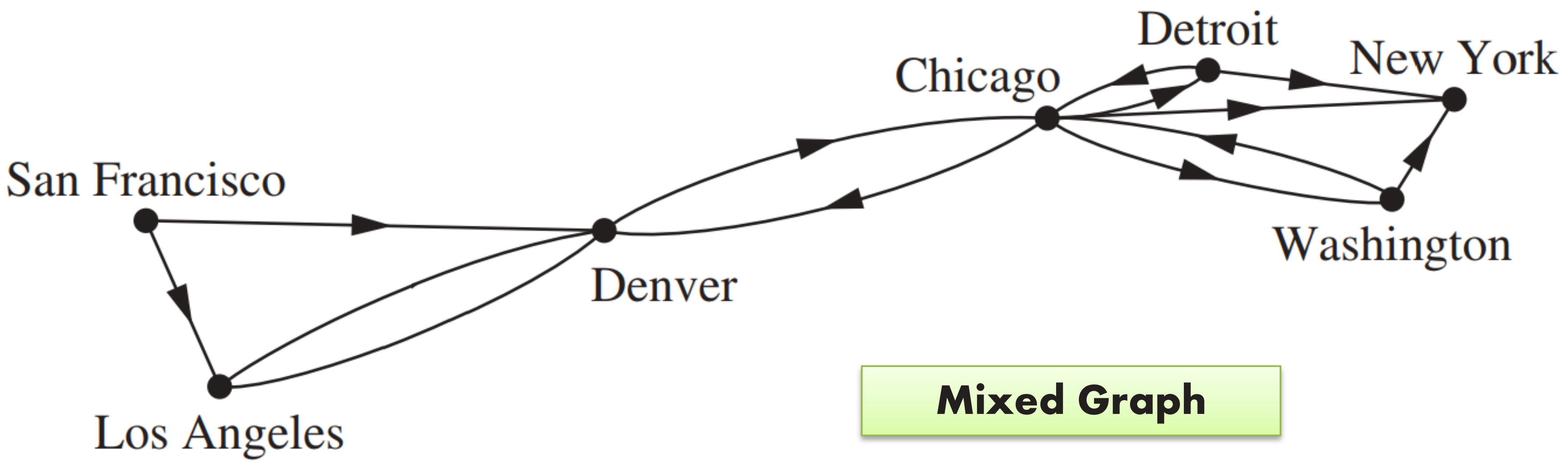


A Computer Network with Multiple One-way Links

Basic Types of Graphs

Mixed Graph:

For some models we may need a graph where some edges are undirected, while others are directed. A graph with both directed and undirected edges is called a ***mixed graph***.



Mixed Graph

TABLE 1 **Graph Terminology.**

Type	*Edges*	*Multiple Edges Allowed?*	*Loops Allowed?*
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudograph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixed graph	Directed and undirected	Yes	Yes

Structure of A Graph:

Three key questions can help us understand the structure of a graph:

1. Are the edges of the graph undirected or directed (or both)?
2. If the graph is undirected, are multiple edges present that connect the same pair of vertices? If the graph is directed, are multiple directed edges present?
3. Are loops present?

Graph Models:

Graphs are used in a wide variety of models.

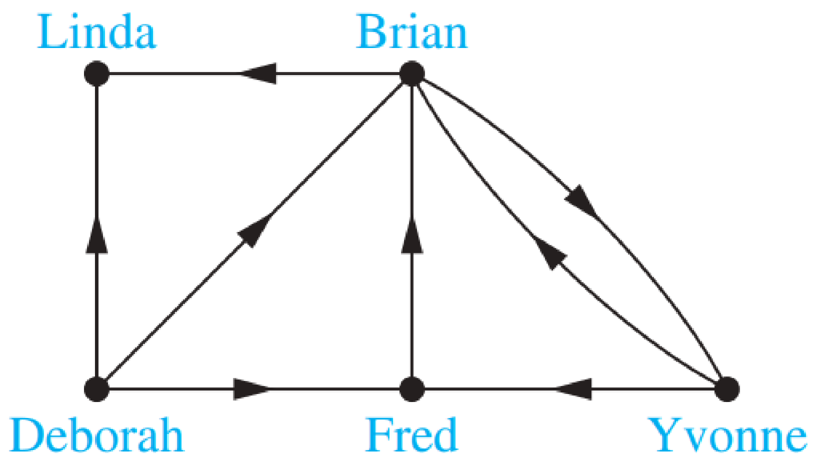
- Social Networks.
- Communication Networks.
- Information Networks.
- Transportation Networks.
- Biological Networks.
- Software Design Applications.
- Tournaments.
- Others…

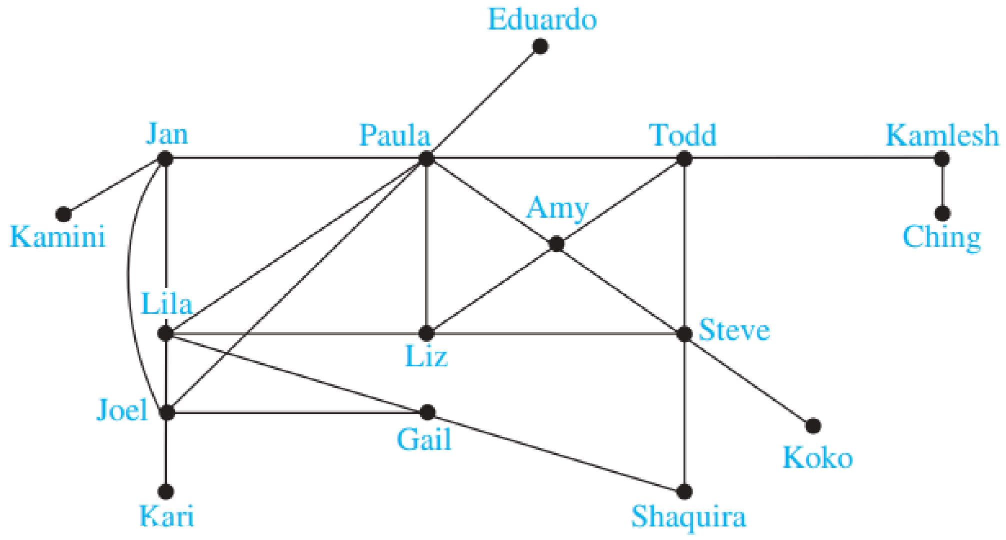
Social Networks:

Graphs are extensively used to model social structures based on different kinds of relationships between people or groups of people.

Friendship Graphs: We can use a simple graph to represent whether two people know each other, that is, whether they are acquainted, or whether they are friends (either in the real world or in the virtual world via a social networking site such as Facebook).

Influence Graphs: In studies of group behavior, it is observed that certain people can influence the thinking of others.



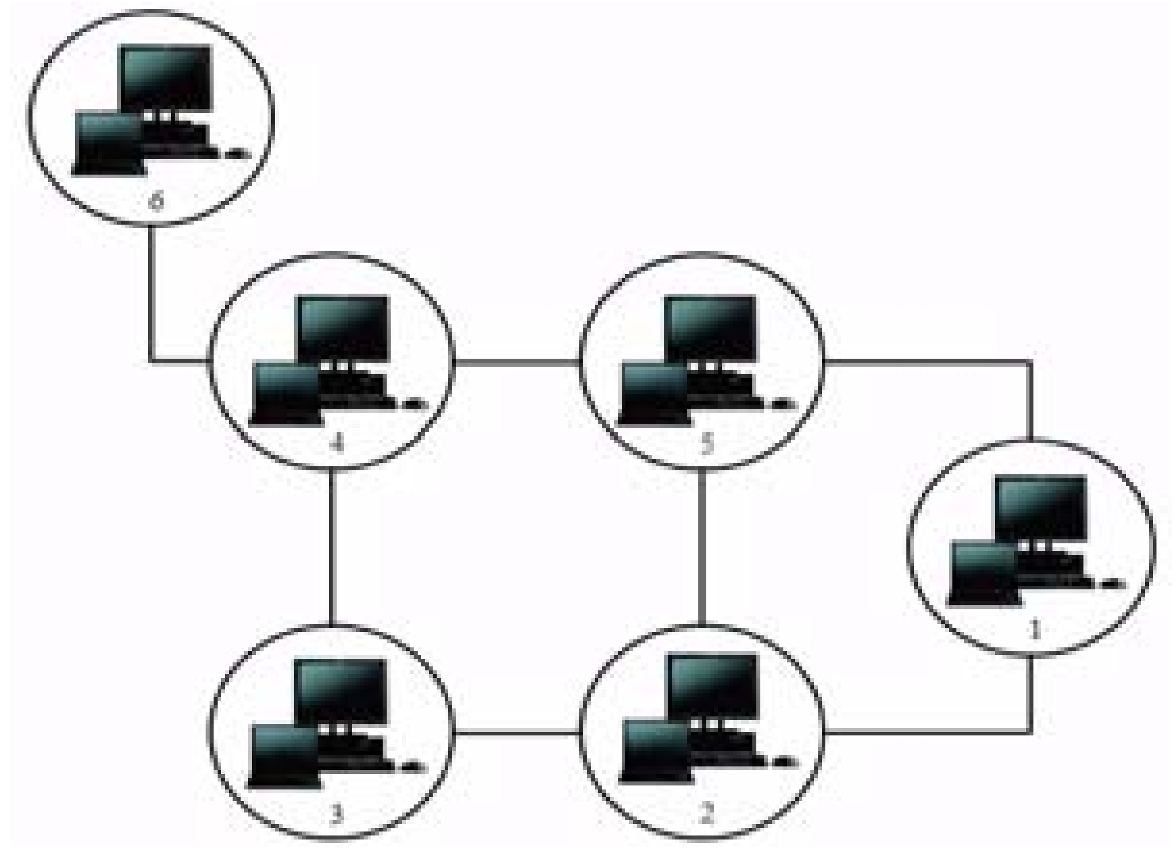


Social Networks:

Collaboration Graphs: is used to model social networks where two people are related by working together in a particular way.

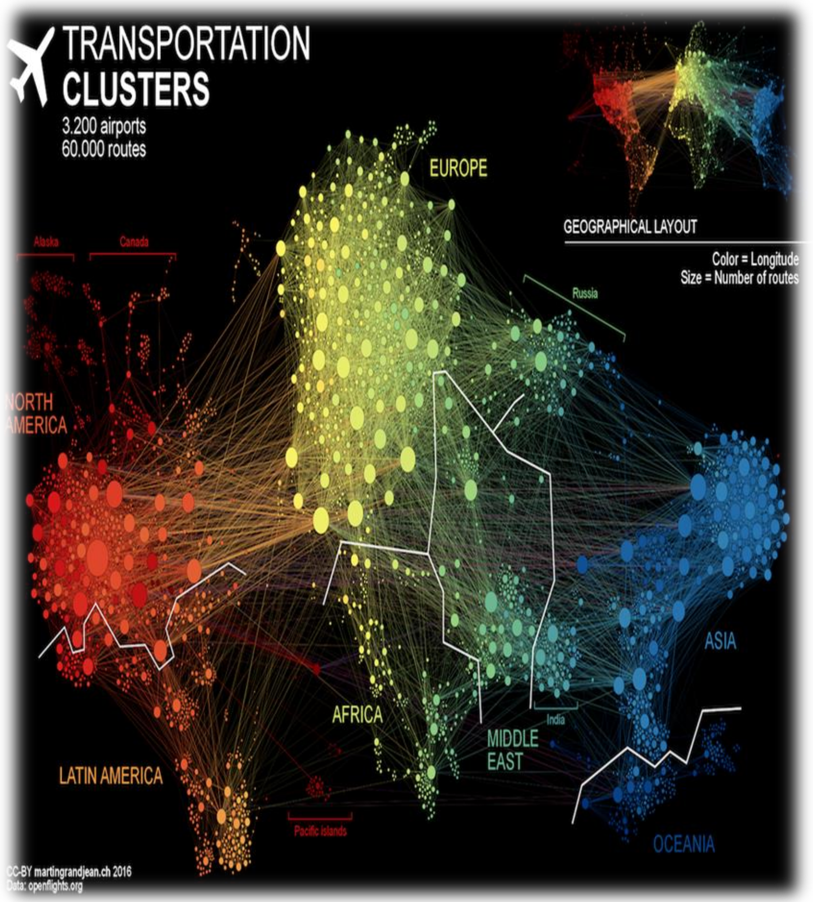
- The Hollywood Links graph is a collaborator graph that represents actors by vertices and connects two actors with an edge if they have worked together on a movie or television show. The Hollywood graph is a huge graph with more than 2.9 million vertices (as of early 2018).
- In an academic collaboration graph, vertices represent people and edges link two people if they have jointly published a paper. The collaboration graph for people who have published research papers in mathematics was found in 2004 to have more than 400,000 vertices and 675,000 edges.

- **Computer networks:**
  - **Nodes – computers**
  - **Edges - connections**



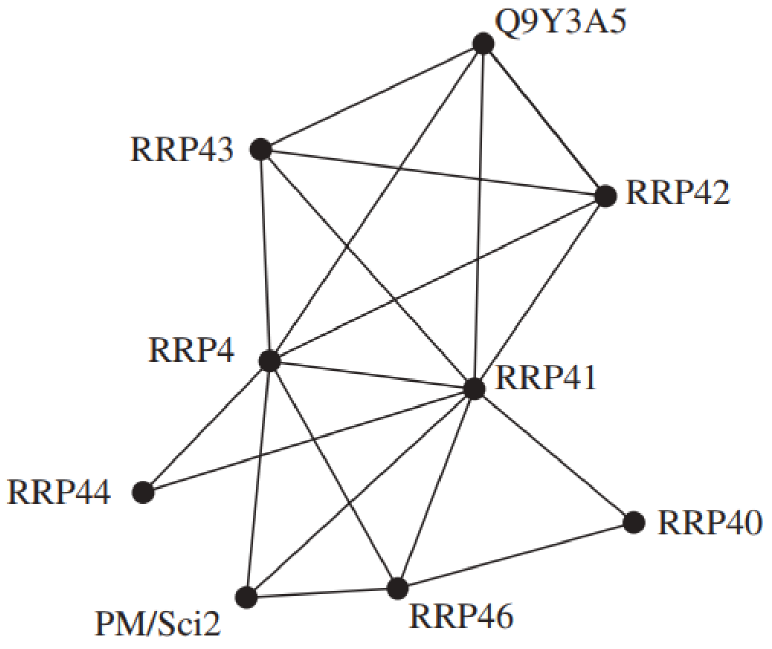
Transportation Networks:

We can use graphs to model many different types of transportation networks, including ***road***, ***air***, and ***rail*** networks, as well as ***shipping*** networks.



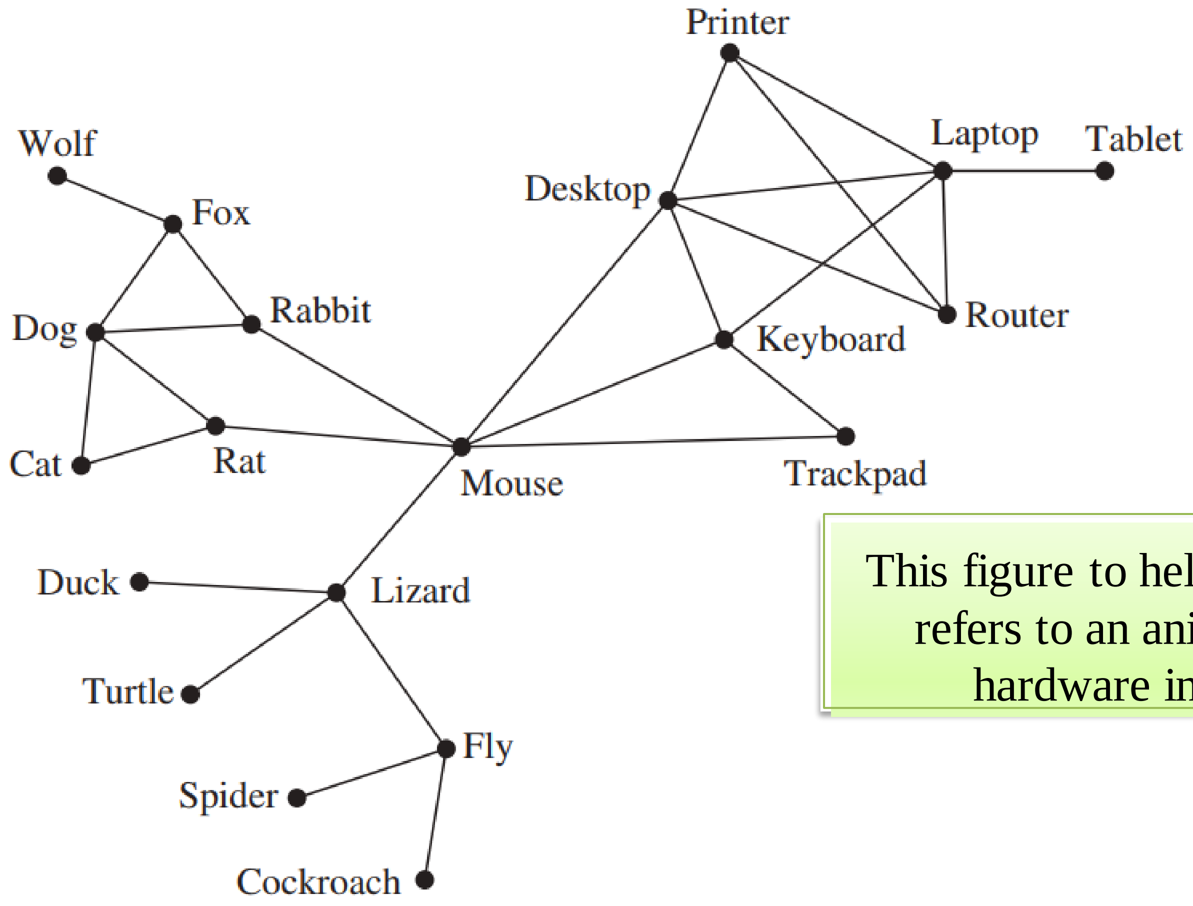
Biological Networks:

Many aspects of the biological sciences can be modeled using graphs. ***Protein Interaction Graphs***: A protein interaction in a living cell occurs when two or more proteins in that cell bind to perform a biological function.



Semantic Networks:

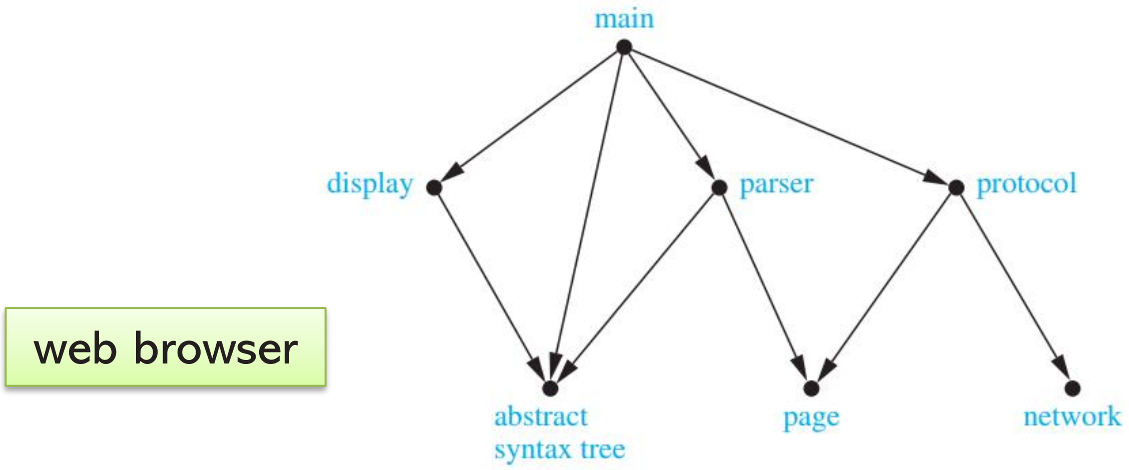
Graph models are used extensively in natural language understanding and in information retrieval. ***Natural language understanding (NLU)*** is the subject of enabling machines to disassemble and parse human speech. Its goal is to allow machines to understand and communicate as humans do.



This figure to help determine ***mouse*** refers to an animal or computer hardware in the sentence.

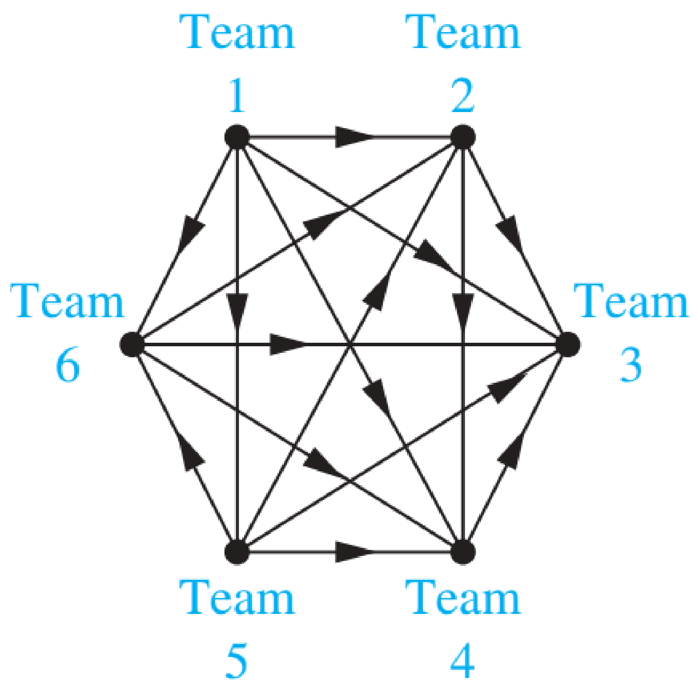
Software Design Applications:

Graph models are useful tools in the design of software. ***Module Dependency Graphs.*** One of the most important tasks in designing software is how to structure a program into different parts, or modules. Understanding how the different modules of a program interact is essential not only for program design, but also for testing and maintenance of the resulting software.



Tournaments:

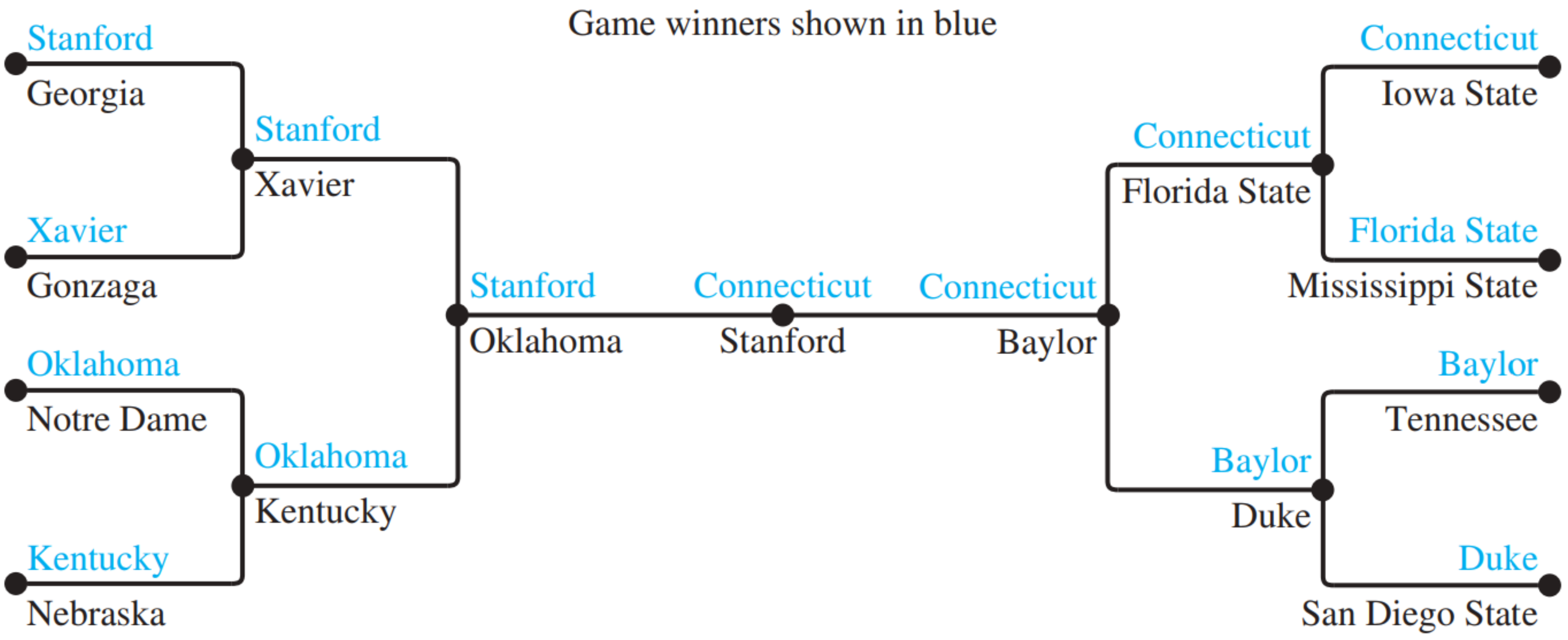
Round-Robin Tournaments. A tournament where each team plays every other team exactly once and no draws are allowed.



We see that Team 1 is undefeated in this tournament, and Team 3 is winless.

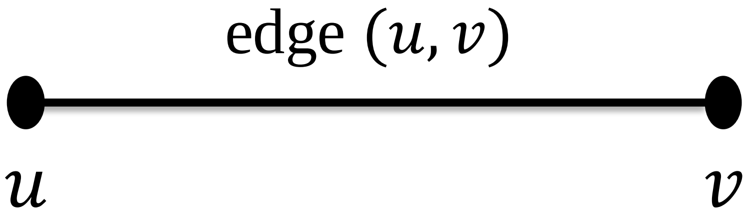
Tournaments:

Single-Elimination Tournaments. A tournament where each contestant is eliminated after one loss



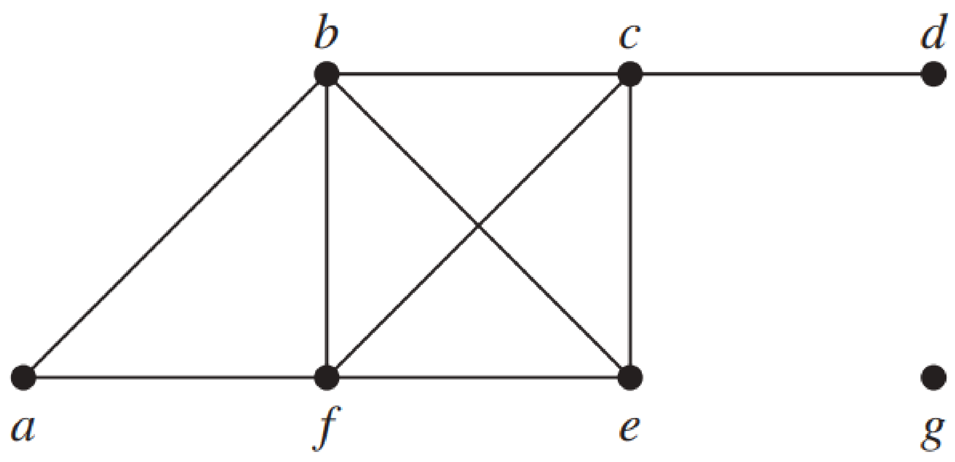
Definition 1:

Two vertices u and v in an undirected graph G are called **adjacent** (or **neighbors**) in G if u and v are endpoints of an edge e of G. Such an edge e is called *incident with* the vertices u and v and e is said to *connect* u and v.



Definition 2:

The set of all neighbors of a vertex v of $G = (V, E)$, denoted by $N(v)$, is called the neighborhood of v. If A is a subset of V, we denote by $N(A)$ the set of all vertices in G that are adjacent to at least one vertex in A. So, $N(A) = \bigcup_{v \in A} N(v)$.



$\boldsymbol{N(a) = \{b, f\}}$

$\boldsymbol{N(b) = \{a, c, e, f\}}$

$\boldsymbol{N(c) = \{b, d, e, f\}}$

$\boldsymbol{N(d) = \{c\}}$

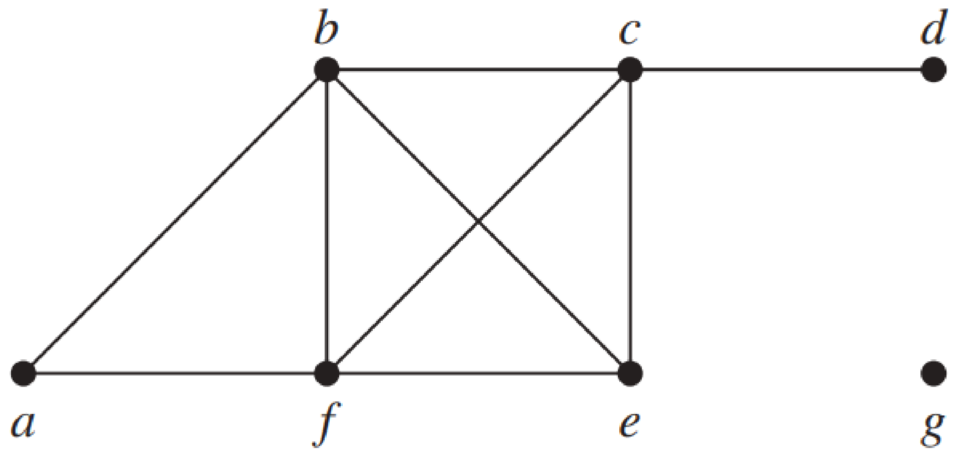
$\boldsymbol{N(e) = \{b, c, f\}}$

$\boldsymbol{N(f) = \{a, b, c, e\}}$

$\boldsymbol{N(\mathbf{g}) = \emptyset}$

Definition 3:

The **degree** of a **vertex** in an *undirected* graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex. The degree of the vertex v is denoted by deg(v).



$\deg(a) = 2$

$\deg(b) = 4$

$\deg(c) = 4$

$\deg(d) = 1$ pendant

$\deg(e) = 3$

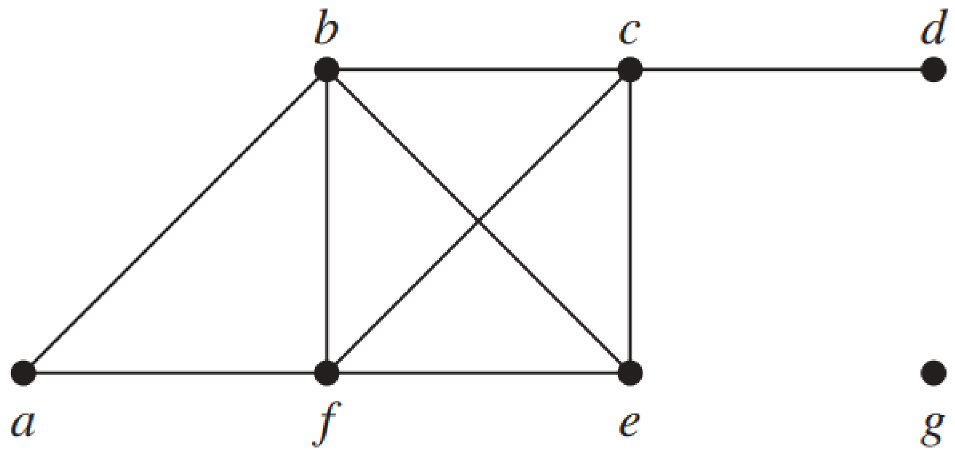
$\deg(f) = 4$

$\deg(g) = 0$ isolated

Isolated:

A vertex of degree zero is called **isolated**. It follows that an isolated vertex is not adjacent to any vertex.

Vertex g is ***isolated***.



$\deg(a) = 2$

$\deg(b) = 4$

$\deg(c) = 4$

$\deg(d) = 1$

$\deg(e) = 3$

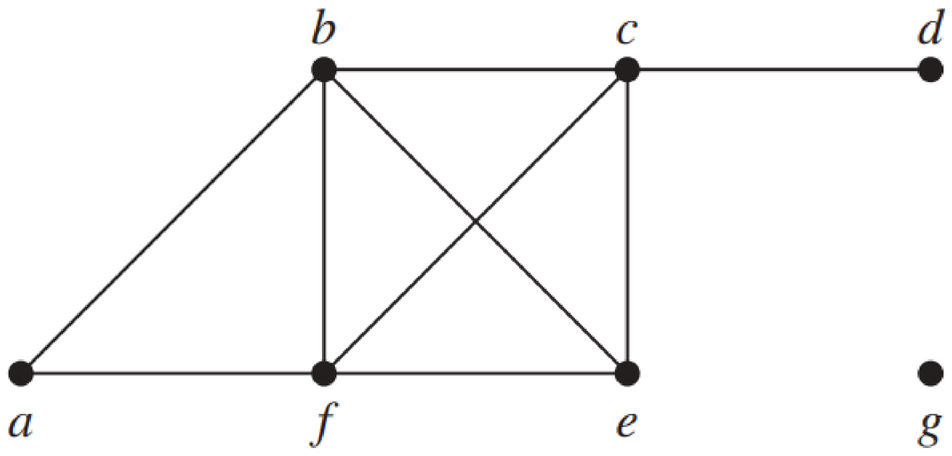
$\deg(f) = 4$

$\boxed{\deg(g) = 0}$

Pendant:

A vertex is **pendant** if and only if it has degree *one*.

Vertex d is ***pendant***.



$\deg(a) = 2$

$\deg(b) = 4$

$\deg(c) = 4$

$\boxed{\deg(d) = 1}$

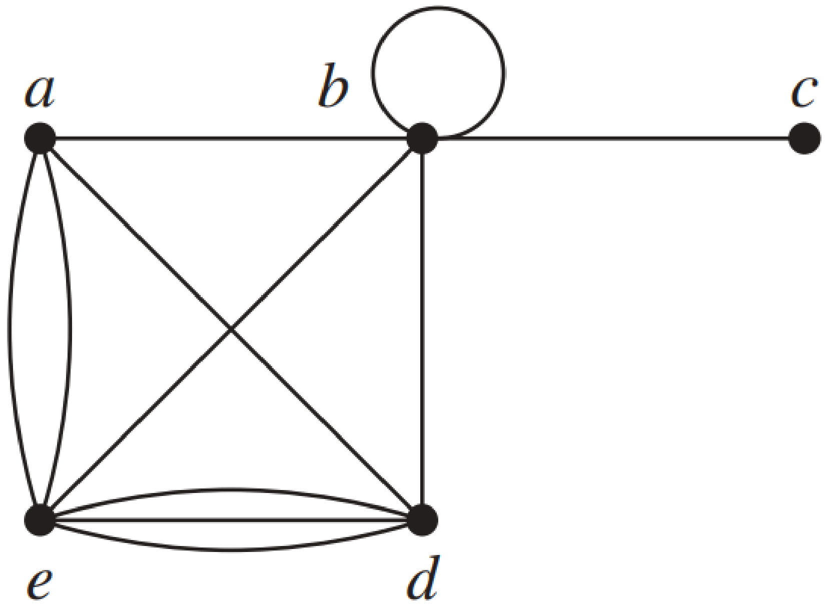
$\deg(e) = 3$

$\deg(f) = 4$

$\deg(g) = 0$

Example 1:

What are the degrees and what are the neighborhoods of the vertices in the following graph?



$\deg(a) =$

$\deg(b) =$

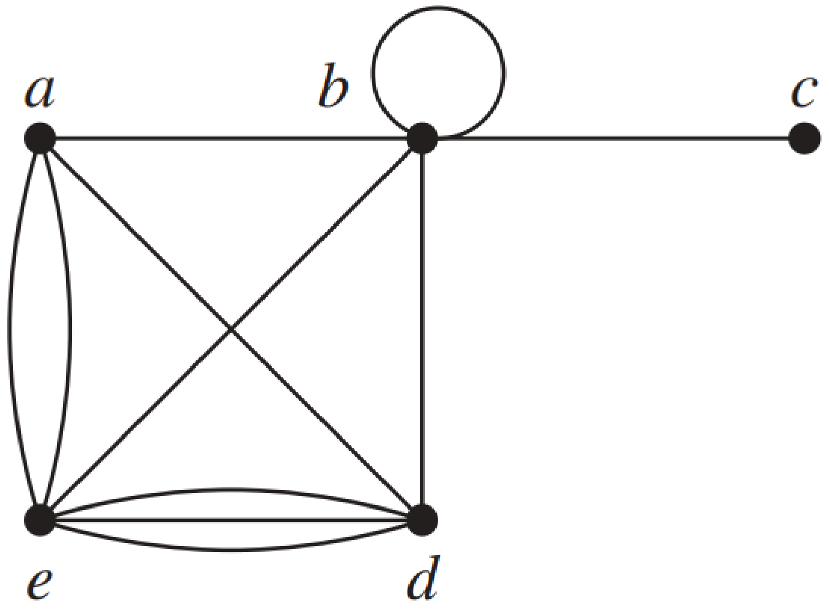
$\deg(c) =$

$\deg(d) =$

$\deg(e) =$

Example 1:

What are the degrees and what are the neighborhoods of the vertices in the following graph?



$\deg(a) = 4$

$\deg(b) = 6$

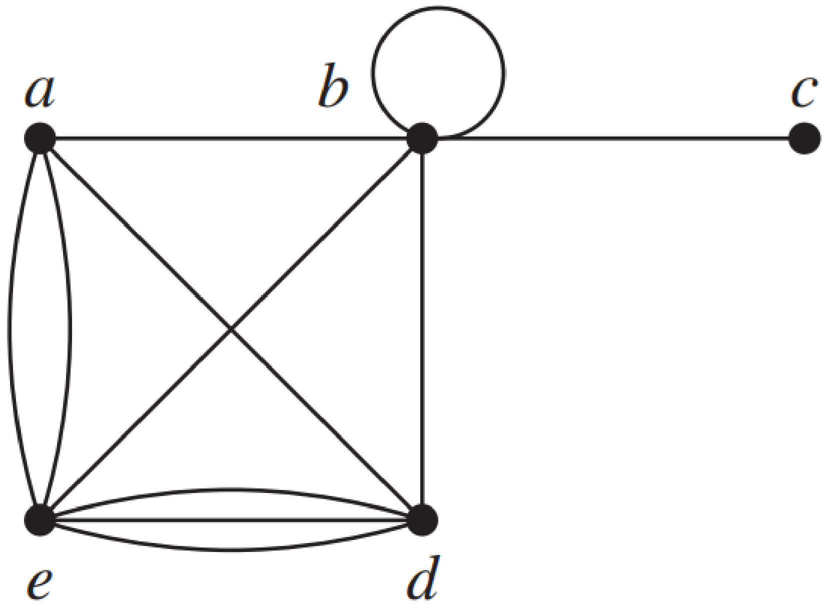
$\deg(c) = 1$

$\deg(d) = 5$

$\deg(e) = 6$

Example 1:

What are the degrees and what are the neighborhoods of the vertices in the following graph?



$\deg(a) = 4$

$\deg(b) = 6$

$\boxed{\deg(c) = 1}$

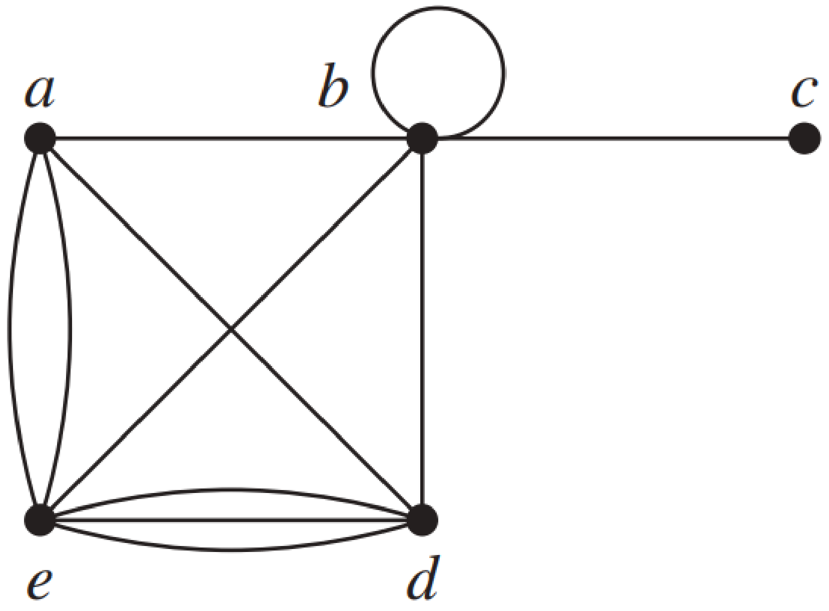
$\deg(d) = 5$

$\deg(e) = 6$

Vertex c is pendant

Example 1:

What are the degrees and what are the neighborhoods of the vertices in the following graph?



$N(a) =$

$N(b) =$

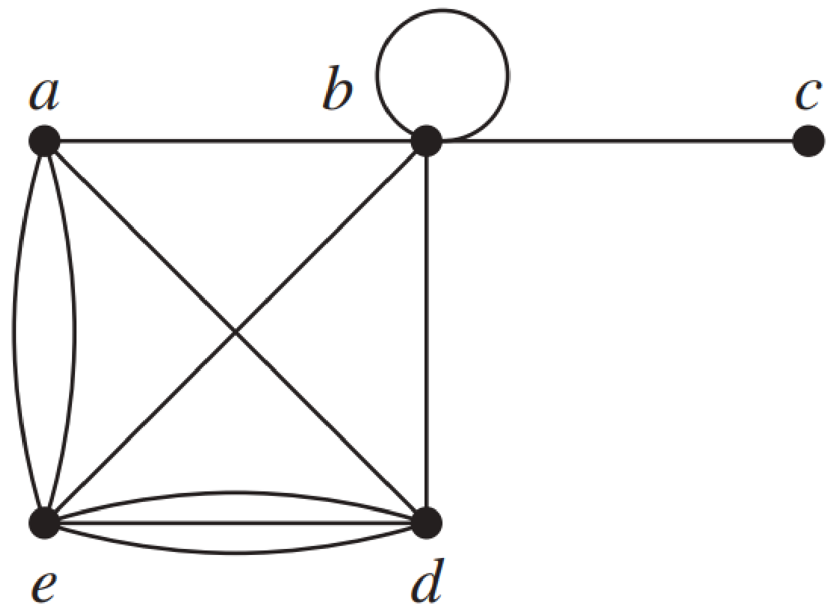
$N(c) =$

$N(d) =$

$N(e) =$

Example 1:

What are the degrees and what are the neighborhoods of the vertices in the following graph?



$N(a) = \{b, d, e\}$

$N(b) = \{a, b, c, d, e\}$

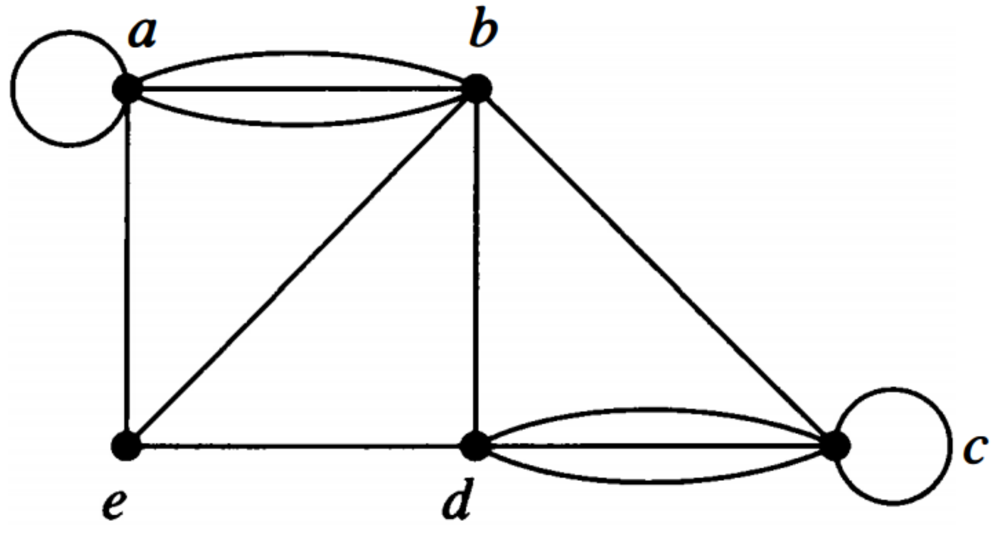
$N(c) = \{b\}$

$N(d) = \{a, b, e\}$

$N(e) = \{a, b, d\}$

Example 2:

What are the degrees and what are the neighborhoods of the vertices in the following graph?

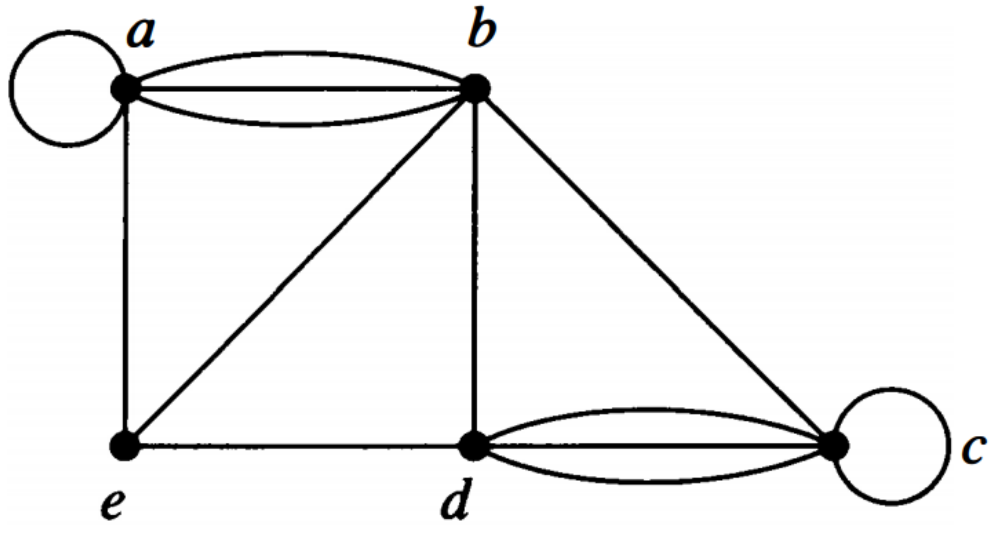


Number of vertices = 5

Number of edges = 13

Example 2:

What are the degrees and what are the neighborhoods of the vertices in the following graph?



$\deg(a) =$

$\deg(b) =$

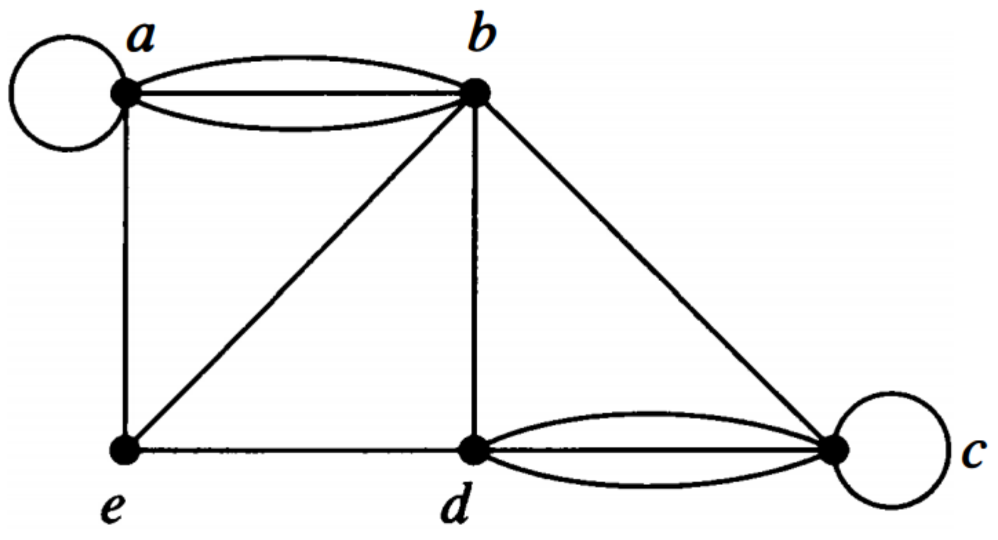
$\deg(c) =$

$\deg(d) =$

$\deg(e) =$

Example 2:

What are the degrees and what are the neighborhoods of the vertices in the following graph?



$\deg(a) = 6$

$\deg(b) = 6$

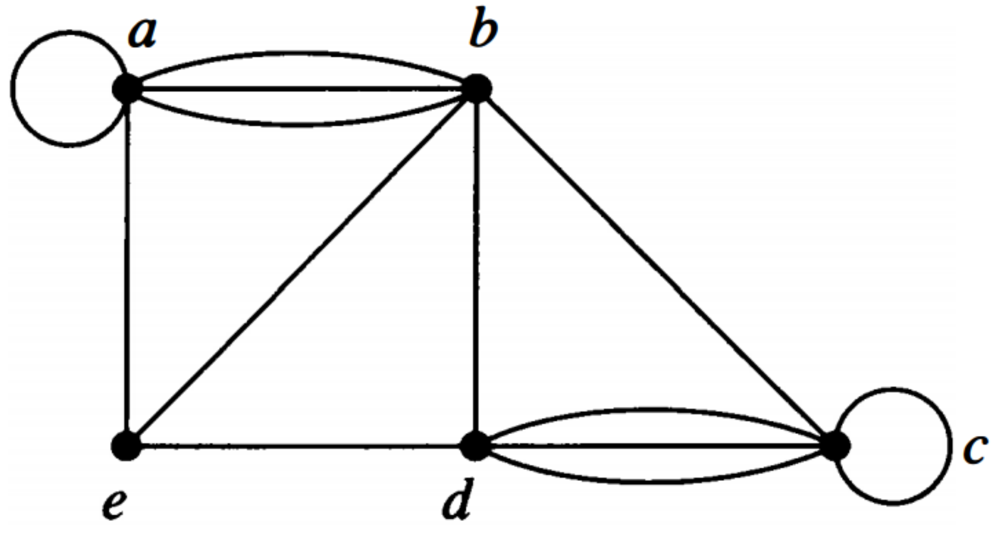
$\deg(c) = 6$

$\deg(d) = 5$

$\deg(e) = 3$

Example 2:

What are the degrees and what are the neighborhoods of the vertices in the following graph?



$N(a) =$

$N(b) =$

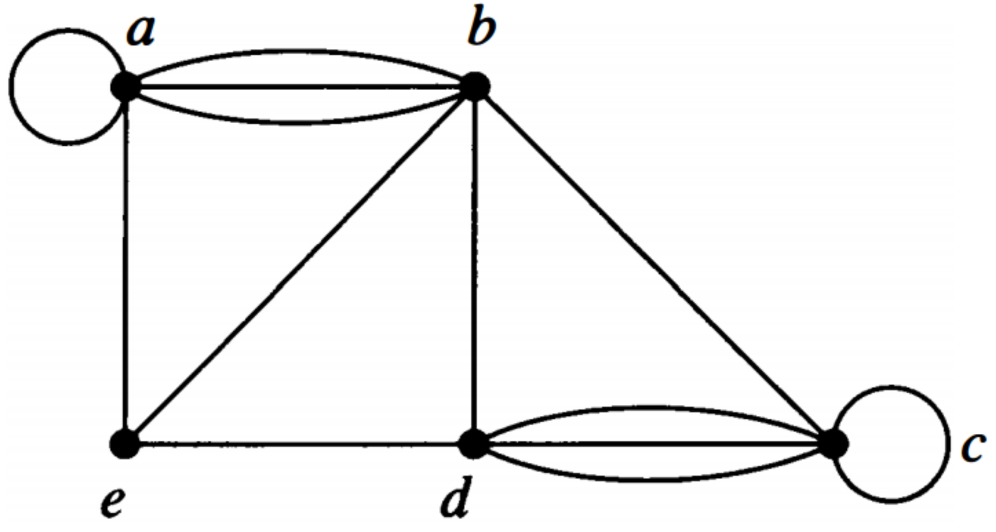
$N(c) =$

$N(d) =$

$N(e) =$

Example 2:

What are the degrees and what are the neighborhoods of the vertices in the following graph?



$N(a) = \{a, b, e\}$

$N(b) = \{a, e, d, c\}$

$N(c) = \{b, c, d\}$

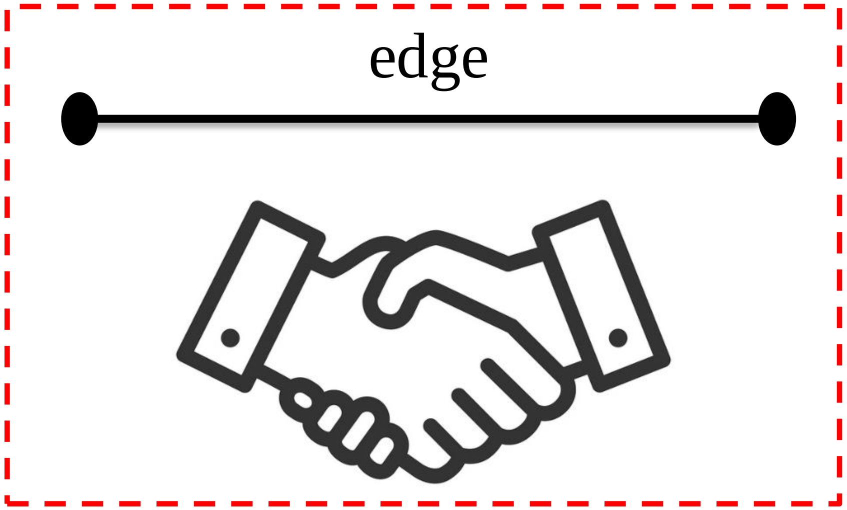
$N(d) = \{e, b, c\}$

$N(e) = \{a, b, d\}$

The Handshaking Theorem:

Let $G = (V, E)$ be undirected graph with m edges. Then

$$2m = \sum_{v \in V} \deg(v)$$



Edge having two endpoints and a handshake involving two hands.

Example 3:

How many edges are there in an undirected graph with 10 vertices each of degree six?

Example 3: Answer

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

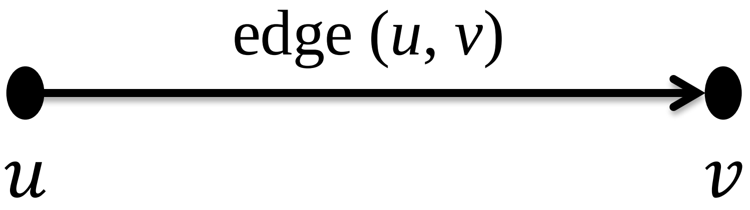
$$2m = \sum_{v \in V} \deg(v)$$

<u>Solution:</u>

Because the sum of the degrees of the vertices is $6 \cdot 10 = 60$, it follows that $2m = 60$. Therefore, $m = 30$.

Definition

When (u, v) is an edge of the graph G with *directed* edges, u is said to be ***adjacent to*** v and v is said to be ***adjacent from*** u. The vertex u is called the **initial vertex** of (u, v), and v is called the **terminal** or **end vertex** of (u, v). The initial vertex and terminal vertex of a loop are the same.



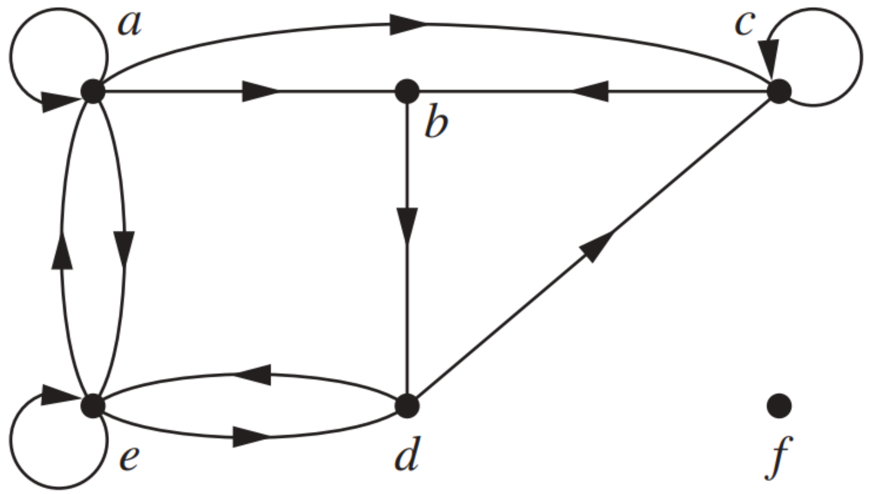
Definition

In a graph with directed edges the **in-degree** of a vertex v, denoted by $\deg^{-}(v)$, is the number of edges with v as their terminal vertex.

The **out-degree** of v, denoted by $\deg^{+}(v)$, is the number of edges with v as their initial vertex.

(Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.)

Example 4:

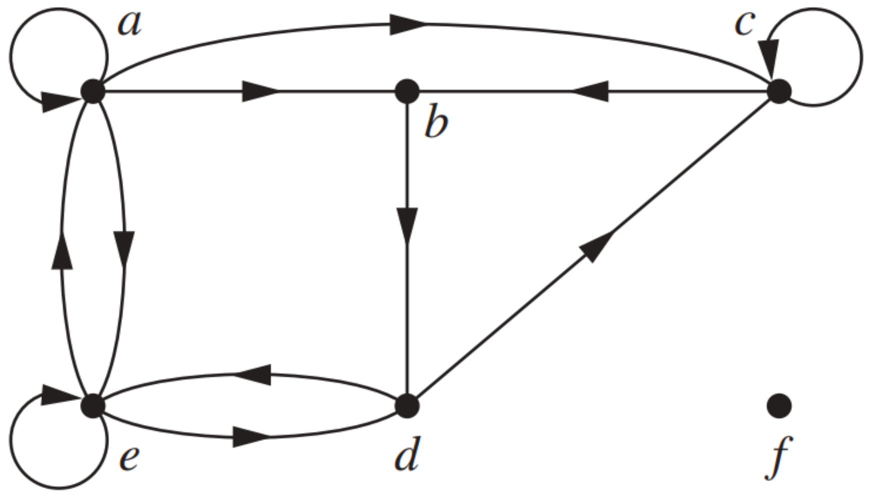


Number of vertices =

Number of edges =

$\deg^-(a) =$	$\deg^+(a) =$
$\deg^-(b) =$	$\deg^+(b) =$
$\deg^-(c) =$	$\deg^+(c) =$
$\deg^-(d) =$	$\deg^+(d) =$
$\deg^-(e) =$	$\deg^+(e) =$
$\deg^-(f) =$	$\deg^+(f) =$

Example 4:



Number of vertices = 6

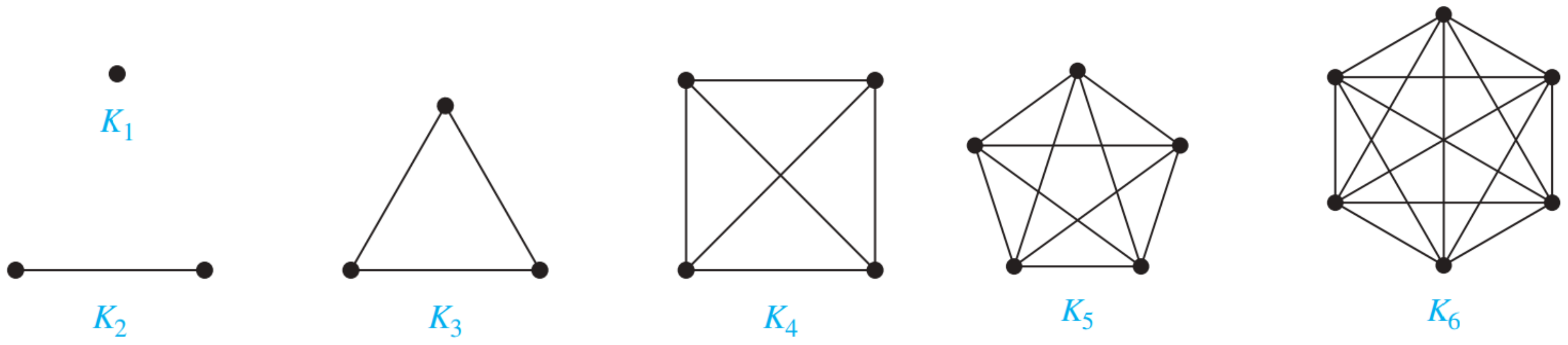
Number of edges = 12

$\deg^-(a) = 2$	$\deg^+(a) = 4$
$\deg^-(b) = 2$	$\deg^+(b) = 1$
$\deg^-(c) = 3$	$\deg^+(c) = 2$
$\deg^-(d) = 2$	$\deg^+(d) = 2$
$\deg^-(e) = 3$	$\deg^+(e) = 3$
$\deg^-(f) = 0$	$\deg^+(f) = 0$

Complete Graphs:

The complete graph on n vertices, denoted by K_n, is the simple graph that contains *exactly one edge between each pair of distinct vertices.*

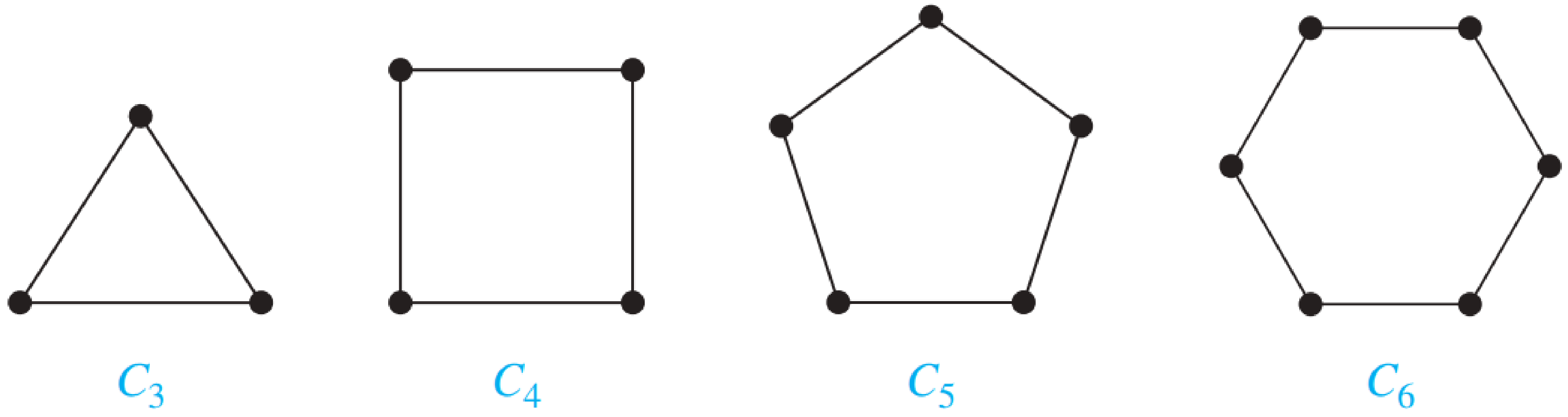
The graphs K_n, for n = 1, 2, 3, 4, 5, 6, are:



Cycles:

The cycle C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}$ and $\{v_n, v_1\}$.

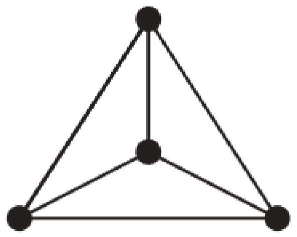
The cycles C_3, C_4, C_5, and C_6 are:

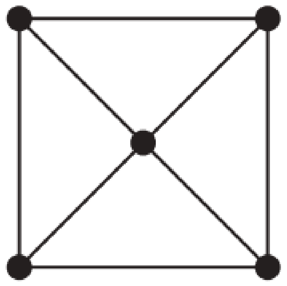


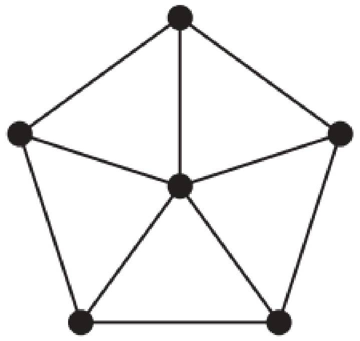
Wheels:

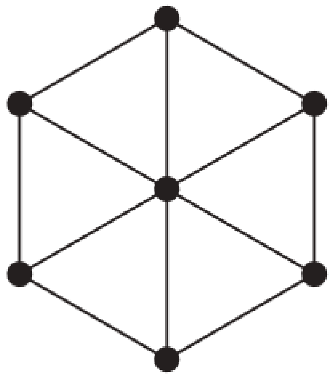
We obtain the wheel W_n when we add an additional vertex to the cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n, by new edges.

The wheels W_3, W_4, W_5, and W_6 are:


W_3


W_4


W_5

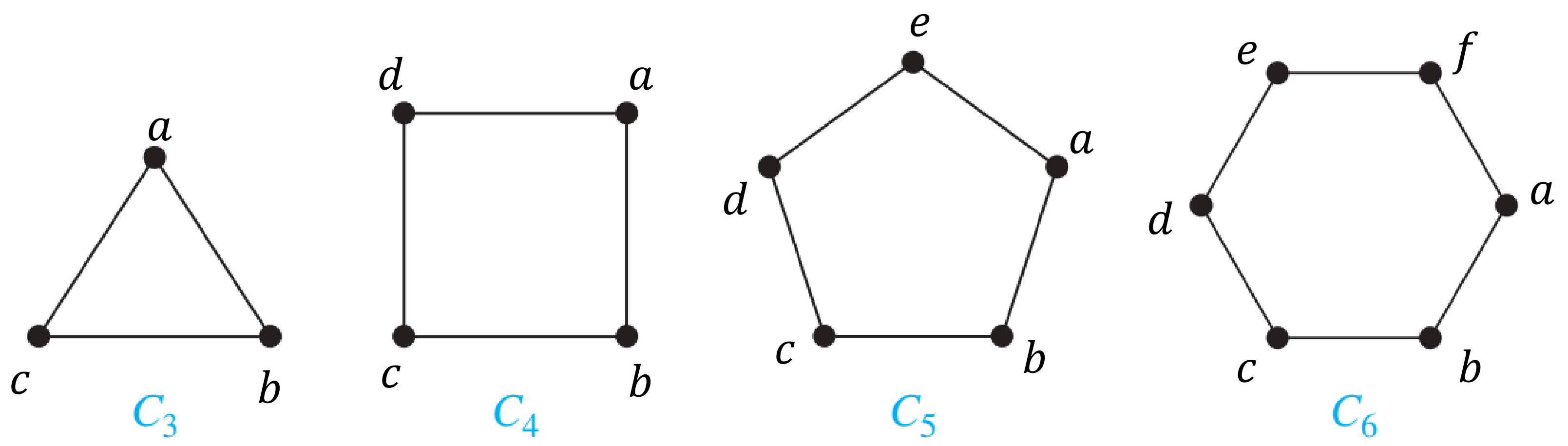

W_6

Definition 6: Bipartite Graphs:

A simple graph G is called **bipartite** if its vertex set V can be partitioned into two disjoint sets V_1 ad V_2 such that every edge in the graph connects a vertex in V_1 and a vertex in V_2 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2). When this condition holds, we call the pair (V_1, V_2) a bipartition of the vertex set V of G.

Example 1:

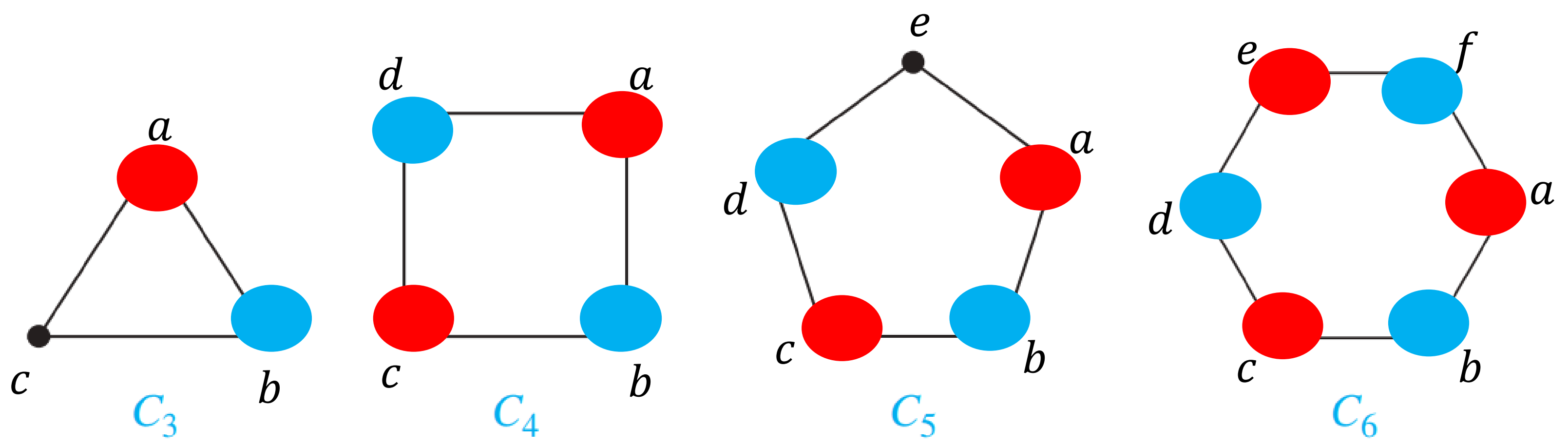
Determine whether the graph is bipartite or not?



Determining whether it is possible to assign either **red** or **blue** to each vertex so that *no two adjacent vertices are assigned the same color*.

Example 1: Answer

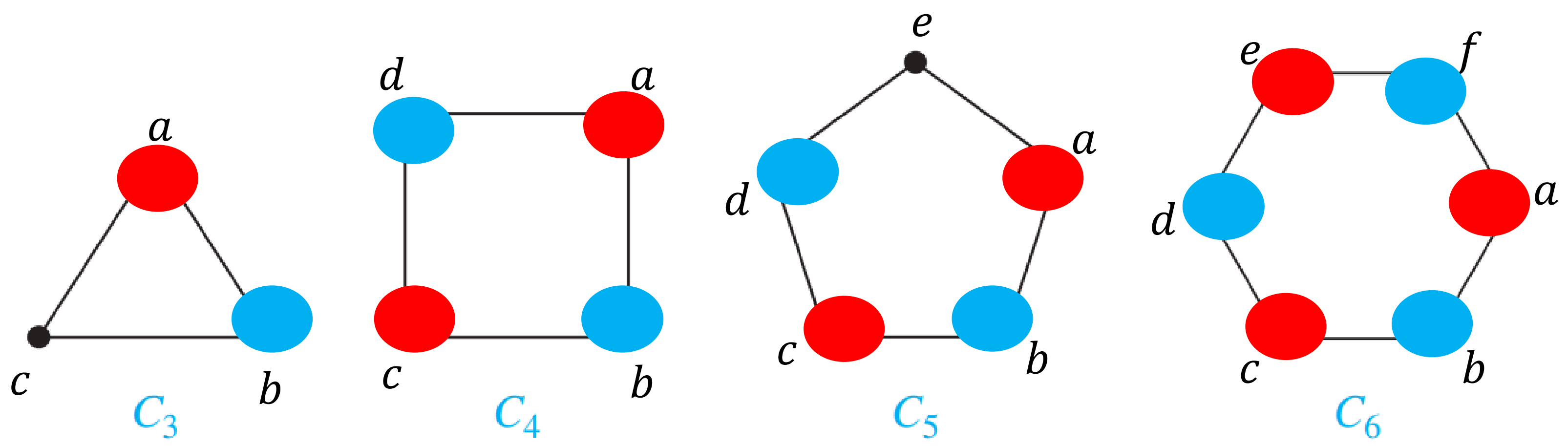
Determine whether the graph is bipartite or not?



Determining whether it is possible to assign either **red** or **blue** to each vertex so that *no two adjacent vertices are assigned the same color*.

Example 1: Answer

Determine whether the graph is bipartite or not?



bipartite

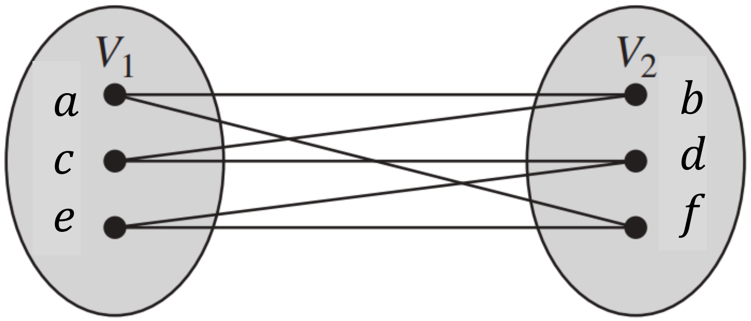
$V_1 = \{a, c\}$
$V_2 = \{b, d\}$
every edge of C_4 connects a vertex in V_1 and a vertex in V_2.

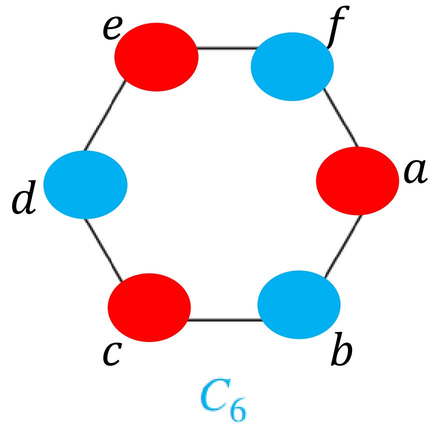
bipartite

$V_1 = \{a, c, e\}$
$V_2 = \{b, d, f\}$
every edge of C_6 connects a vertex in V_1 and a vertex in V_2.

Example 1: Answer

Determine whether the graph is bipartite or not?



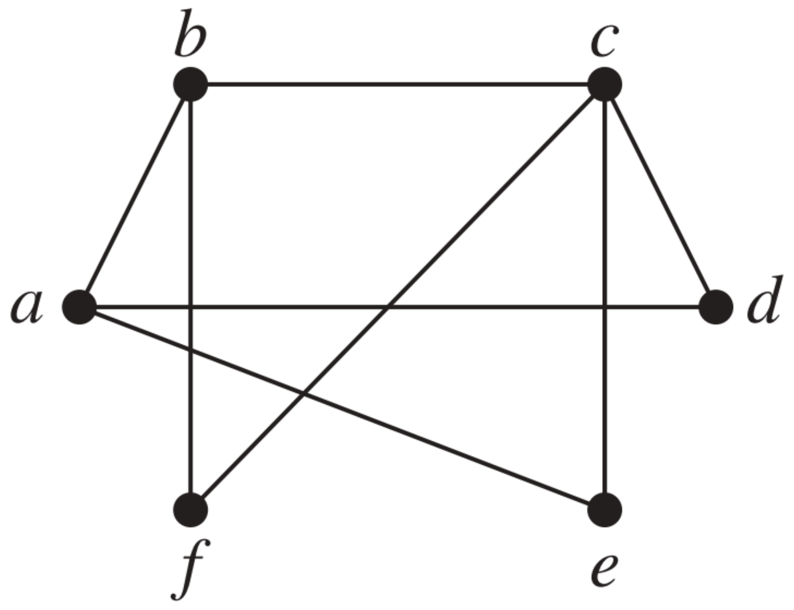


$bipartite$

$V_1 = \{a, c, e\}$
$V_2 = \{b, d, f\}$
every edge of C_6 connects a vertex in V_1 and a vertex in V_2.

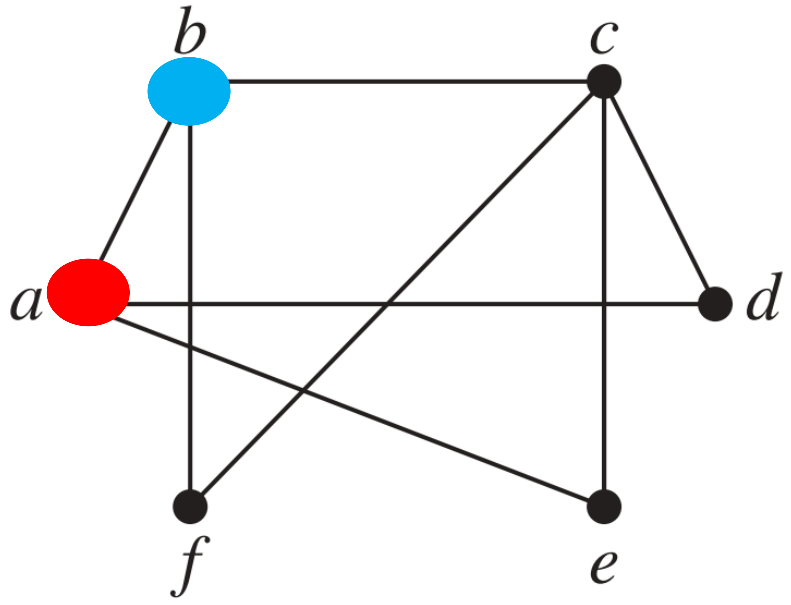
Example 2:

Determine whether the graph is bipartite or not?



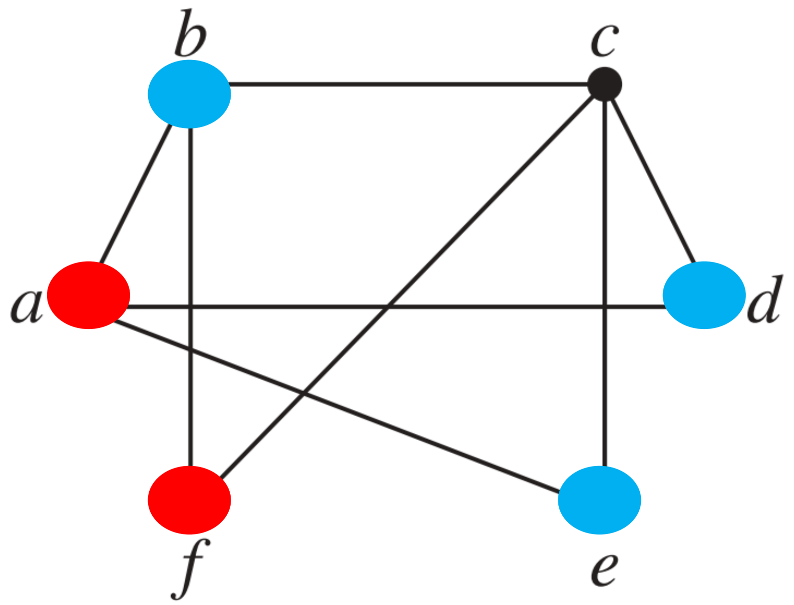
Example 2: Answer

Determine whether the graph is bipartite or not?



Example 2: Answer

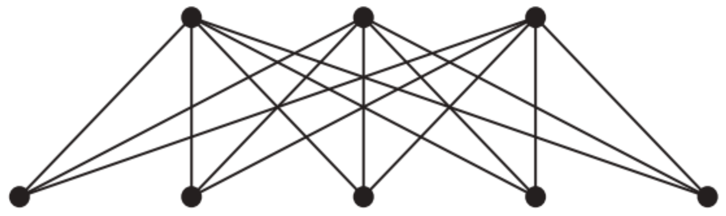
Determine whether the graph is bipartite or not?

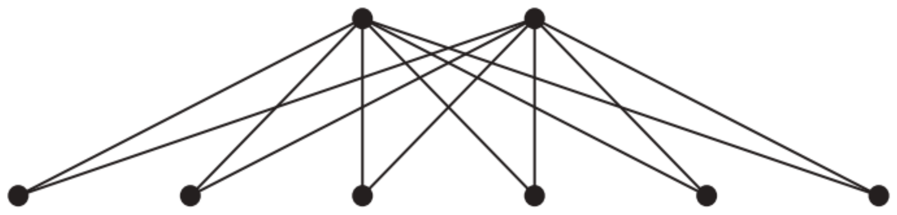


Not bipartite

Complete Bipartite Graphs:

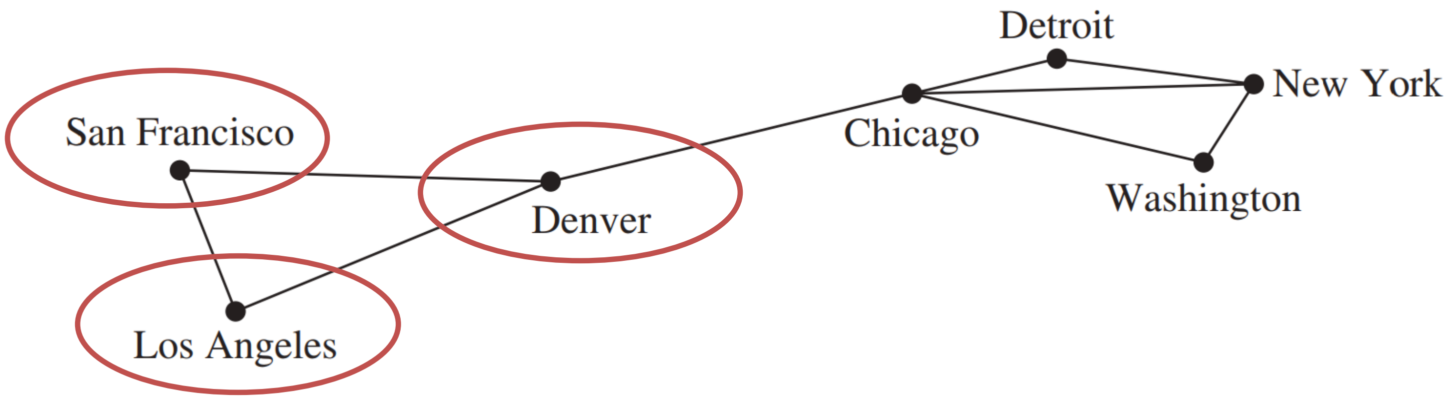
A complete bipartite graph $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets of m and n vertices, respectively with an edge between two vertices if and only if one vertex is in the first subset and the other vertex is in the second subset.


$K_{3,5}$


$K_{2,6}$

New Graphs from Old:

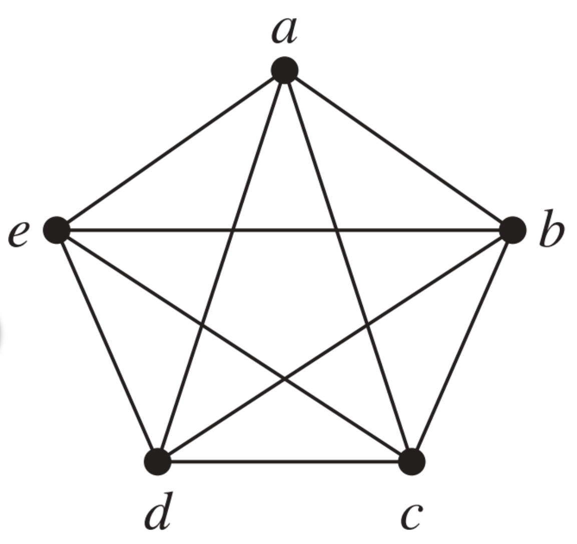
Sometimes we need only part of a graph to solve a problem. For instance, we may care only about the part of a large computer network that involves the computer centers in some cities. Then we can ignore the other computer centers and all telephone lines not linking two of these specific computer centers.

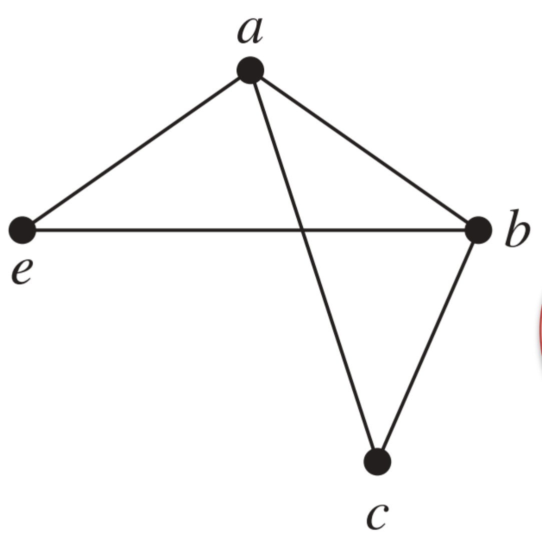


Definition 7:

A ***subgraph*** of a graph $G = (V, E)$ is a graph $H = (W, F)$, where $W \subseteq V$ and $F \subseteq E$. A subgraph H of G is a ***proper subgraph*** of G if $H \neq G$.

Original Graph K_5





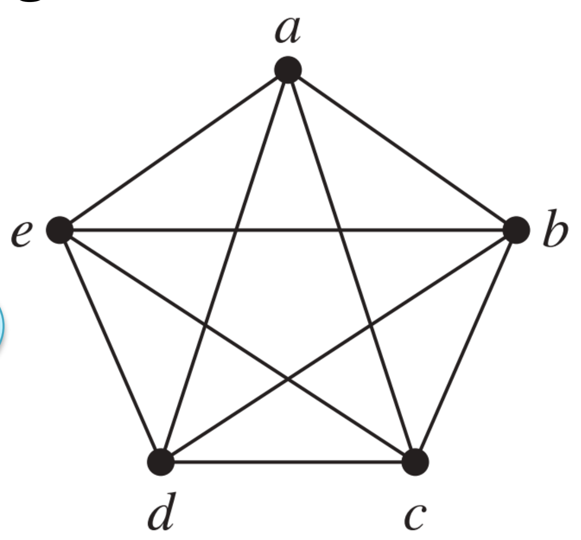
Proper Subgraph of The Original Graph

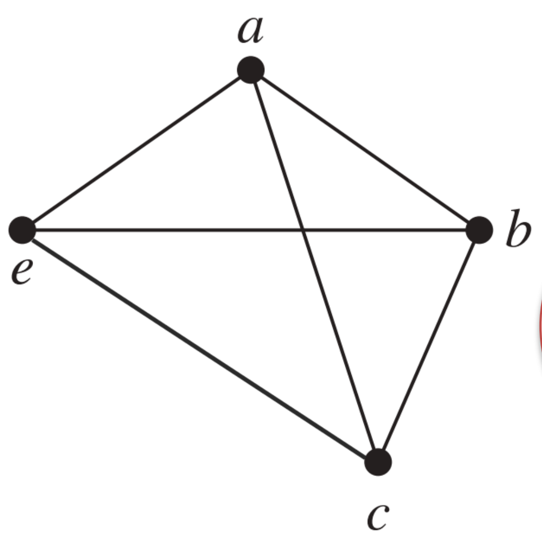
We can remove the vertices and edges from original graph

Definition 8:

Let $G = (V, E)$ be a simple graph. The ***subgraph induced*** by a subset W of the vertex set V is the graph (W, F), where the edge set F contains an edge in E if and only if both endpoints of this edge are in W

Original Graph K_5



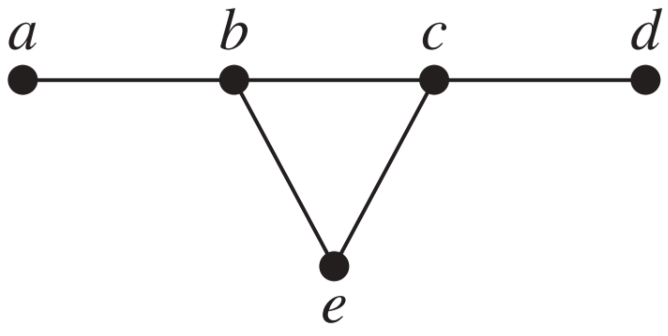


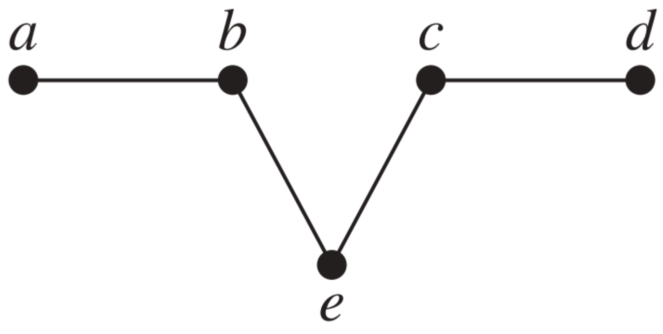
Subgraph Induced by $W = \{a, b, c, e\}$

Remove the vertices and its edges from original graph

Removing or Adding Edges of A Graph:

Original Graph (G)

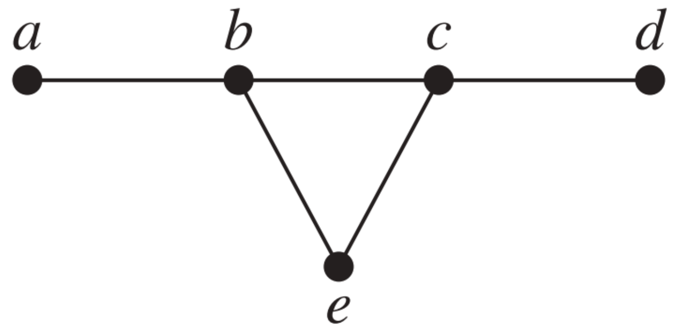


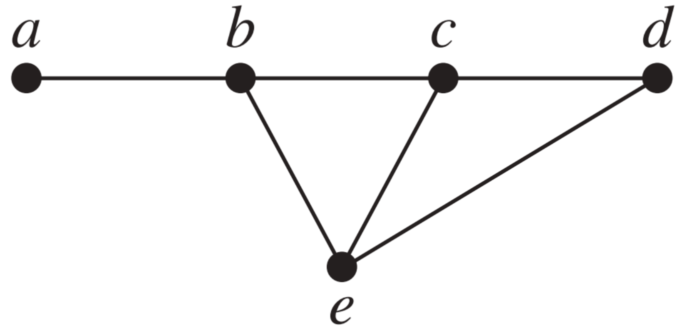


$G - \{b, c\}$: Remove the edge $\{b, c\}$

Removing or Adding Edges of A Graph:

Original Graph (G)

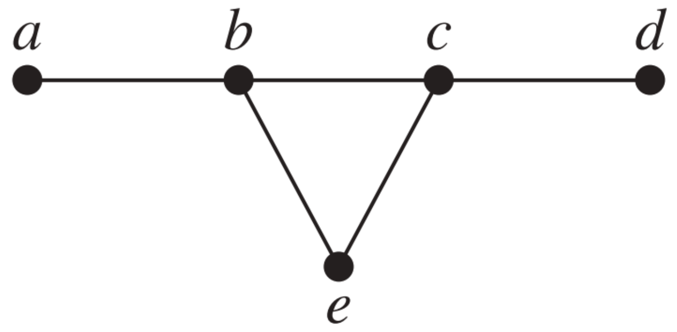


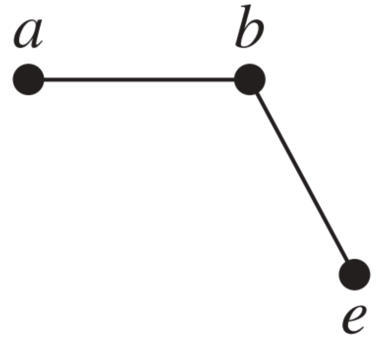


$G + \{e, d\}$: Add the edge $\{e, d\}$

Removing Vertices from A Graph:

Original Graph (G)

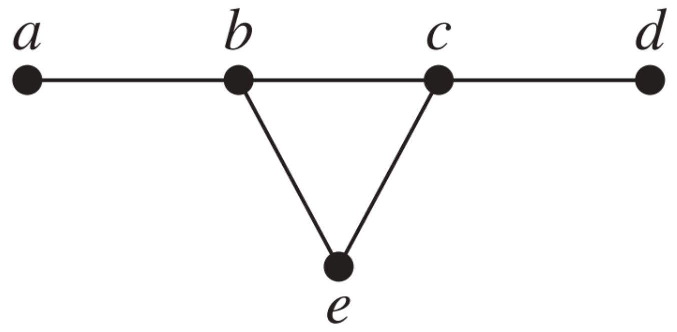


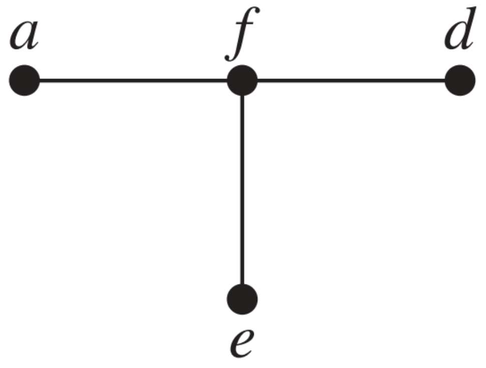


$G - c$: Remove the vertex c

Edge Contractions:

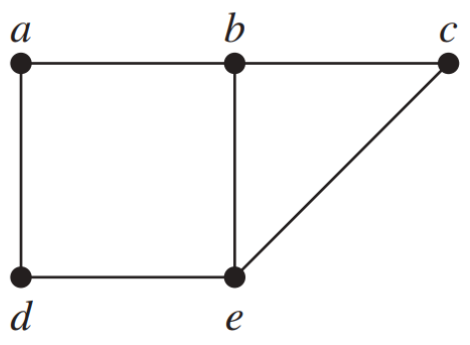
Original Graph (G)



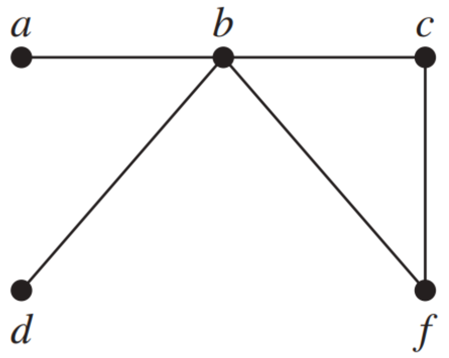


G contracted by replacing $\{b, c\}$ by f

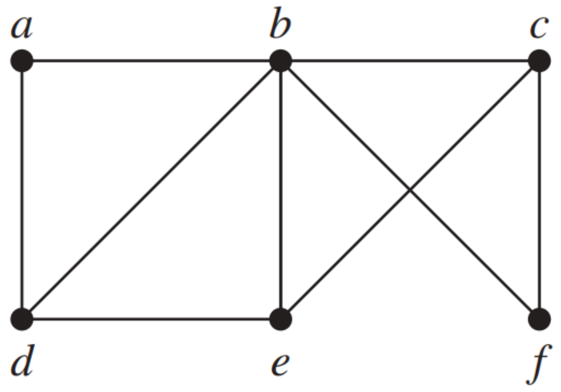
Graph Unions:



G_1

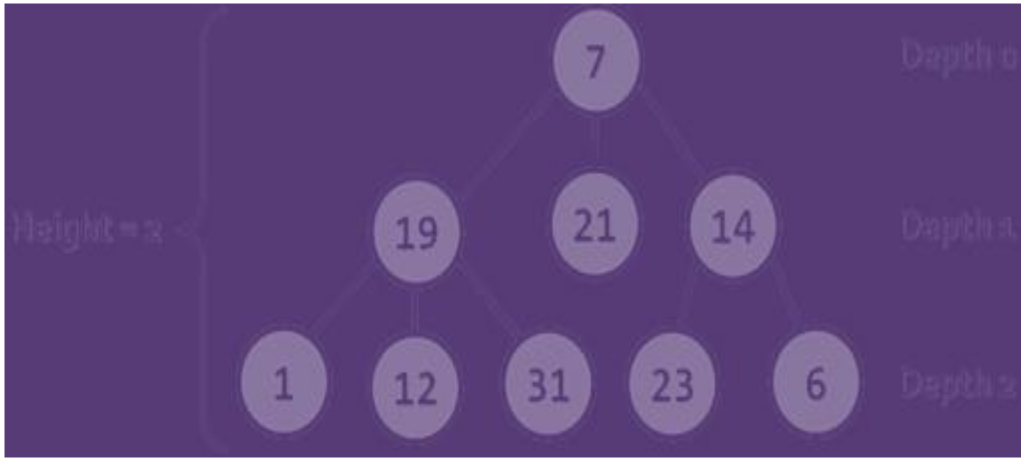


G_2



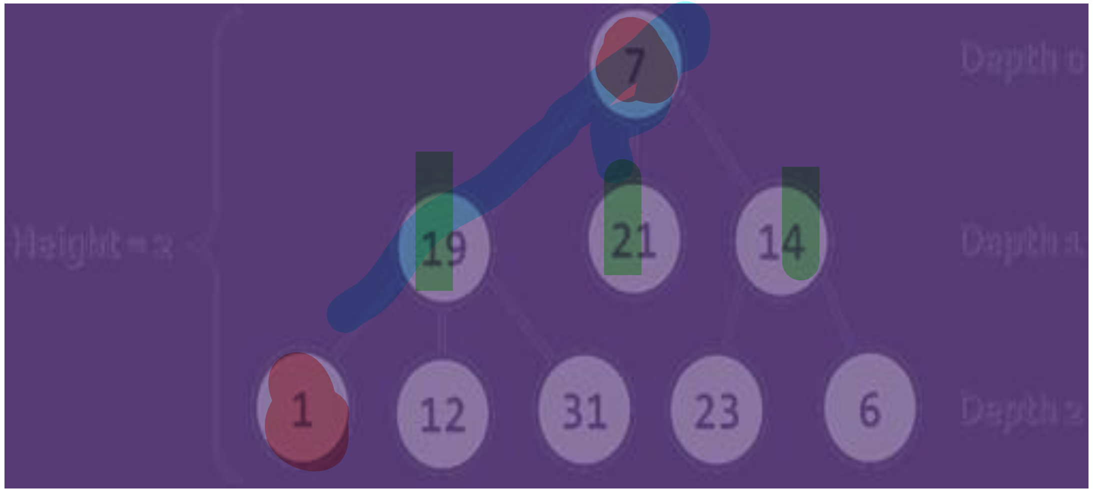
$G_1 \cup G_2$

Trees

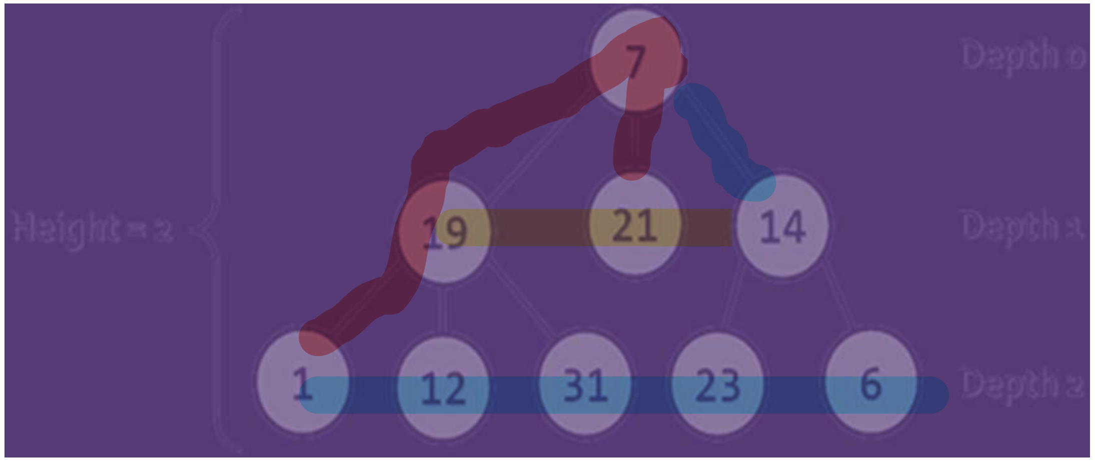


• Tree

- We will call every circle a **node** and each line an **edge**.
- Nodes "19", "21", "14" are below node "7" and are directly connected to it. This nodes we are called **direct descendants (child nodes)** of node "7", and node "7" their **parent**.
- The same way "1", "12" and "31" are children of "19" and "19" is their parent. Intuitively we can say that "21" is **sibling** of "19", because they are both children of "7“
- .For "1", "12", "31", "23" and "6" node "7" precedes them in the hierarchy, so he is their indirect parent – **ancestor**, ant they are called his **descendants**.



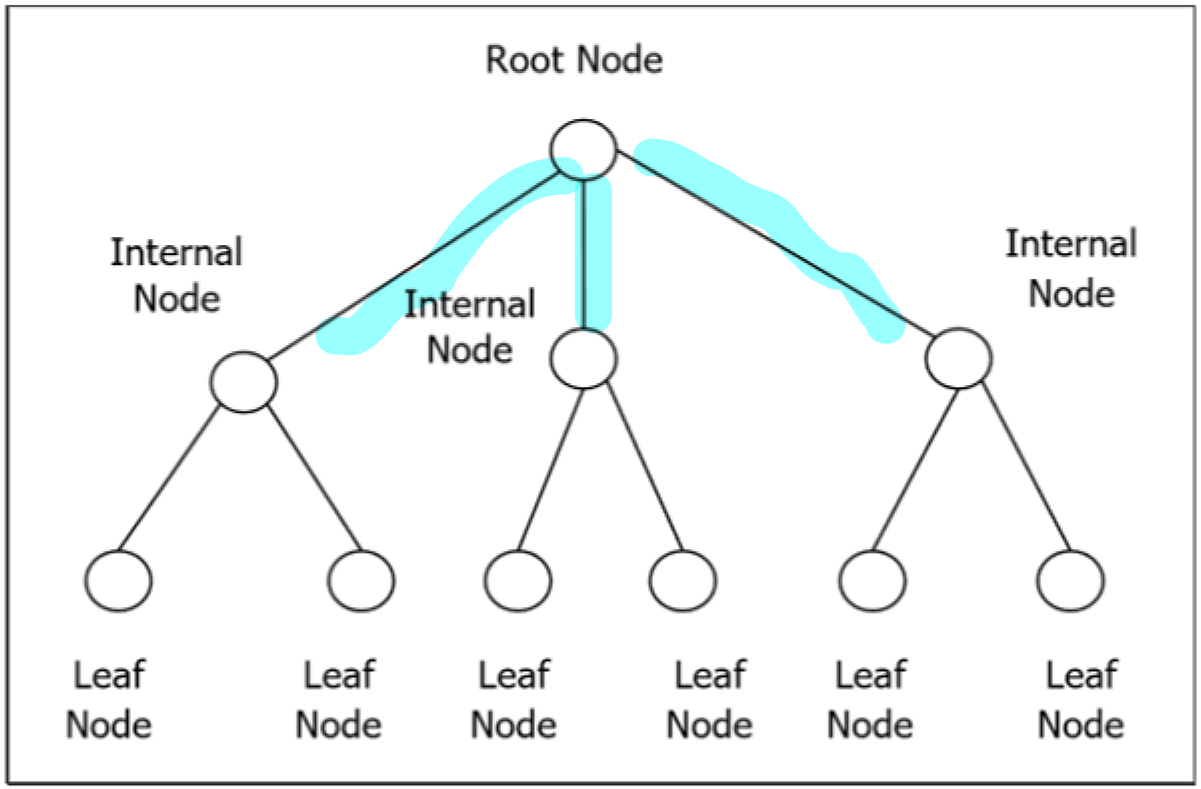
- **Root** is called the **node without parent**. In our example this is node "7“
- **Leaf** is a **node without child nodes**. In our example – "1", "12", "31", "21", "23" and "6".
- **Internal nodes** are the nodes, which are **not leaf or root** (all nodes, which have parent and at least one child). Such nodes are "19" and "14".
- **Path** is called a **sequence of nodes connected with edges**, in which there is no repetition of nodes. Example of path is the sequence "1", "19", "7" and "21". The sequence "1", "19" and "23" is not a path, because "19" and "23" are not connected.



- **Path length** is the number of edges, connecting the sequence of nodes in the path. Actually it is equal to the **number of nodes in the path minus 1**. The length of our example for path ("1", "19", "7" and "21") is three.
- **Depth** of a node we will call the **length of the path from the root to certain node**. In our example "7" as root has depth zero, "19" has depth one and "23" – depth two.
- We can give more simple definition of tree: **a node is a tree and this node can have zero or more children, which are also trees**.
- **Height of tree** – is the **maximum depth** of all its nodes. In our example the tree height is 2.
- **Degree** of node we call the **number of direct children** of the given node. The degree of "19" and "7" is three, but the degree of "14" is two. The leaves have degree zero.

Rooted Tree

- A rooted tree *G* is a connected acyclic graph with a special node that is called the root of the tree and every edge directly or indirectly originates from the root.
- An ordered rooted tree is a rooted tree where the children of each internal vertex are ordered.
- If every internal vertex of a rooted tree has not more than m children, it is called an m-ary tree.
- If every internal vertex of a rooted tree has exactly m children, it is called a full m-ary tree.
- If m=2, the rooted tree is called a binary tree.

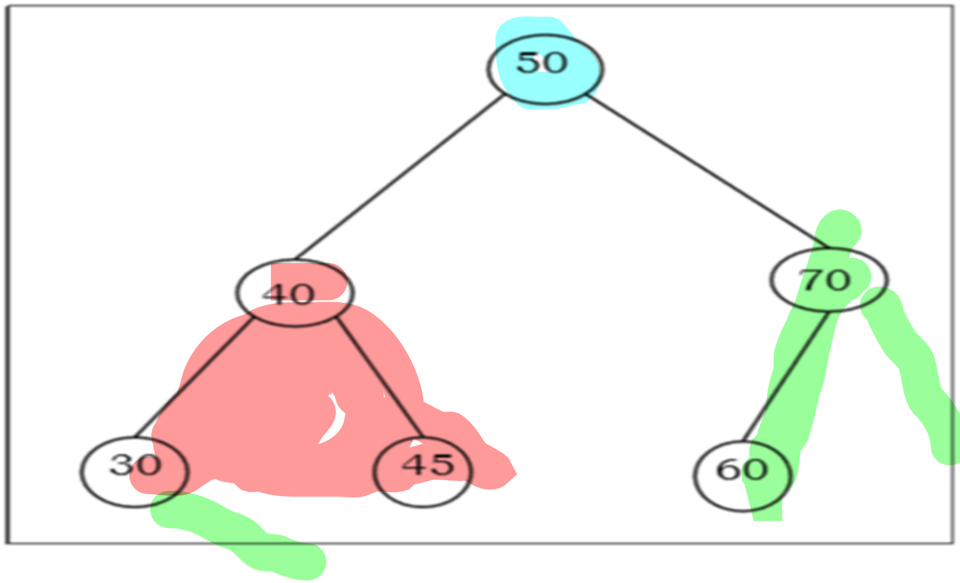

Root Node
Internal Node
Internal Node
Internal Node
Leaf Node
Leaf Node
Leaf Node
Leaf Node
Leaf Node
Leaf Node

Binary Search Tree

- Binary Search tree is a binary tree which satisfies the following property –
- *X* in left sub-tree of vertex *V*, *Value*(*X*) ≤ *Value*(*V*)
- *Y* in right sub-tree of vertex *V*, *Value*(*Y*) ≥ *Value*(*V*)
- So, the value of all the vertices of the left sub-tree of an internal node *V* are less than or equal to *V*

and the value of all the vertices of the right sub-tree of the internal node *V* are greater than or equal to *V*.

The number of links from the root node to the deepest node is the height of the Binary Search Tree.

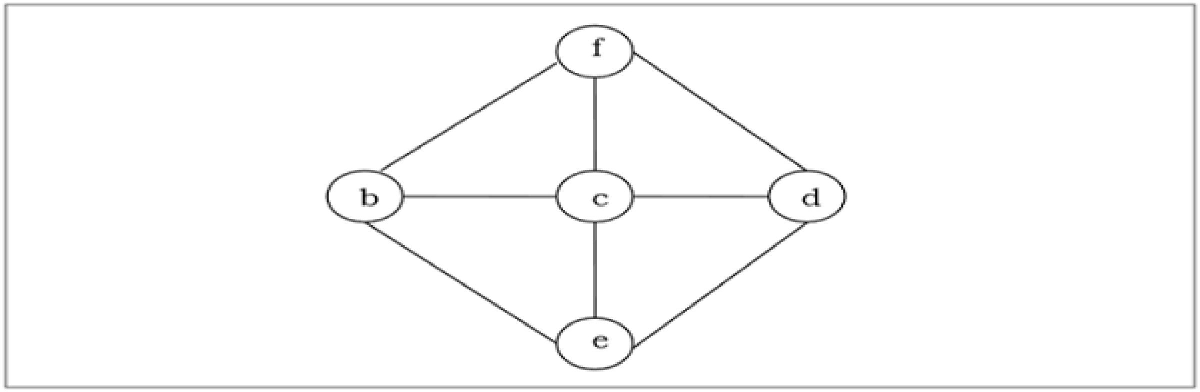


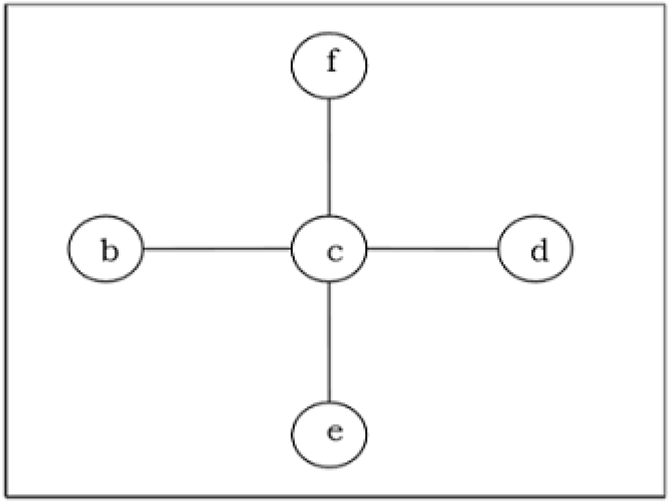
A spanning tree

of a connected undirected graph *G* is a tree that minimally includes all of the vertices of *G*. A graph may have many spanning trees.

Example

Spanning Trees

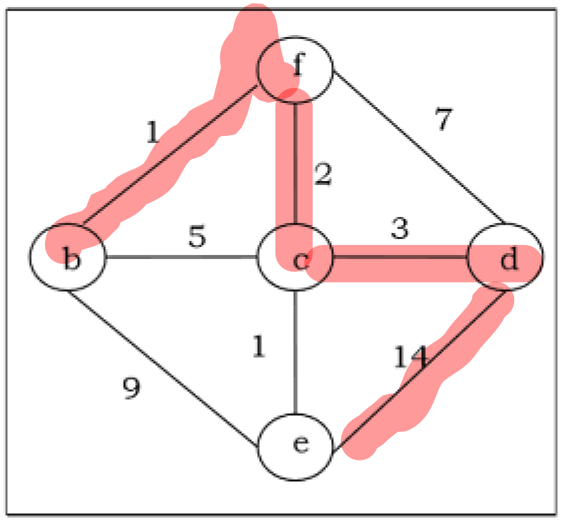


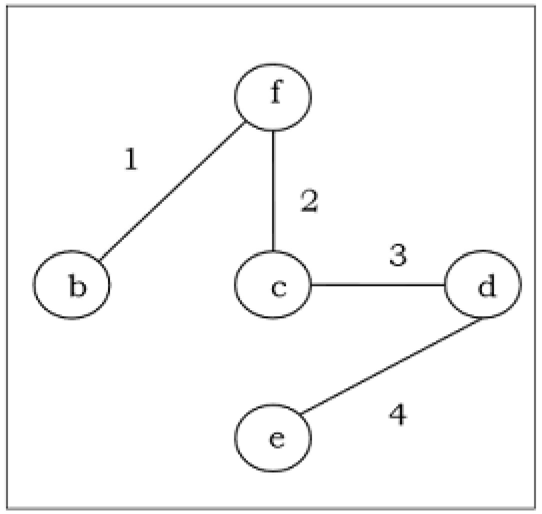


Minimum Spanning Tree

A spanning tree with assigned weight less than or equal to the weight of every possible spanning tree of a weighted, connected and undirected graph G , it is called minimum spanning tree (MST). The weight of a spanning tree is the sum of all the weights assigned to each edge of the spanning tree.

Example





Kruskal's Algorithm

- Kruskal's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weighted graph. It finds a tree of that graph which includes every vertex and the total weight of all the edges in the tree is less than or equal to every possible spanning tree.
- **Algorithm**
- **Step 1** − Arrange all the edges of the given graph $G(V,E)$
- in non-decreasing order as per their edge weight.
- **Step 2** − Choose the smallest weighted edge from the graph and check if it forms a cycle with the spanning tree formed so far.
- **Step 3** − If there is no cycle, include this edge to the spanning tree else discard it.
- **Step 4** − Repeat Step 2 and Step 3 until $(V-1)$
- number of edges are left in the spanning tree

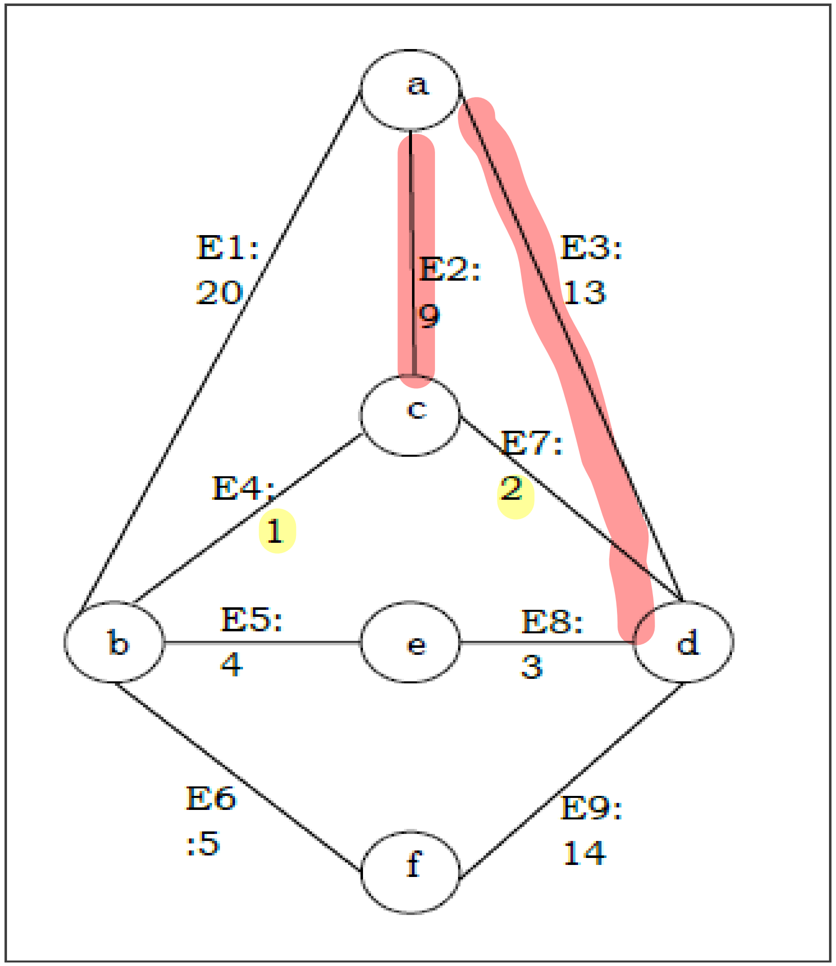
Problem

Suppose we want to find minimum spanning tree for the following graph G using Kruskal’s algorithm

Solution

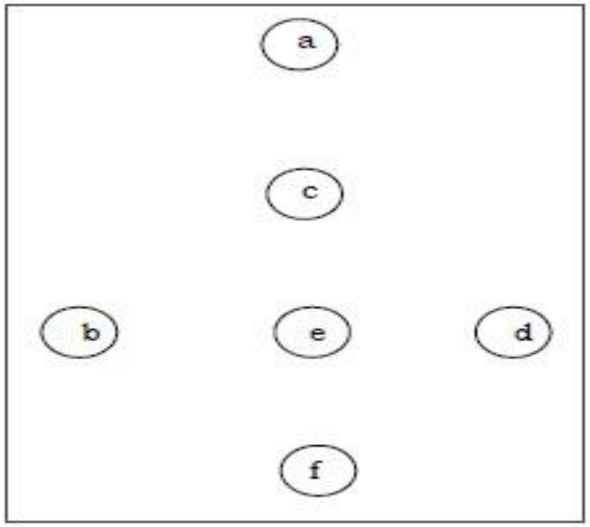
From the above graph we construct the following table −

Edge No.	Vertex Pair	Edge Weight
E1	(a, b)	٢٠
E2	(a, c)	٩
E3	(a, d)	١٣
E4	(b, c)	١
E5	(b, e)	٤
E6	(b, f)	٥
E7	(c, d)	٢
E8	(d, e)	٣
E9	(d, f)	١٤

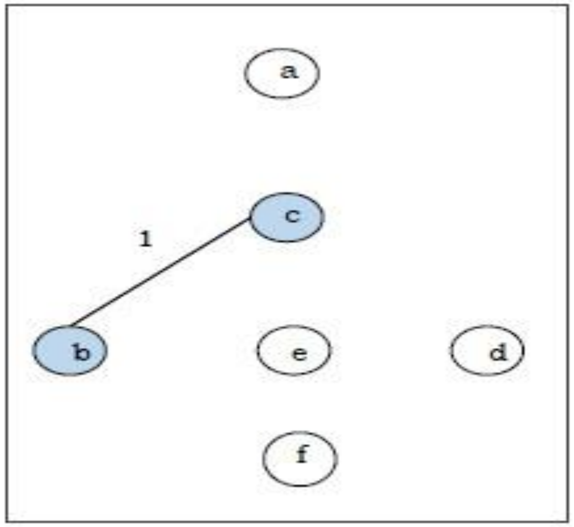


Now we will rearrange the table in ascending order with respect to Edge weight –

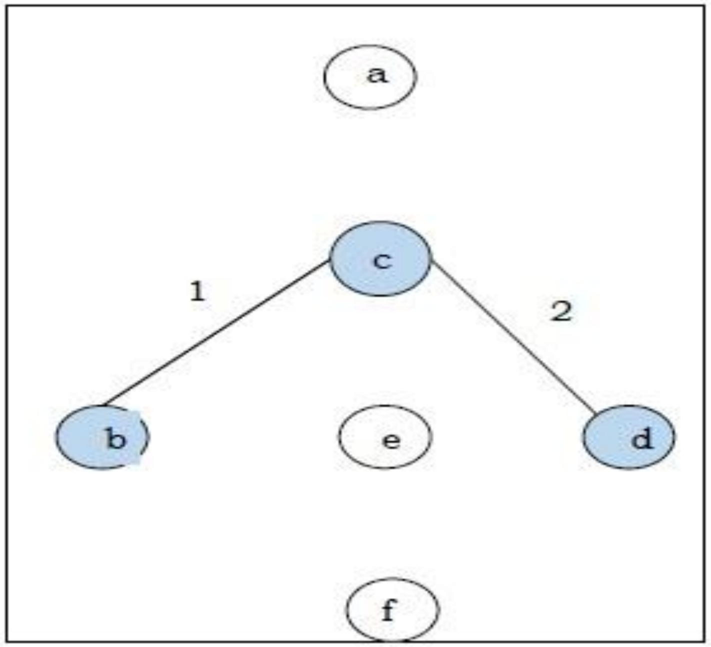
Edge No.	Vertex Pair	Edge Weight
E4	(b, c)	١
E7	(c, d)	٢
E8	(d, e)	٣
E5	(b, e)	٤
E6	(b, f)	٥
E2	(a, c)	٩
E3	(a, d)	١٣
E9	(d, f)	١٤
E1	(a, b)	٢٠



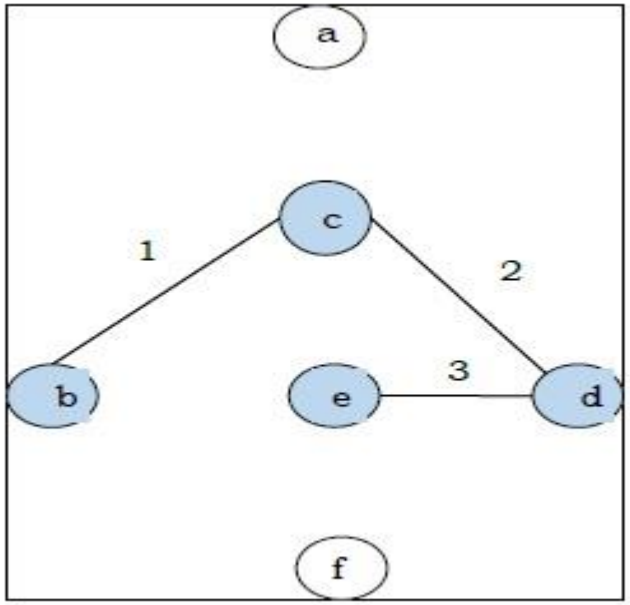
After adding vertices



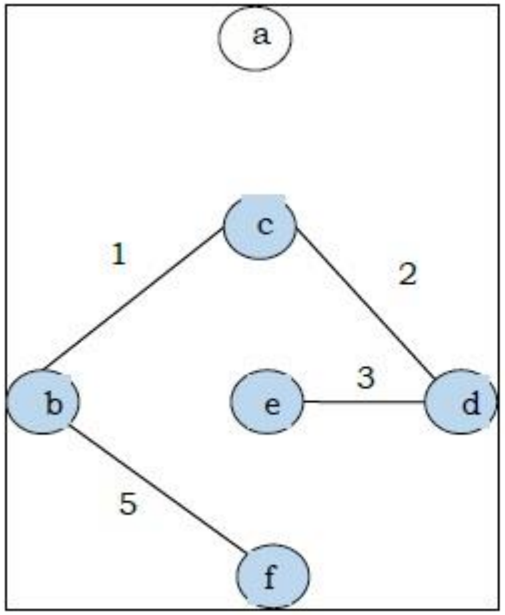
After adding edge E4



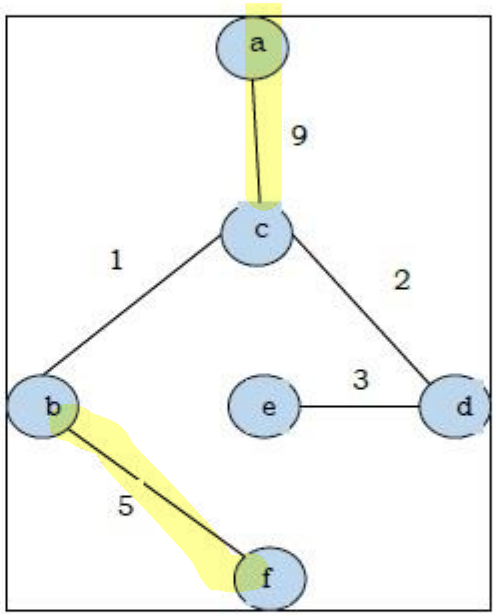
After adding edge E7



After adding edge E8



After adding edge E6
(don't add E5 since it forms cycle)



After adding edge E2

Since we got all the 5 edges in the last figure, we stop the algorithm and this is the minimal spanning tree and its total weight is (1+2+3+5+9)=20.