KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act 1956)

Coimbatore – 641 021.

LECTURE PLAN
DEPARTMENT OF MATHEMATICS

STAFF NAME: R.GAYATHRI
SUBJECT NAME: DISCRETE STRUCTURES SUB.CODE:19CSU202
SEMESTER: II CLASS: I B.Sc(CS) -'B'

S.No	Lecture Duration Period	Topics to be Covered	Support Material/Page Nos
UNIT-I			
1	1	Introduction, Sets , finite and infinite sets	S2:chapter-2,Pg.No:104-114
2	1	Uncountably infinite sets	S2:chapter-2,Pg.No:104-114
3	1	Functions	S2:chapter-2,Pg.No:148-150
4	1	Relations	S2:chapter-2,Pg.No:151-153
5		properties of binary relations	S2:chapter-2,Pg.No:154-155
6	1	closure, partial ordering relations, counting.	S2:chapter-2,Pg.No:183-186
7	1	continuation on closure,partial ordering relations, counting.	S2:chapter-2,Pg.No:186-191
8	1	Recapitulation and Discussion of possible questions	
	Total No of Hours Planned For Unit 1=8		
UNIT-II			
1	1	Pigeonhole principle	S2:chapter-2,Pg.No:192-197
2	1	Permutation and Combination	S1: chapter -4 Pg.No:313-315
3	1	continuation on Permutation and Combination	S1: chapter -4 Pg.No:316-318
4	1	Mathematical Induction	S1: chapter -4 Pg.No:320-323

5	1	continuation on Mathematical Induction	S1: chapter -4 Pg.No:323-326
6	1	Principle of inclusion and Exclusion.	S6: chapter -5 Pg.No:-172-176
7	1	continuation on Principle of inclusion and Exclusion.	S6: chapter -5 Pg.No:-176-181
8	1	Recapitulation and Discussion of possible questions	
	Total No of Hours Planned For Unit II=8		
		UNIT-III	
1	1	Recurrence relations-Definition and basic concepts	S1: chapter -7 Pg.No:449-452
2	1	continuation on basic concepts of Recurrence Relation	S1: chapter -7 Pg.No:452-455
3	1	Problems on Generating functions	S1: chapter -7 Pg.No:484 - 487
4	1	continuation on Problems on Generating functions	S1: chapter -7 Pg.No:487 - 490
5	1	Linear recurrence relation with constant coefficient	S1: chapter -7 Pg.No:460-465
6	1	continuation on Linear recurrence relation with constant coefficient	S1: chapter -7 Pg.No:465-470
7	1	Recapitulation and Discussion of possible questions	
	Total No of Hours Planned For Unit III= 7		
		UNIT-IV	
1	1	Introduction to Graph theory Basic terminology, models and types, multigraphs and weighted graphs	S1: chapter -9 Pg.No:589-595
2	1	Graph Representation and isomorphism of graphs	S1: chapter -8 Pg.No:611-620
3	1	Connectivity- Definition and theorems	S1: chapter -8 Pg.No:621-630
4	1	Euler’s and Hamiltonian paths	S1: chapter -8 Pg.No:633-645
5	1	Planner graph-theorem	S1: chapter -8 Pg.No:657-665
6	1	Graph coloring-Definition and theorems	S1: chapter -8 Pg.No:666-674

7	1	Tree and its Properties, Spanning trees	S1: chapter -9 Pg.No:724-735
8	1	Recapitulation and Discussion of possible questions	
	Total No of Hours Planned For Unit IV=8		
		UNIT-V	
1	1	Introduction to Statement and Notation Logical Connectives	S6: chapter -1 Pg.No:2-6 S4: chapter-1 Pg. No: 11 -12
2	1	Well formed formulae	S5: chapter -7 Pg.No:356-358
3	1	Tautologies-Problems	S2: chapter -1 Pg.No:24-25
4	1	Equivalence of formulae-Problems	S5: chapter -7 Pg.No:368-373
5	1	Theory of Inference	S2: chapter -1 Pg.No:65-67
6	1	Recapitulation and Discussion of possible questions	
7	1	Discuss on Previous ESE Question Papers	
8	1	Discuss on Previous ESE Question Papers	
9	1	Discuss on Previous ESE Question Papers	
	Total No of Hours Planned for unit V=9		
Total Planned Hours	**40**		

SUGGESTED READINGS

1. Kenneth Rosen. (2012). Discrete Mathematics and Its Applications (6th ed.). New Delhi: McGraw Hill.

2. Tremblay , J .P. , & Manohar, R. (1997). Discrete Mathematical Structures with Applications to Computer Science. New Delhi: McGraw-Hill Book Company.

3. Coremen, T.H., Leiserson, C.E. , & R. L. Rivest. (2009). Introduction to algorithms, (3rd ed.). New Delhi: Prentice Hall on India.

4. Albertson, M. O.,& Hutchinson, J. P. (1988). Discrete Mathematics with Algorithms . New Delhi: John wiley Publication.

5. Hein, J. L. (2009). Discrete Structures, Logic, and Computability(3rd ed.). New Delhi: Jones and Bartlett Publishers.

6. Hunter, D.J. (2017). Essentials of Discrete Mathematics. New Delhi: Jones and Bartlett Publishers.

KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University Established Under Section 3 of UGC Act1956)

Coimbatore – 641 021.

DEPARTMENT OF MATHEMATICS

SEMESTER-II

19CSU202	**DISCRETE STRUCTURE**	**Semester – II** **4H – 4C**

Instruction Hours / week: L: 4 T: 0 P: 0 **Marks:** Int : **40** Ext : **60** Total: **100**

COURSE OBJECTIVES

- To provides a deep knowledge to the learners to develop and analyze algorithms as well as enable them to think about and solve problems in new ways.
- To express ideas using mathematical notation and solve problems using the tools of mathematical analysis.

COURSE OUTCOME

On successful completion of the course,students will be able to

1.Familiar with elementary algebraic set theory

2.Acquire a fundamental understanding of the core concepts in growth of functions.

3.Describe the method of recurrence relations

4.get wide knowledge about graphs and trees

5.initiate to knowledge from inference theory

UNIT I

Sets: Introduction, Sets , finite and infinite sets, uncountably infinite sets, functions, relations, properties of binary relations, closure, partial ordering relations.

UNIT II

Pigeonhole principle, Permutation and Combination, Mathematical Induction, Principle of inclusion and Exclusion.

UNIT III

Recurrences: Recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution, Substitution Method, recurrence trees, Master theorem.

UNIT IV

Graph Theory : Basic terminology, models and types, multigraphs and weighted graphs, graph representation, graph isomorphism, connectivity, Euler and Hamiltonian Paths and circuits, Planar graphs, graph coloring, trees, basic terminology and properties of trees, introduction to Spanning trees

UNIT V

Prepositional Logic: Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory.

SUGGESTED READINGS

1. Kenneth Rosen. (2006). Discrete Mathematics and Its Applications (6th ed.). New Delhi: McGraw Hill.
2. Tremblay , J .P. , & Manohar, R. (1997). Discrete Mathematical Structures with Applications to Computer Science. New Delhi: McGraw-Hill Book Company.
3. Coremen, T.H., Leiserson, C.E. , & R. L. Rivest. (2009). Introduction to algorithms, (3rd ed.). New Delhi: Prentice Hall on India.
4. Albertson, M. O.,& Hutchinson, J. P. (1988). Discrete Mathematics with Algorithms . New Delhi: John wiley Publication.
5. Hein, J. L. (2009). Discrete Structures, Logic, and Computability(3rd ed.). New Delhi: Jones and Bartlett Publishers.
6. Hunter, D.J. (2008). Essentials of Discrete Mathematics. New Delhi: Jones and Bartlett Publishers.

UNIT – I

Sets: Introduction, Sets , finite and infinite sets, uncountably infinite sets, functions, relations, properties of binary relations, closure, partial ordering relations.

1. Introduction

Set: any collection of objects (individuals)

Naming sets: A, B, C,....

Members of a set: the objects in the set

Naming objects: a, b, c, ...

Notation: Let A be a set of 3 letters a, b, c.
We write A = {a, b, c}
a is a member of **A**, **a** is in **A**, we write **a** $\in$ **A**
d is not a member of **A**, we write **d** $\notin$ **A**

Important:
1. $\{a\} \neq a$
   {a} - a set consisting of one element a.
   a - the element itself
2. A set can be a member of another set:
   B = { 1, 2, {1}, {2}, {1,2}}

Finite sets: finite number of elements
Infinite sets: infinite number of elements
Cardinality of a finite set **A**: the number of elements in **A**: #A, or |A|

Describing sets:

a. by enumerating the elements of A:
   for finite sets: {red, blue, yellow}, {1,2,3,4,5,6,7,8,9,0}
   for infinite sets we write: {1,2,3,4,5,....}

b. by property, using predicate logic notation
   Let P(x) is a property, D - universe of discourse

The set of all objects in D, for which P(x) is true, is :

$$A = \{x \mid P(x)\}$$

we read: A consists of all objects x in D such that P(x) is true

c. by recursive definition, e.g. sequences

Examples:

1. The set of the days of the week:

{Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}

2. The set of all even numbers :

$\{ x \mid even(x) \}$
$\{ 2,4,6,8,\ldots\}$

3. The set of all even numbers, greater than 100:

$\{ x \mid even(x) \wedge x > 100\}$
$\{ 102, 104, 106, 108,\ldots\}$

4. The set of integers defined as follows:
$a_1 = 1$, $a_{n+1} = a_n + 2$ (the odd natural numbers)

Universal set: **U** - the set of all objects under consideration

Empty set: Ø set without elements.

2. Relations between sets

2.1. Equality

Let **A** and **B** be two sets.
We say that **A** is equal to **B**, **A = B** if and only if they **have the same members**.

Example:

$A = \{2,4,6\}$, $B = \{2,4,6\}$ $A = B$

$A = \{a, b, c\}$, $B = \{c, a, b\}$ $A = B$

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

Written in predicate notation:

$$A = B \text{ if and only if } \forall x, x \in A \leftrightarrow x \in B$$

2.2. Subsets

The set of all numbers contains the set of all positive numbers. We say that the set of all positive numbers is a **subset** of the set of all numbers.

Definition: A is a subset of **B** if all elements of **A** are in **B.** However B may contain elements that are not in **A**

Notation: **A** $\subseteq$ B
Formal definition:

$A \subseteq B$ if and only if $\forall x, x \in A \rightarrow x \in B$

Example: $A = \{2,4,6\}$, $B = \{1,2,3,4,5,6\}$, $A \subseteq B$

Definition: if **A** is a subset of **B, B** is called **a superset of A.**

Other definitions and properties:

a. If $A \subseteq B$ and $B \subseteq A$ then $A = B$

If **A** is a subset of **B**, and B is a subset of **A**, **A** and B are equal.

b. Proper subsets: **A** is a proper subset of **B**, $A \subset B$, if and only if **A** is a subset of B and there is at least one element in **B** that is not in **A**.

$A \subset B$ iff $\forall x, x \in A \rightarrow x \in B \wedge \exists x, x \in B \wedge x \notin A$

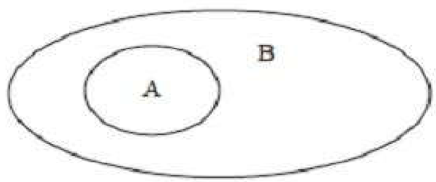
2.3. Disjoint sets

Definition: Two sets A and B are disjoint if and only if they have no common elements

A and B are disjoint if and only if $\sim\exists\, x,\ (x \in A) \wedge (x \in B)$
i.e. $\forall\, x,\quad x \notin A \vee x \notin B$

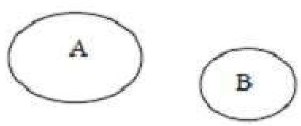
If two sets are not disjoint they have common elements.

Picturing sets: **Venn diagrams** - used to represent relations between sets

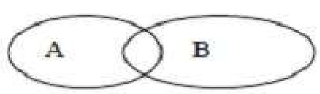


A is a (proper) subset of B

All elements in the set A are also elements in the set B



Disjoint sets



Not disjoint sets

3. Operations on sets

3.1. Intersections

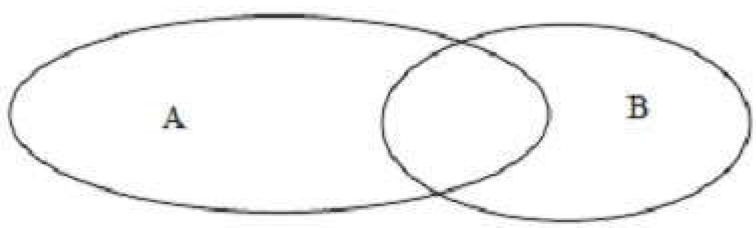
The set of all students at Simpson and the set of all majors in CS have some elements in common - the set of all students in Simpson that are majoring in CS. This set is formed as the intersection of all students in CS and all students at Simpson.

Definition: Let **A** and B are two sets. The set of all elements common to **A** and **B** is called the intersection of **A** and **B**

Notation: $\mathbf{A} \cap \mathbf{B}$
Formal definition:

$$A \cap B = \{x \mid (x \in A) \wedge (x \in B)\}$$

Venn diagram:



Example: $\mathbf{A = \{2,4,6\}, B = \{1,2,5,6\}, \quad A \cap B = \{2,6\}}$

Other properties:

$A \cap B \subseteq A, \quad A \cap B \subseteq B$

The intersection of two sets A and B is a subset of A, and a subset of B

$A \cap \emptyset = \emptyset$ The intersection of any set A with the empty set is the empty set
$A \cap U = A$ The intersection of any set A with the universal set is the set A itself.

Intersection corresponds to conjunction in logic.

Let $A = \{x \mid P(x)\}$, $B = \{x \mid Q(x)\}$

$A \cap B = \{x \mid P(x) \wedge Q(x)\}$

3. 2. Unions

The set of all rational numbers and the set of all irrational numbers taken together form the set of all real numbers - as a **union** of the rational and irrational numbers.

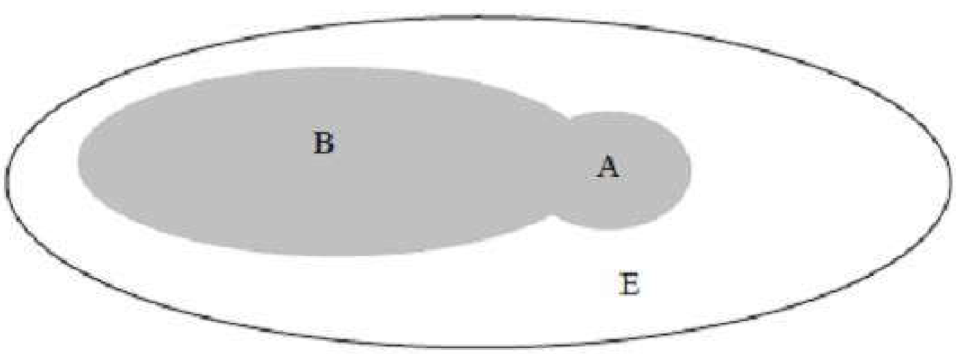
All classes at Simpson consist of students. If we take the elements of all classes, we will get all students - as the union of all classes.

Definition: The union of two sets A and B consists of all elements that are in A combined with all elements that are in B.
(note that an element may belong both to A and B)

Notation: $\mathbf{A \cup B}$
Formal definition:

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

Venn diagram:



Example: $A = \{2,4,6,8,10\}$, $B = \{1,2,3,4,5,6\}$, $A \cup B = \{1,2,3,4,5,6,8,10\}$
$A \cup B$ contains all elements in A and B without repetitions.

Other properties of unions:

$A \subseteq A \cup B \quad B \subseteq A \cup B$

A is a subset of the union of A and B.
B is a subset of the union of A and B

$A \cup \emptyset = A$ The union of any set A with the empty set is A
$A \cup U = U$ The union of any set A with the universal set E is the universal set.

Union corresponds to disjunction in logic.
Let $A = \{x \mid P(x)\}$, $B = \{x \mid Q(x)\}$
$A \cup B = \{x \mid P(x) \vee Q(x)\}$

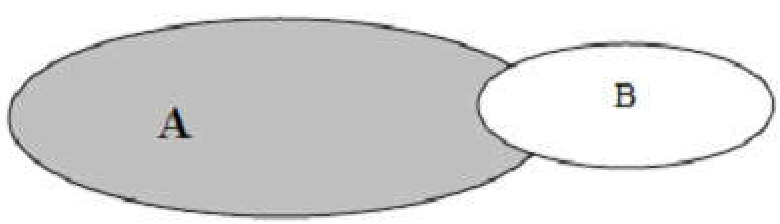
3.3. Differences

Definition: Let A and B be two sets. The set **A - B**, called the difference between A and B, is the set of all elements that are in A and are not in B.

Notation: **A – B or A \ B**
Formal definition:

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

Venn diagram:



Example: A = {2,4,6}, B = {1, 5,6}, A - B = {2,4}

A – Ø = A The difference between A and the empty set is A
A – U = Ø The difference between A and the universal set is the empty set.

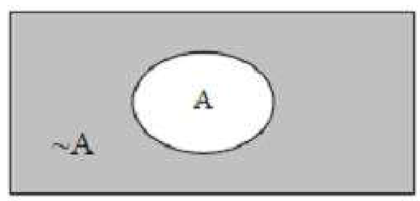
3.4. Complements

Definition: Let **A** be a set. The set of all objects within the universal set that are not in **A**, is called the complement of **A**.

Notation: ~**A**
Formal definition:

$$\sim A = \{x \mid x \notin A\}$$

Venn diagram:



SETS IDENTITIES

Using the operation unions, intersection and complement we can build expressions ov
sets.

Example:
A - set of all black objects
B - set of all cats
A ∩ B - set of all black cats

The set identities are used to manipulate set expressions

$A \cup \sim A = U$	Complementation Law
$A \cap \sim A = \emptyset$	Exclusion Law
$A \cap U = A$ $A \cup \emptyset = A$	Identity Laws
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Domination Laws
$A \cup A = A$ $A \cap A = A$	Idempotent Laws
$\sim(\sim A) = A$	Double Complementation Law
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Commutative Laws
$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$	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
$\sim(A \cap B) = \sim A \cup \sim B$ $\sim(A \cup B) = \sim A \cap \sim B$	De Morgan's Laws
$A - B = A \cap \sim B$	Alternate representation for set difference

Proof problems for sets

A. Element Proofs

Definitions used in the proofs

Def 1: $A \cup B = \{ x \mid x \in A \ \vee \ x \in B\}$

Def 2: $A \cap B = \{ x \mid x \in A \ \wedge \ x \in B\}$

Def 3: $A - B = \{ x \mid x \in A \ \wedge \ x \notin B\}$

Def 4: $\sim A = \{ x \mid x \notin A \}$

Inference rules often used:

$P \wedge Q \models P, Q$
$P, Q \models P \wedge Q$
$P \models P \vee Q$

How to prove that two sets are equal:
$A = B$
1) show that $A \subseteq B$, i.e. choose an arbitrary element in A and show that it is in B
2) show that $B \subseteq A$, i.e. choose an arbitrary element in B and show that it is in A

The element was chosen arbitrary, hence any element that is a member of the left se also a member of the right set, and vice versa.

Example:
Prove that $A - B = A \cap \sim B$

1. Show that $A - B \subseteq A \cap \sim B$

Let $x \in A - B$
By Def 3:
$x \in A \wedge x \notin B$ (1)
By (1) $x \in A$ (2)
By (1) $x \notin B$ (3)
By (3) and Def 4: $x \in \sim B$ (4)
By (2), (4)
$x \in A \wedge x \in \sim B$ (5)
By (5) and Def 2:
$x \in A \cap \sim B$

x was an arbitrary element in A – B, therefore $A - B \subseteq A \cap \sim B$ (6)

2. Show that $A \cap \sim B \subseteq A - B$

Let $x \in A \cap \sim B$

By Def 2:

$$x \in A \wedge x \in \sim B \quad (7)$$

By (7) $x \in A$ (8)

By (7) $x \in \sim B$ (9)

By (9) and Def 4: $x \notin B$ (10)

By (8), (10)

$$x \in A \wedge x \notin B \quad (11)$$

By (11) and Def 3:

$$x \in A - B$$

x was an arbitrary element in $A \cap \sim B$, therefore $A \cap \sim B \subseteq A - B$ (12)

by (6) and (12):

$$A - B \subseteq A \cap \sim B$$

Q.E.D.

B. Using set identities

Prove that $A \cap (\sim A \cup B) = A \cap B$

Method: Apply the set identities to the expression on the left, until the expression on th right is obtained.

By Distribution Laws: $A \cap (\sim A \cup B) = (A \cap \sim A) \cup (A \cap B)$

By the Exclusion Law $A \cap \sim A = \emptyset$

Hence $A \cap (\sim A \cup B) = \emptyset \cup (A \cap B)$

By the Identity Law: $\emptyset \cup (A \cap B) = A \cap B$

Hence $A \cap (\sim A \cup B) = A \cap B$

1. Set partitions

Two sets are disjoint if they have no elements in common, i.e. their intersection is the empty set.

A and B are disjoint sets iff $A \cap B = \varnothing$

Definition: Consider a set A, and sets $A_1, A_2, \ldots A_n$, such that:

a. $A_1 \cup A_2 \cup \ldots \cup A_n = A$

b. $A_1, A_2, \ldots A_n$, are mutually disjoint, i.e. for all i and j, $A_i \cap A_j = \varnothing$

The set $\{A_1, A_2, \ldots A_n\}$ is called a **partition** of A

Example:

1. A = {a, b, c, d, e, f, g}
$A_1 = \{a, c, d\}$
$A_2 = \{b, f\}$
$A_3 = \{e, g\}$

The set {{a, c, d}, {b, f}, {e, g}} is a partition of A.

2. Cartesian product

Consider the identification numbers on license plates: $x_1x_2x_3$ $Y_1Y_2Y_3$
where $x_1x_2x_3$ is a 3-digit number and $Y_1Y_2Y_3$ is a combination of 3 letters

How do we make sure that each license plate would have a different identification number?

The program that assigns numbers uses Cartesian product of sets.

Definition: Let A and B be two sets. The Cartesian product of A and B is defined as t set

$$A \times B = \{(x,y) \mid x \in A \ \wedge \ y \in B\}$$

Example 1:

A = {0, 1, 2, 3}
B = {a, b}

A x B = {(0,a), (0,b), (1,a), (1,b), (2,a), (2,b), (3,a), (3,b)}

Example 2:

A = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
A x A = {(0,0),(0,1), (0,2), ... (0,9),
(1,0),(1,1), (1,2), ...,(1,9),
......
(9,0),(9,1), (9,2), ... (9,9)}

We can consider the result to be the set of all 2-digit numbers.

3. Power sets

Definition: The set of all subsets of a given set A is called power set of A.

Notation 2^A , or $\mathcal{P}(A)$

Example:

A - { a,b,c,d}

$\mathcal{P}(A)$ = {∅, {a},{b},{c},{d}
{a,b},{a,c},{a,d},{b,c},{b,d},{c,d}
{a,b,c},{a,b,d},{a,c,d}, {b, c, d}
{a,b,c,d}}

Number of elements in $\mathcal{P}(A)$ is 2^N, where N = number of elements in A
Why 2^N?

<u>Bit notation</u>: For a set A with **n** elements, each subset of A can be represented by a string of length *n* over {0,1}, i.e. a string consisting of 0s and 1s.

For example:

{a,b} = 1 1 0 0
{a,c} = 1 0 1 0
{b,c,d} = 0 1 1 1

The i-th element in the string is 1 if the element a_i is in the subset, otherwise it is 0.
Thus the subset {a,b,d} of the set {a,b,c,d} can be represented by the string '1101'

There are 2^n different strings with length **n** over {0,1}(why?), hence the number of the subsets is 2^n .

Set Relations

2. Definition

Let A and B be two sets. A relation **R** from **A** to **B** is any set of pairs (x,y), x ϵ A, y ϵ B, i.e. any subset of A x B.

If **x** and **y** are in relation **R**, we write xRy, or $(x,y) \in R$..

R is a set defined as
$R = \{(x,y) \mid x \in A, y \in B, xRb.\}$

3. Relations and Cartesian products

Relations between two sets **A** and **B** are sets of pairs of elements of **A** and **B**.
The Cartesian product **A x B** consists of all pairs of elements of **A** and **B.**

Thus relations between two sets are subsets of the Cartesian product of the sets.

Example:

Let $A = \{1, 3, 4, 5\}$
$B = \{2, 7, 8\}$

The relation R1 :"less than" from set A to set B is defined by the following set:

$R1 = \{(1, 2), (1, 7), (1, 8), (3, 7), (3, 8), (4, 7), (4, 8), (5, 7), (5, 8)\}$

This set is a subset of the Cartesian product of A and B:

A x B = **{(1,2),(1,7),(1,8),**
(3,2), **(3,7), (3,8),**
(4,2), **(4,7), (4,8)**,
(5,2),**(5,7),(5,8)**}
(the members of R1 are in boldface)

The relation R2: "greater than" from set A to set B is defined by the set:

$R2 = \{(3, 2), (4, 2), (5, 2)\}$

It is also a subset of A x B.

The relation R3 "equal to" from A to B is the empty set, since no element in A is equal to an element in B.

7. Domains and ranges

Let R be a relation from X to Y,
the **domain of R** is the set of all elements in X that occur in at least one pair of the relation,
the **range of R** is the set of all elements in Y that occur in at least one pair of the relation.

In the above example, the domain of **R: choose(x,y)** is the set of students {Ann, Tom , Paul}, and the range is the set of food items: {spaghetti, fish, pie, cake}.

The domain and the range are easily found using the matrix or the graph representations of the relation.

1. Definition

Let A and B be two sets. A relation **R** from **A** to **B** is any set of pairs (x,y), x ϵ A, y ϵ B, i.e. any subset of A x B.
The empty set is a subset of the Cartesian product – the empty relation

2. How to write relations

a. as set of pairs

A = {1,2,3}, {B = 4,5,6}
R = {{1,4), (1,5), (1,6), (2,4), (2,6), (3,6)}

b. using predicates

A = {1,2,3}, {B = 4,5,6}
R = {(x,y) | x $\in$ A, y $\in$ B, y is a multiple of x}

3. Graph and matrix representation

A = {1,2,3}, {B = 4,5,6}
R = {{1,4), (1,5), (1,6), (2,4), (2,6), (3,6)}

1. Set operations and relations

Relations are sets. All set operations are applicable to relations

Examples:

Let A= {3, 5, 6, 7}
B ={4, 5, 9}

Consider two relations R and S from A to B:

$R = \{(x,y) | x \in A, y \in B, x < y\}$

If $(x,y) \in R$ we write xRy

R is a finite set and we can write down explicitly its elements:
R= {(3,4),(3,5),(3,9),(5,9), (6,9),(7,9)}

$S = \{(x,y) | x \in A, y \in B, |x-y| = 2\}$

If $(x,y) \in S$ we write xSy

S is a finite set and we can write down explicitly its elements:
S = {(3,5), (6,4), (7,5), (7,9)}

For R and S the universal set is A x B:

{(3,4), (3, 5), (3, 9),
(5, 4), (5, 5), (5, 9),
(6, 4), (6, 5), (6, 9),
(7, 4), (7, 5), (7, 9)}

a) intersection of R and S:

$R \cap S = \{(x,y) \mid xRy \wedge xSy\}$ $R \cap S = \{(3,5),(7,9)\}$

b) union of R and S:

$R \cup S = \{(x,y) \mid xRy \vee xSy\}$

$R \cup S = \{ (3,4),(3,5),(3,9),(5,9), (6,9),(7,9), (6,4), (7,5) \}$

c) complementation:

$\sim R = \{(x,y) \mid \sim(xRy)\}$

$\sim R = U - R$

The universal set for R is the Cartesian product A x B

$A = \{ 3,5,6,7\}$

$B = \{4,5,9\}$

$U = A \times B = \{(3,4), (3,5), (3,9), (5,4), (5,5), (5,9), (6,4), (6,5), (6,9), (7,4), (7,5), (7,9)\}$

$R = \{(3,4), (3,5), (3,9), (5,9), (6,9), (7,9)\}$

$U - R = \{(5,4), (5,5), (6,4), (7,4), (7,5)\}$

Note that for any two sets A and B, $A - B = A \cap \sim B$

d) difference R - S, S - R:

$R - S = \{(x,y) \mid xRy \wedge \sim(xSy)\}$

$R - S = \{(3,4),(3,9),(5,9), (6,9)\}$

2. Inverse relation

Let R: A→B be a relation from A to B. The inverse relation R^{-1} : B→A is defined as in the following way:

$$R^{-1} : B \to A \ \{(y,x) \mid (x,y) \in R\}$$

Thus $xRy \equiv yR^{-1}x$

Examples:

a. Let A = {1,2,3}, B = {1,4,9}

Let R: B→A be the set {(1,1), (1,4), (2,2), (2,4), (3,3)}
R^{-1} : B→A is the relation {(1,1), (4,1), (2,2), (4,2), (3,3)}

b. Let A = {1,2,3}, R: A→A be the relation {(1,2), (1,3), (2,3)}

R^{-1} is the relation: {(2,1), (3,1), (3,2)}

3. Composition of relations

Let X, Y and Z be three sets, R be a relation from X to Y, S be a relation from Y to Z.

A composition of R and S is a relation from X to Z :

$S \circ R = \{(x,z) \mid x \in X, z \in Z, \exists y \in Y$, such that xRy, and ySz$\}$

Note that the operation is right-associative, i.e. we first apply R and then S

Example 1:

Let X, Y, and Z be the sets:

X: {1,3,5}
Y: {2,4,8}
Z :{2,3,6}

Let R : X → Y, and S : Y → Z, be the relation "less than":

R = {(1,2),(1,4),(1,8),(3,4),(3,8),(5,8)}
S = {(2,3),(2,6),(4,6)}

S ° R :{(1,3), (1,6), (3,6)}

The element (1,3) is formed by combining (1,2) from R and (2,3) from S
The element (1,6) is formed by combining (1,2) from R and (2,6) from S
Note, that (1,6) can also be obtained by combining (1,4) from R and (4,6) from S.
The element (3,6) is formed by combining (3,4) from R and (4,6) from S

4. Identity relation

Identity relation on a set A is defined in the following way:

I = {(x,x)|x ∈A}

Example:

Let A = {a, b, c}, I = {(a,a), (b,b), (c,c)}

5. Problems:

Let A = {1, 2, 3}, B = {a, b}, C = {x, y, z}

a. Let R = {(1,a), (2,b), (3,a)} and S = {(a,y),(a,z),(b,x),(b,z)}

Find S ° R

b. Let R = {(1,a), (2,b), (3,a)} and S = {(a,y),(a,z)}

Find S ° R

a. Let R = {(1,a), (2,b)} and S = {(a,y), (b, y), (b,z)}
Find S ° R

b. Let R = {(1,a), (2,b), (3,a)} and S = {(a,y),(a,z),(b,x),(b,z)}
Find R^{-1} , S^{-1} and $R^{-1} \circ S^{-1}$

Solutions

Let A = {1, 2, 3}, B = {a, b}, C = {x, y, z}

a. Let R = {(1,a), (2,b), (3,a)} and S = {(a,y),(a,z),(b,x),(b,z)}

Find S ° R

Solution: {(1,y), (1, z), (2,x),(2,z), (3,y), (3, z)}

b. Let R = {(1,a), (2,b), (3,a)} and S = {(a,y),(a,z)}

Find S ° R Solution: {(1,y), (1, z), (3,y), (3, z)}

c. Let R = {(1,a), (2,b)} and S = {(a,y), (b, y), (b,z)}
Find S ° R
Solution: {(1,y), (2,y), (2, z)}

d. Let R = {(1,a), (2,b), (3,a)} and S = {(a,y),(a,z),(b,x),(b,z)}
Find R^{-1} , S^{-1} and $R^{-1} \circ S^{-1}$

Solution:
R^{-1} = {(a,1), (b,2), (a,3)}
S^{-1} = {(y,a),(z,a),(x,b),(z,b)}

$R^{-1} \circ S^{-1}$ = {(y,1), (y,3), (x,2), (z,1), (z,3), (z,2)}

Definitions:

Let R be a binary relation on a set A.

1. R is **reflexive,** iff for all $x \in A$, $(x,x) \in R$, i.e. **xRx** is true.

2. R is **symmetric**, iff for all $x, y \in A$, if $(x, y) \in R$, then $(y, x) \in R$

   i.e **xRy** → **yRx** is true

3. R is **transitive** iff for all $x, y, z \in A$, if $(x, y) \in R$ and $(y,z) \in R$, then $(x, z) \in R$

   i.e. **(xRy Λ yRz)** → **xRz** is true

A. Reflexive relations

Let R be a binary relation on a set A.

R is **reflexive, iff** for all $x \in A$, $(x,x) \in R$, i.e. **xRx** is true.

1. Examples:

1. Equality is a reflexive relation

   for any object x: $x = x$ is true.
2. "less then" (defined on the set of real numbers) is **not a reflexive relation.**

   for any number x: $x < x$ is **not** true
3. " less then or equal to" (defined on the set of real numbers) **is a reflexive relation**

   for any number x $x \le x$ is true

4. Reflexive and irreflexive relations

Compare the three examples below:

1. A = {1,2,3,4}, R1 = {(1,1), (1,2), (2,2), (2,3), (3,3), (3,4), (4,4)}
2. A = {1,2,3,4}, R2 = {(1,2), (2,3), (3,4), (4,1)}
3. A = {1,2,3,4}, R3= {(1,1), (1,2), (3,4), (4,4)}

R1 is a reflexive relation. R2 ? R3 ?

Definition: Let R be a binary relation on a set A.
R is **irreflexive** iff for all $x \in A$, $(x,x) \notin R$

Definition: Let R be a binary relation on a set A.
R is **neither reflexive, nor irreflexive iff**
there is $x \in A$, such that $(x, x) \in R$. and there is $y \in A$ such that $(y, y) \notin R$

Thus R2 is irreflexive, R3 is neither reflexive nor irreflexive.

reflexive:	for all x: xRx
irreflexive:	for no x: xRx
neither:	for some x: xRx is true, for some y: yRy is false

B. Symmetric relations

R is **symmetric**, iff for all $x, y \in A$, if $(x, y) \in R$, then $(y, x) \in R$

i.e **xRy** $\rightarrow$ **yRx** is true

This means: if two elements **x** and **y** are in relation R, then **y** and **x** are also in R, i.e. if xRy is true, yRx is also true.

1. Examples:

1. equality is a symmetric relation: if a = b then b = a
2. "less than" is <u>not</u> a symmetric relation : if a < b is true then b < a is false
3. "sister" on the set of females is symmetric
4. "sister" on the set of all human beings is not symmetric

4. Symmetric and anti-symmetric relations

Compare the relations:

1. A = {1,2,3,4}, R1 = {(1,1), (1,2), (2,1), (2,3), (3,2), (4,4)}
2. A = {1,2,3,4}, R2 = {(1,1), (1,2), (2,3), (4,4)}
3. A = {1,2,3,4}, R3 = {(1,1), (1,2), (2,1) , (2,3), (4,4)}

Definition: Let R be a binary relation on a set A.
R is **anti-symmetric** if for all x, y $\in$ A, x $\neq$ y, **if (x, y) $\in$ R, then (y, x) $\notin$ R.**

Definition: R is **neither symmetric nor anti-symmetric** iff it is not symmetric and not anti-symmetric.

symmetric: xRy → yRx for all x and y
anti-symmetric: xRy and yRx → x = y
neither: for some x and y: xRy, and yRx
for others xRy is true, yRx is not true

C. Transitive relations

Let R be a binary relation on a set A.
R is **transitive** iff for all x, y, z $\in$ A, if (x, y) $\in$ R and (y,z) $\in$ R , then (x, z) $\in$ R

i.e. **(xRy $\wedge$ yRz) → xRz** is true

1. Examples:

1. Equality is a transitive relation a = b, b = c, hence a = c
2. "less than" is a transitive relation a < b, b < c, hence a < c
3. mother_of(x,y) is <u>not</u> a transitive relation
4. sister(x,y) is a transitive relation
5. brother (x,y) is a transitive relation.
6. A = {1,2,3,4} R = {(1,1), (1,2), (1,3), (2,3), (4,3)} - transitive
7. A = {1,2,3,4} R = {(1,1), (1,2), (1,3), (2,3), (3,4)} - not transitive

Equivalence Relations, Partial Orders

Compare the relations:

1. **Equivalence relations**

Definition: A relation R is an equivalence relation if and only if it is reflexive, symmetric, and transitive.

Examples:

Let **m** and **n** be integers and let **d** be a positive integer. The notation

$$m \equiv n \pmod{d}$$

is read "m is congruent to n modulo d".

The meaning is: the integer division of **d** into **m** gives the same remainder as the integer division of **d** into **n**.

Consider the relation

$$R=\{(x,y) \mid x \bmod 3 = y \bmod 3\}$$

4 mod 3 = 1, 7 mod 3 = 1, hence (4,7) $\in$ R

The relation is <u>reflexive</u>: $x \bmod 3 = x \bmod 3$
<u>symmetric</u>: if $x \bmod 3 = y \bmod 3$, then $y \bmod 3 = x \bmod 3$
<u>transitive</u>: if $x \bmod 3 = y \bmod 3$, and $y \bmod 3 = z \bmod 3$, then $x \bmod 3 = z \bmod 3$

Consider the sets $[x] = \{y \mid yRx\}$

$[0] = \{0,3,6,9,12,\ldots.\}$
$[1] = \{1,4,7,10,13,\ldots.\}$
$[2] = \{2,5,8,11,14,\ldots\}$
……

From the definition of [x] it follows that

$[0] = [3] = [6] \ldots$
$[1] = [4] = \ldots$
$[2] = [5] = \ldots$

Thus the relation R produces three different sets [0], [1] and [2].
Each number is exactly in one of these sets. Thus {[0], [1], [2]} is a **partition** of the set of non-negative integers.

2. Partial Orders

Definition: Let R be a binary relation defined on a set A. R is a partial order relation iff R is transitive and anti-symmetric

Examples:

1. Let A be a set, and P(A) be the power set of A. The relation 'subset of' on P (A) is a partial order relation
   It is reflexive, anti-symmetric, and transitive

2. Let N be the set of positive integers, and R be a relation defined as follows:
   $(x, y) \in R$ iff y is a multiple of x
   e.g. $(3,12) \in R$, while $(3, 4) \notin R$
   R is a partial order relation. It is reflexive, anti-symmetric, and transitive

Functions

1. Definition: A function **f** from a set **X** to a set **Y** is a subset of the Cartesian pr
X x Y, $f \subseteq X \times Y$, such that

$$\forall x \in X \quad \exists y \in Y, \text{ such that } (x,y) \in f, \text{ and}$$

$$(x,y1) \in f \ \Lambda \ (x,y2) \in f \ \rightarrow y1 = y2$$

i.e. if $(x,y1) \in f$ and $(x,y2) \in f$, then $y1 = y2$

Thus all elements in X can be found in exactly one pair of **f.**

Notation: Let **f** be a function from **A** to **B.** We write

$$f : A \rightarrow B$$
$$a \in A,\ f(a) = b,\ \ b \in B$$

Examples:

$A = \{1,2,3\}$, $B = \{a,b\}$

$R = \{(1,a),(2,a),(3,b)\}$ is a function

Other definitions:

Let **f** be a function from **A** to **B.**

1. **Domain** of **f:** the set **A**
2. **Range** of **f**: $\{b: b \in B$ and there is an $a \in A$, $f(a) = b\}$
3. **Image** of a under **f: f(a)**

Example:

$A = \{1,2,3\}$, $B = \{a,b\}$

$f = \{(1,a),(2,a),(3,b)\}$
domain: $\{1,2,3\}$,
range: $\{a,b\}$
a is image of 1 under f: $f(1) = a$, $f(2) = \ldots\ldots$ $f(3) = \ldots\ldots$

2. Functions with more arguments

Let A = A1 x A2 , and **f** be a function from **A** to **B**

We write: f(a1,a2) = b

If A = A1 x A2 x … x An, we write f(a1,a2,…,an) = b

a1, a2, ..,an: arguments of **f**
b : value of **f**

3. Functions of special interest

a. one-to-one

distinct elements have distinct images

if a1 $\neq$ a2, then f(a1) $\neq$ f(a2)

Example:
A = {1,2,3}, B = {a,b,c,d}

one-to-one function **f** = {(1,a), (2,c), (3,b)}

b. onto

Every element in **B** is an image of some element in **A**

Example:

A = {1,2,3}, B = {a,b }

onto function f = {(1,a), (2,b), (3,b)}

c. bijection

f is **bijection** iff **f** is a **one-to-one** function and **f** is a **onto** function

Example:

A = {1,2,3}, B = {a,b,c }

bijection f = {(1,a), (2,c), (3,b)}

4. Inverse function

If **f is a bijection,** f^{-1} is a function, also a bijection.

$f^{-1} = \{(y,x) \mid (x,y) \in f\}$

Example:

A = {1,2,3}, B = {a,b,c }

f = {(1,a), (2,c), (3,b)}

$f^{-1} = \{(a,1), (b,3), (c,2)\}$

5. Composition of functions

Let **f : A → B, g : B → C** be two functions.
The composition **h = g ° f** is a function from **A** to **C** such that **h(a) = g(f(a)**

Example: Let f (x) = x +1, g(x) = x^2.

The composition h(x) = f (x) ° g(x) = f(g(x)) = (x^2)+ 1

The composition p(x) = g(x) ° f(x) = g(f(x))= $(x+1)^2$

When **f** is a bijection and **f^{-1}** exists, we have:
$f^{-1}(f(a)) = a$, $f(f^{-1}(b)) = b$, a ϵ A, b ϵ B,

Counting Principles

The Multiplication Principle

The Multiplication Principle

Let $m \in \mathbb{N}$. For a procedure of m successive distinct and independent steps with n_1 outcomes possible for the first step, n_2 outcomes possible for the second step, . . . , and n_m outcomes possible for the mth step, the total number of possible outcomes is

$$n_1 \cdot n_2 \cdots n_m$$

Addition Principle

The Addition Principle

For a collection of m disjoint sets with n_1 elements in the first, n_2 elements in the second, . . . , and n_m elements in the mth, the number of ways to choose one element from the collection is

$$n_1 + n_2 + \cdots + n_m$$

POSSIBLE QUESTIONS

TWO MARKS

1. Define finite set .
2. Define partial order Relations.
3. Define Equivalence Relations
4. Define Equal function.
5. Define constant function.

SIX MARKS

1. Explain about types of relation with examples.
2. Explain types of sets .
3. Explain types of functions .
4. If A,B,C,D of four sets and f,g and h are 3 functions defined as
   f:A→ B $g: B \to C$ &$h: C \to A$ then prove that(hog)of=ho(gof).
5. If f:X→ Y and A,B are two sunsets of Y , then prove that (a) $f^{-1}(AUB) = f^{-1}(A) \cup f^{-1}(B)$
   b) $f^{-1}(A \cap B) = f^{-1}(A) \cap f^{-1}(B)$
6. If R is the set of real numbers, then show that the function ,f:R→R defined by f(x) $5x^3$ -1 is one-one onto function.
7. Let A={ 1,2,3,4} ,B={a,b,c,d} and C={x,y,z}.Consider the function f:A→B and g:B→C defined by f={ (1,a),(2,c),(3,b),(4,a)} and g={ (a,x),(b,x),(c,y),(d,y) }.Find the Composition function (gof).
8.) Let A={1,2,3} and f,g,h and s be functions from A to A given by
   f ={ (1,2), (2,3),(3,1) } ; g = { (1,2), (2,1), (3,3) } ;
   h = { (1,1), (2,2), (3,1) } and s = { (1,1), (2,2), (3,3) }. Find $f_o g$, $g_o f$, $f_o h_o g$, $g_o s$, $s_o s$, $f_o s$.

UNIT – II

Growth of Functions: Pigeonhole principle,permutation and combination,Mathematical Induction,Principle of Inclusion and Exclusion.

Growth of Functions

- We will use something called *big-O notation* (and some siblings described later) to describe how a function grows.
  - What we're trying to capture here is how the function grows.
  - … without capturing so many details that our analysis would depend on processor speed, etc.
  - … without worrying about what happens for small inputs: they should always be fast.
- For functions $f(x)$ and $g(x)$, we will say that "$f(x)$ is $O(g(x))$" [pronounced "$f(x)$ is big-oh of $g(x)$"] if there are positive constants C and k such that

$$|f(x)| \le C|g(x)| \text{ for all } x > k.$$

  - The big-O notation will give us a order-of-magnitude kind of way to describe a function's growth (as we will see in the next examples).
  - Roughly speaking, the k lets us only worry about big values (or input sizes when we apply to algorithms), and C lets us ignore a factor difference (one, two, or ten steps in a loop).
  - I might also say "$f(x)$ is in $O(g(x))$", then thinking of $O(g(x))$ as the set of all functions with that property.
- *Example:* The function $f(x)=2x^3+10x$ is $O(x^3)$.

  Proof: To satisfy the definition of big-O, we just have to find values for C and k that meet the condition.

  Let $C=12$ and $k=2$. Then for $x>k$,

$$|2x^3+10x| = 2x^3+10x < 2x^3+10x^3 = |12x^3|. \blacksquare$$

- Note: there's nothing that says we have to find the *best* C and k. Any will do.
  - Also notice that the absolute value doesn't usually do much: since we're worried about running times, negative values don't usually come up. We can just demand that x is big enough that the function is definitely positive and then remove the $|\cdots|$.
- Now it sounds too easy to put a function in a big-O class. But…
- *Example:* The function $f(x)=2x^3+10x$ is **not in** $O(x^2)$.

Proof: Now we must show that no C and k exist to meet the condition from the definition.

For any candidate C and k, we can take $x>k$ and $x>0$ and we would have to satisfy

$$|2x^3+10x| 2x^3+10x 2x^3 x<C|x^2|<Cx^2<Cx^2<C/2$$

So no such C and k can exist to let the inequality hold for large x.■

- *Example:* The function $f(x)=2x^3+10x$ is $O(x^4)$.

  Proof idea: For large x, we know that $x^4>x^3$. We could easily repeat the $O(x^3)$ proof above, applying that inequality in a final step.

- *Example:* The function $f(x)=5x^2-10000x+7$ is $O(x^2)$.

  Proof: We have to be a little more careful about negative values here because of the "$-10000x$" term, but as long as we take $k\geq 2000$, we won't have any negative values since the $5x^2$ term is larger there.

  Let $C=12$ and $k=2000$. Then for $x>k$,

  $$|5x^2-10000x+7|=5x^2-10000x+7<5x^2+7x^2=|12x^2|.$$ ■

- It probably wouldn't take many more proofs to convince you that x^n is always in $O(x^n)$ but never in $O(x^{n-1})$.
  - We can actually do better than that…
- The big-O operates kind of like a $\leq$ for growth rates.

Big-O Results

- *Theorem:* Any degree-n polynomial, $f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$ is in $O(x^n)$.

  Proof: As before, we can assume that $x>1$ and then,

  $$|f(x)|=|a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0|\le|a_n|x^n+|a_{n-1}|x^{n-1}+\cdots+|a_1|x+|a_0|=x^n(|a_n|+|a_{n-1}|/x+\cdots+|a_1|/x^{n-1}+|a_0|/x^n)\le x^n(|a_n|+|a_{n-1}|+\cdots+|a_1|+|a_0|).$$

  Now, if we let $C=\sum|a_i|$ and $k=1$, we have satisfied the definition for $O(x^n)$. ■

- *Theorem:* If we have two functions $f_1(x)$ and $f_2(x)$ both $O(g(x))$, then $f_1(x)+f_2(x)$ is also $O(g(x))$.

  Proof: From the definition of big-O, we know that there are C_1 and k_1 that make $|f_1(x)|\le C|h(x)|$ for $x>k_1$, and similar C_2 and k_2 for $f_2(x)$.

  Let $C=C_1+C_2$ and $k=\max(k_1,k_2)$. Then for $x>k$,

  $$|f_1(x)+f_2(x)|\le|f_1(x)|+|f_2(x)|\le C_1|g(x)|+C_2|g(x)|=C|g(x)|.$$

  Thus, $f_1(x)+f_2(x)$ is $O(g(x))$. ■

- The combination of functions under big-O is generally pretty sensible…
  - *Theorem:* If for large enough x, we have $f(x)\le g(x)$, then $f(x)$ is $O(g(x))$.
    - Sometimes the big-O proof is even easier.
  - *Theorem:* If we have two functions $f_1(x)$ which is $O(g_1(x))$ and $f_2(x)$ which is $O(g_2(x))$, then $f(x)+g(x)$ is $O(\max(|g_1(x)|,|g_2(x)|))$.
    - When adding, the bigger one wins.

- *Theorem:* If we have three functions f,g,h where $f(x)$ is $O(g(x))$ and $g(x)$ is $O(h(x))$, then $f(x)$ is $O(h(x))$.
  - Approximately: if h is bigger than g and g is bigger than f, then h is bigger than f.
- *Corollary:* Given $f_1(x)$ which is $O(g_1(x))$ and $f_2(x)$ which is $O(g_2(x))$ **and** $g_1(x)$ is $O(g_2(x))$ then $f_1(x)+f_2(x)$ is $O(g_2(x))$.
  - That is, if we have two functions we know a big-O bound for, and we add them together, we can ignore the smaller one in the big-O.
- *Theorem:* If we have two functions $f_1(x)$ which is $O(g_1(x))$ and $f_2(x)$ which is $O(g_2(x))$, then $f(x)g(x)$ is $O(g_1(x)g_2(x))$.
  - Multiplication happens in the obvious way.
- *Theorem:* Any constant value is is $O(1)$.
  - Aside: You will often hear a constant running time algorithm described as $O(1)$.
- *Corollary:* Given $f(x)$ which is $O(g(x))$ and a constant a, we know that $af(x)$ is $O(g(x))$.
  - That is, if we have a function multiplied by a constant, we can ignore the constant in the big-O.

- All of that means that it's usually pretty easy to guess a good big-O category for a function.
  - $f(x)=2^x+x^2$ is in $O(\max(|2^x|,|x^2|))=O(2^x)$, since 2^x is larger than x^2 for large x.
  - $f(x)=\frac{1}{100}x^{12}+100x^{11}-87$ is in $O(x^{12})$.
    - Directly from the theorem about polynomials.
    - For small x, the $100x^{11}$ is the largest, but as x grows, the $\frac{1}{100}x^{12}$ term takes over.
  - $f(x)=14x2^x+x$ is in $O(x2^x)$.

- What is a good big-O bound for $17x^4-12x^2+\log_2 x$?
  - We can start with the obvious:

$$17x^4-12x^2+\log_2 x \text{ is in } O(17x^4-12x^2+\log_2 x).$$

  - From the above, we know we can ignore smaller-order terms:

$$17x^4-12x^2+\log_2 x \text{ is in } O(17x^4).$$

- And we can ignore leading constants:

$$17x^4-12x^2+\log_2 x \text{ is in } O(x^4).$$

- The "ignore smaller-order terms and leading constants" trick is very useful and comes up a lot.

Big-Ω

- As mentioned earlier, big-O feels like $\leq$ for growth rates.
    - … then there must be $\geq$ and $=$ versions.
- We will say that a function $f(x)$ is $\Omega(g(x))$ ("big-omega of $g(x)$") if there are positive constants C and k such that when $x>k$,

$$|f(x)|\geq C|g(x)|.$$

- This is the same as the big-O definition, but with a $\geq$ instead of a $\leq$.
- *Example:* The function $3x^2+19x$ is $\Omega(x^2)$.

Proof: If we let $C=3$ and $k=1$ then for $x>k$,

$$|3x^2+19x|\geq 3x^2+19x\geq 3|x^2|.$$

From the definition, we have that $3x^2+19x$ is $\Omega(x^2)$. ■

- As you can guess, the proofs of big-Ω are going to look just about like the big-O ones.
    - We have to be more careful with negative values: in the big-O proofs, we could just say that the absolute value was bigger and ignore it. Now we need smaller values, so can't be so quick.
    - But the basic ideas are all the same.
- *Theorem:* $f(x)$ is $O(g(x))$ iff $g(x)$ is $\Omega(f(x))$.

Proof: First assume we have $f(x)$ in $O(g(x))$. Then there are positive C and k so that when $x>k$, we know $|f(x)|\leq C|g(x)|$. Then for $x>k$, we have $|g(x)|\geq \frac{1}{C}|f(x)|$ and we can use k and $\frac{1}{C}$ as constants for the definition of big-Ω.

Similarly, if we assume that $g(x)$ is $\Omega(f(x))$, we have positive C and k so that when $x>k$, we have $|g(x)|\geq C|f(x)|$. As above we then have for $x>k$, $|f(x)|\leq \frac{1}{C}|g(x)|$. ■

Big-Θ

- We will say that a function $f(x)$ is $\Theta(g(x))$ ("big-theta of $g(x)$") if $f(x)$ is **both** $O(g(x))$ and $\Omega(g(x))$.
  - For a function that is $\Theta(g(x))$, we will say that that function "is order $g(x)$."
- *Example:* The function 2^x+x^2 is order 2^x.

Proof: To show that 2^x+x^2 is $O(2^x)$, we can take $C=2$ and $k=4$. Then for $x>k$,

$$|2^x+x^2|=2^x+x^2\leq 2\cdot 2^x.$$

To show that 2^x+x^2 is $\Omega(2^x)$, we can use $C=1$ and $k=1$. For $x>k$,

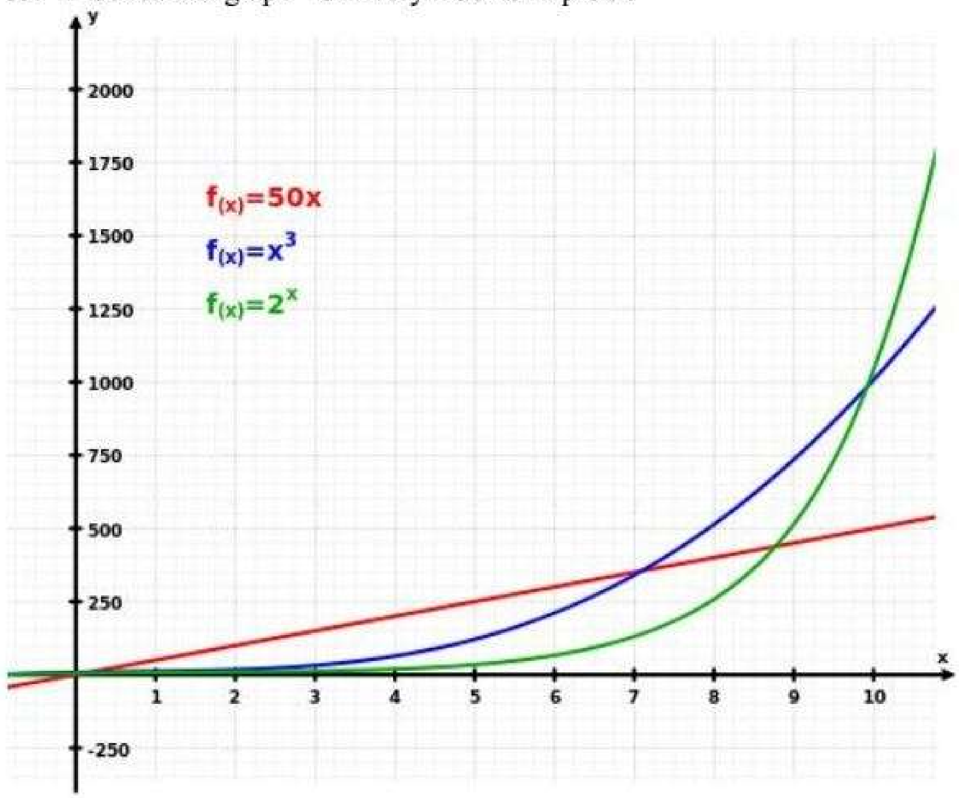
$$|2^x+x^2|=2^x+x^2\geq 2^x.$$

Thus, 2^x+x^2 is $\Theta(2^x)$. ■

- The above theorem gives another way to show big-Θ: if we can show that $f(x)$ is $O(g(x))$ and $g(x)$ is $O(f(x))$, then $f(x)$ is $\Theta(g(x))$.
- *Theorem:* Any degree-n polynomial with $a_n\neq 0$, $f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$ with $a_n>0$ is in $\Theta(x^n)$.
- A few results on big-Θ…
  - *Theorem:* If we have two functions $f_1(x)$ which is $\Theta(g_1(x))$ and $f_2(x)$ which is $\Theta(g_2(x))$, and $g_2(x)$ is $O(g_1(x))$, then $f_1(x)+f_2(x)$ is $\Theta(g_1(x)))$.
    - That is, when adding two functions together, the bigger one "wins".

- *Theorem:* If we have two functions $f_1(x)$ which is $\Theta(g(x))$ and $f_2(x)$ which is $O(g(x))$, then $f(x)+g(x)$ is $\Theta(g(x)))$.
- *Theorem:* for a positive constant a, a function $af(x)$ is $\Theta(g(x))$ iff $f(x)$ is $\Theta(g(x))$.
  - That is, leading constants don't matter.

  - *Corollary:* Any degree-n polynomial, $f(x)=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$ with $a_n>0$ is in $\Theta(x^n)$.
- What functions have a "higher" big-Θ than others is usually fairly obvious from a graph, but "I looked at a graph" isn't very much of a proof.



Source: [Wikipedia Exponential.svg]

- The big-O notation sets up a hierarchy of function growth rates. Here are some of the important "categories":

$$n!\quad 2^n\quad n^3\quad n^2\quad n\log n\quad n\quad \sqrt{n}=n^{1/2}\quad \log n\quad 1$$

  - Each function here is big-O of ones above it, but not below.
  - e.g. $n\log n$ is $O(n^2)$, but n^2 is **not** $O(n\log n)$.
  - So in some important way, n^2 grows faster than $n\log n$.

- Where we are headed: we will be able to look at an algorithm and say that one that takes $O(n\log n)$ steps is faster than one that takes $O(n2)$ steps (for large input).

Asymptotic Notation

9.7.1 Little Oh

Definition 9.7.1. For functions $f, g : \mathbb{R} \to \mathbb{R}$, with g nonnegative, we say f is *asymptotically smaller* than g, in symbols,

$$f(x) = o(g(x)),$$

iff

$$\lim_{x\to\infty} f(x)/g(x) = 0.$$

For example, $1000x^{1.9} = o(x^2)$, because $1000x^{1.9}/x^2 = 1000/x^{0.1}$ and since $x^{0.1}$ goes to infinity with x and 1000 is constant, we have $\lim_{x\to\infty} 1000x^{1.9}/x^2 = 0$. This argument generalizes directly to yield

9.7.2 Big Oh

Big Oh is the most frequently used asymptotic notation. It is used to give an upper bound on the growth of a function, such as the running time of an algorithm.

Definition 9.7.5. Given nonnegative functions $f, g : \mathbb{R} \to \mathbb{R}$, we say that

$$f = O(g)$$

iff

$$\limsup_{x\to\infty} f(x)/g(x) < \infty.$$

This definition[12] makes it clear that

Definition 9.7.12. Given functions $f, g : \mathbb{R} \to \mathbb{R}$, we say that

$$f = \Omega(g)$$

iff there exists a constant $c > 0$ and an x_0 such that for all $x \geq x_0$, we have $f(x) \geq c|g(x)|$.

In other words, $f(x) = \Omega(g(x))$ means that $f(x)$ is greater than or equal to $g(x)$, except that we are willing to ignore a constant factor and to allow exceptions for small x.

If all this sounds a lot like big-Oh, only in reverse, that's because big-Omega is the opposite of big-Oh. More precisely,

Little Omega

There is also a symbol called little-omega, analogous to little-oh, to denote that one function grows strictly faster than another function.

Definition 9.7.14. For functions $f, g : \mathbb{R} \to \mathbb{R}$ with f nonnegative, we say that

$$f(x) = \omega(g(x))$$

iff

$$\lim_{x \to \infty} \frac{g(x)}{f(x)} = 0.$$

In other words,

$$f(x) = \omega(g(x))$$

iff

$$g(x) = o(f(x)).$$

Definition 9.7.15.

$$f = \Theta(g) \quad \text{iff} \quad f = O(g) \text{ and } g = O(f).$$

The statement $f = \Theta(g)$ can be paraphrased intuitively as "f and g are equal to within a constant factor." Indeed, by Theorem 9.7.13, we know that

$$f = \Theta(g) \quad \text{iff} \quad f = O(g) \text{ and } f = \Omega(g).$$

Example: $n^2 + n = O(n^3)$

Proof:

- Here, we have $f(n) = n^2 + n$, and $g(n) = n^3$
- Notice that if $n \geq 1$, $n \leq n^3$ is clear.
- Also, notice that if $n \geq 1$, $n^2 \leq n^3$ is clear.
- **Side Note:** In general, if $a \leq b$, then $n^a \leq n^b$ whenever $n \geq 1$. This fact is used often in these types of proofs.
- Therefore,

$$n^2 + n \leq n^3 + n^3 = 2n^3$$

- We have just shown that

$$n^2 + n \leq 2n^3 \text{ for all } n \geq 1$$

- Thus, we have shown that $n^2 + n = O(n^3)$ (by definition of Big-O, with $n_0 = 1$, and $c = 2$.)

Example: $n^3 + 4n^2 = \Omega(n^2)$

Proof:

- Here, we have $f(n) = n^3 + 4n^2$, and $g(n) = n^2$
- It is not too hard to see that if $n \geq 0$,

$$n^3 \leq n^3 + 4n^2$$

- We have already seen that if $n \geq 1$,

$$n^2 \leq n^3$$

- Thus when $n \geq 1$,

$$n^2 \leq n^3 \leq n^3 + 4n^2$$

- Therefore,

$$1n^2 \leq n^3 + 4n^2 \text{ for all } n \geq 1$$

- Thus, we have shown that $n^3 + 4n^2 = \Omega(n^2)$ (by definition of Big-Ω, with $n_0 = 1$, and $c = 1$.)

Example: $n^2 + 5n + 7 = \Theta(n^2)$

Proof:

- When $n \geq 1$,

$$n^2 + 5n + 7 \leq n^2 + 5n^2 + 7n^2 \leq 13n^2$$

- When $n \geq 0$,

$$n^2 \leq n^2 + 5n + 7$$

- Thus, when $n \geq 1$

$$1n^2 \leq n^2 + 5n + 7 \leq 13n^2$$

Thus, we have shown that $n^2 + 5n + 7 = \Theta(n^2)$ (by definition of Big-Θ, with $n_0 = 1$, $c_1 = 1$, and $c_2 = 13$.)

Show that $\frac{1}{2}n^2 + 3n = \Theta(n^2)$

Proof:

- Notice that if $n \geq 1$,

$$\frac{1}{2}n^2 + 3n \leq \frac{1}{2}n^2 + 3n^2 = \frac{7}{2}n^2$$

- Thus,

$$\frac{1}{2}n^2 + 3n = O(n^2)$$

- Also, when $n \geq 0$,

$$\frac{1}{2}n^2 \leq \frac{1}{2}n^2 + 3n$$

- So

$$\frac{1}{2}n^2 + 3n = \Omega(n^2)$$

- Since $\frac{1}{2}n^2 + 3n = O(n^2)$ and $\frac{1}{2}n^2 + 3n = \Omega(n^2)$,

$$\frac{1}{2}n^2 + 3n = \Theta(n^2)$$

Show that $(n \log n - 2n + 13) = \Omega(n \log n)$

Proof: We need to show that there exist positive constants c and n_0 such that

$0 \le c n \log n \le n \log n - 2n + 13$ for all $n \ge n_0$.

Since $\quad n \log n - 2n \le n \log n - 2n + 13,$

we will instead show that

$$c n \log n \le n \log n - 2n,$$

which is equivalent to

$$c \le 1 - \frac{2}{\log n}, \text{ when } n > 1.$$

If $n \ge 8$, then $2/(\log n) \le 2/3$, and picking $c = 1/3$ suffices. Thus if $c = 1/3$ and $n_0 = 8$, then for all $n \ge n_0$, we have

$0 \le c\,n\log n \le n\log n - 2\,n \le n\log n - 2\,n + 13.$

Thus $(n\log n - 2\,n + 13) = \Omega(n\log n)$.

Show that $\frac{1}{2}n^2 - 3n = \Theta(n^2)$

Proof:

- We need to find positive constants c_1, c_2, and n_0 such that

$$0 \le c_1 n^2 \le \frac{1}{2}n^2 - 3\,n \le c_2 n^2 \text{ for all } n \ge n_0$$

- Dividing by n^2, we get

$$0 \le c_1 \le \frac{1}{2} - \frac{3}{n} \le c_2$$

- $c_1 \le \frac{1}{2} - \frac{3}{n}$ holds for $n \ge 10$ and $c_1 = 1/5$
- $\frac{1}{2} - \frac{3}{n} \le c_2$ holds for $n \ge 10$ and $c_2 = 1$.
- Thus, if $c_1 = 1/5$, $c_2 = 1$, and $n_0 = 10$, then for all $n \ge n_0$,

$$0 \le c_1 n^2 \le \frac{1}{2}n^2 - 3\,n \le c_2 n^2 \text{ for all } n \ge n_0.$$

Thus we have shown that $\frac{1}{2}n^2 - 3n = \Theta(n^2)$.

Summary of the Notation

- $f(n) = O(g(n)) \Rightarrow f \preceq g$
- $f(n) = \Omega(g(n)) \Rightarrow f \succeq g$
- $f(n) = \Theta(g(n)) \Rightarrow f \approx g$
- It is important to remember that a Big-O bound is only an *upper bound*. So an algorithm that is $O(n^2)$ might not ever take that much time. It may actually run in $O(n)$ time.
- Conversely, an Ω bound is only a *lower bound*. So an algorithm that is $\Omega(n \log n)$ might actually be $\Theta(2^n)$.
- Unlike the other bounds, a Θ-bound is precise. So, if an algorithm is $\Theta(n^2)$, it runs in quadratic time.

POSSIBLE QUESTION

TWO MARKS

1. Prove that the function $f(x) = 2x^3+10x$ is $O(x^3)$.
2. Prove that the function $3x^2+19x$ is $\Omega(x^2)$.
3. Prove that $n^2 + 5n + 7 = \Theta(n^2)$.
4. Evaluate $\sum_{k=1}^{9}(5k+8)$.
5. Evaluate the limit n tends to infinity $\lim_{n\to\infty}\frac{(2n+1)^2}{5n^2+2n+1}$

SIX MARKS

1. Show that $(n \log n - 2n + 13) = \Omega(n \log n)$.
2. Evaluate the sum $\sum_{k=1}^{8}(5k^2+8k+1)$
3. Evaluate the sum $\sum_{k=7}^{12}(k+1)$
4. Evaluate the sum -5-4-3-2-1+0+1+2+3+4+.....+30
5. Evaluate the limit n tends to infinity $\lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\left(k\frac{7}{n}\right)^2$
6. The function $f(x)=2x^3+10x$ is $o(x^4)$.
7. Evaluate the limit n tends to infinity $\lim_{n\to\infty}\frac{1}{n}\sum_{k=1}^{n}\frac{k+9}{n}$.
8. Show that if we have two functions $f_1(x)$ and $f_2(x)$ both $O(g(x))$, then $f_1(x)+f_2(x)$ is also $O(g(x))$.
9. If $\sum_{k=1}^{n}k^4 = \frac{4n(n+1)(2n+1)(3n^2+3n-1)}{A}$ then find A?
10. Show that $\frac{1}{2}n^2 - 3n = \Theta(n^2)$.
11. The integral $\int_0^5 x^2dx$ is computed as the limit of the sum $\sum_{k=1}^{n}\frac{A}{n}\left(k\frac{A}{n}\right)^2$. What value of A must appear in the sum ?

UNIT – III

Recurrences: Recurrence relations, generating functions, linear recurrence relations with constant coefficients and their solution.

Solving the Recurrence

Claim 10.1.1. *$T_n = 2^n - 1$ satisfies the recurrence:*

$$T_1 = 1$$
$$T_n = 2T_{n-1} + 1 \qquad (\text{for } n \geq 2).$$

Proof. The proof is by induction on n. The induction hypothesis is that $T_n = 2^n - 1$. This is true for $n = 1$ because $T_1 = 1 = 2^1 - 1$. Now assume that $T_{n-1} = 2^{n-1} - 1$ in order to prove that $T_n = 2^n - 1$, where $n \geq 2$:

$$\begin{aligned} T_n &= 2T_{n-1} + 1 \\ &= 2(2^{n-1} - 1) + 1 \\ &= 2^n - 1. \end{aligned}$$

Linear Recurrences

In general, a *homogeneous linear recurrence* has the form

$$f(n) = a_1 f(n-1) + a_2 f(n-2) + \ldots + a_d f(n-d)$$

where $a_1, a_2, \ldots, a_d$ and d are constants. The *order* of the recurrence is d. Commonly, the value of the function f is also specified at a few points; these are called *boundary conditions*. For example, the Fibonacci recurrence has order $d = 2$ with coefficients $a_1 = a_2 = 1$ and $g(n) = 0$. The boundary conditions are $f(0) = 1$ and $f(1) = 1$. The word "homogeneous" sounds scary, but effectively means "the simpler kind". We'll consider linear recurrences with a more complicated form later.

Theorem 10.3.1. *If $f(n)$ and $g(n)$ are both solutions to a homogeneous linear recurrence, then $h(n) = sf(n) + tg(n)$ is also a solution for all $s, t \in \mathbb{R}$.*

Proof.

$$\begin{aligned} h(n) &= sf(n) + tg(n) \\ &= s\,(a_1 f(n-1) + \ldots + a_d f(n-d)) + t\,(a_1 g(n-1) + \ldots + a_d g(n-d)) \\ &= a_1(sf(n-1) + tg(n-1)) + \ldots + a_d(sf(n-d) + tg(n-d)) \\ &= a_1 h(n-1) + \ldots + a_d h(n-d) \end{aligned}$$

Solving First-Order Recurrences Using Back Substitution

Theorem 2. **(Solution of First-Order Recurrence Relations)** The solution of

$$T(n) = \begin{cases} cT(n-1) + f(n) & \text{for } n \geq k \\ f(k) & \text{for } n = k \end{cases}$$

where c is a constant and f is a nonzero function of n for $n \geq k$ is

$$T(n) = \sum_{l=k}^{n} c^{n-l} f(l)$$

Motivation for the Proof. First, use back substitution to decide what the general form of the solution might be, and then prove by induction that this is the solution:

$$\begin{aligned} T(n) &= cT(n-1) + f(n) \\ &= c(cT(n-2) + f(n-1)) + f(n) \\ &= c^2T(n-2) + cf(n-1) + f(n) \\ &= c^2(cT(n-3) + f(n-2)) + cf(n-1) + f(n) \\ &= c^3T(n-3) + c^2f(n-2) + cf(n-1) + f(n) \end{aligned}$$

Using back substitution one more time gives

$$\begin{aligned} T(n) &= c^3[cT(n-4) + f(n-3)] + \sum_{l=n-2}^{n} c^{n-l} f(l) \\ &= c^4T(n-4) + c^3 f(n-3) + \sum_{l=n-2}^{n} c^{n-l} f(l) \\ &= c^4T(n-4) + \sum_{l=n-3}^{n} c^{n-l} f(l) \end{aligned}$$

If back substitution is continued until the argument of T is k—that is, for $n-k$ steps—then the expression for $T(n)$ becomes

$$\begin{aligned} T(n) &= c^{n-k}T(n-(n-k)) + \sum_{l=n-k+1}^{n} c^{n-l} f(l) \\ &= c^{n-k}T(k) + \sum_{l=n-k+1}^{n} c^{n-l} f(l) \end{aligned}$$

Since $T(k) = f(k)$, replace the reference to T on the right-hand side of the equation, getting

$$\begin{aligned} T(n) &= c^{n-k} f(k) + \sum_{l=n-k+1}^{n} c^{n-l} f(l) \\ &= \sum_{l=n-k}^{n} c^{n-l} f(l) \end{aligned}$$

Proof. By induction, show that

$$T(n) = \sum_{l=k}^{n} c^{n-l} f(l)$$

Let $n_0 = k$. Let $\mathcal{T} = \{n \in \mathbb{N} : n \geq k \text{ and } T(n) \text{ is a solution}\}$.

(Base step) First, show that

$$\sum_{l=k}^{n} c^{n-l} f(l)$$

is a solution for $n = k$ so that $k \in \mathcal{T}$.

$$\sum_{l=k}^{k} c^{k-l} f(l) = c^{k-k} f(k) = f(k) = T(k)$$

(Inductive step) Now, assume that $T(n)$ is given by this expression for $n \geq n_0$, that is, $T(n) = \sum_{l=k}^{n} c^{n-l} f(l)$. Now prove that $T(n+1)$ is also given by this expression: In this case, prove that $T(n+1) = \sum_{l=k}^{n+1} c^{n-l} f(l)$.

$$\begin{aligned} T(n+1) &= cT(n) + f(n+1) \quad \text{(Definition of recurrence relation)} \\ &= c\sum_{l=k}^{n} c^{n-l} f(l) + f(n+1) \quad \text{(Inductive hypothesis)} \\ &= \sum_{l=k}^{n} c^{n-l+1} f(l) + f(n+1) \\ &= \sum_{l=k}^{n+1} c^{n+1-l} f(l) \end{aligned}$$

This proves $n+1 \in \mathcal{T}$.

By the Principle of Mathematical Induction, $\mathcal{T} = \{n \in \mathbb{N} : n \geq k\}$. ■

Example 1. Solve

$$T(n) = \begin{cases} T(n-1) + n^2 & \text{for } n \geq 1 \\ 0 & \text{for } n = 0 \end{cases}$$

Solution. In the general formula, $f(n) = n^2$ for $n \geq 0$, $c = 1$, and $k = 0$. Since $T(0) = f(0)$, by Corollary 1 the solution is

$$T(n) = \sum_{l=1}^{n} l^2 = \frac{1}{6} \cdot (2n+1) \cdot n \cdot (n+1)$$

See Theorem 9(b) in Section 7.10 for a derivation of this formula. ■

Example 2. Solve

$$T(n) = \begin{cases} 3T(n-1) + 4 & \text{for } n \geq 1 \\ 4 & \text{for } n = 0 \end{cases}$$

Solution. In the general formula, $f(n) = 4$ for $n \geq 0$, $c = 3$, and $k = 0$. By Corollary 2, the solution is

$$T(n) = 4 \cdot \frac{3^{n+1} - 1}{3 - 1} = 2 \cdot (3^{n+1} - 1)$$ ■

Rules for Solving Second-Order Recurrence Relations

Solving Second-Order Homogeneous Recurrence Relations with Constant Coefficients Using the Complementary Equation with Distinct Real Roots

$$H(n) + AH(n-1) + BH(n-2) = 0,$$
$$H(n_1) = D, \text{ and } H(n_2) = E.$$

STEP 1: Assume $f(n) = c^n$ is a solution, and substitute for $H(n)$, yielding the characteristic equation

$$c^2 + Ac + B = 0$$

STEP 2: Find the roots of the characteristic equation: c_1 and c_2. Use the quadratic formula if the equation does not factor. If $c_1 \neq c_2$, then the general solution is

$$S(n) = Ac_1^n + Bc_2^n$$

STEP 3: Use the initial conditions to form the system of equations

$$H(n_1) = D = Ac_1^{n_1} + Bc_2^{n_2}$$
$$H(n_2) = E = Ac_1^{n_2} + Bc_2^{n_2}$$

STEP 4: Solve the system of equations found in step 3, getting A_0 and B_0 as the two solutions. Form the particular solution

$$H(n) = A_0 {c_1}^n + B_0 {c_2}^n$$

Example 1. Solve the recurrence relation $a_n - 6a_{n-1} - 7a_{n-2} = 0$ for $n \geq 5$ where $a_3 = 344$ and $a_4 = 2400$.

Solution. Form the characteristic equation and then factor it:

$$c^2 - 6c - 7 = 0$$
$$c = 7, -1$$

Form the general solution of the recurrence relation $a_n = A7^n + B(-1)^n$, and solve the system of equations determined by the boundary values $a_3 = 344$ and $a_4 = 2400$ to get the particular solution:

$$a_3 = A7^3 + B(-1)^3$$
$$a_4 = A7^4 + B(-1)^4$$

Now, substituting 344 and 2400 for a_3 and a_4 gives

$$344 = 343A - B$$
$$2400 = 2401A + B$$

Adding the two equations gives

$$2744 = 2744A$$
$$1 = A$$

It follows that $B = -1$. Therefore, $a_n = 7^n + (-1)^{n+1}$ for $n \geq 3$ is the particular solution. ∎

Substitution Method

- Guess the form of solution and use induction to find constants
- Determine upper bound on the recurrence

$$T_n = 2T_{\lfloor \frac{n}{2} \rfloor} + n$$

Guess the solution as: $T_n = O(n \lg n)$
Now, prove that $T_n \leq cn \lg n$ for some $c > 0$
Assume that the bound holds for $\lfloor \frac{n}{2} \rfloor$
Substituting into the recurrence

$$\begin{aligned} T_n &\leq 2(c \left\lfloor \frac{n}{2} \right\rfloor \lg(\left\lfloor \frac{n}{2} \right\rfloor)) + n \\ &\leq cn \lg\left(\frac{n}{2}\right) + n \\ &= cn \lg n - cn \lg 2 + n \\ &= cn \lg n - cn + n \\ &\leq cn \lg n \quad \forall c \geq 1 \end{aligned}$$

Boundary condition: Let the only bound be $T_1 = 1$

$$\nexists c \mid T_1 \leq c1 \lg 1 = 0$$

Problem overcome by the fact that asymptotic notation requires us to prove

$$T_n \leq cn \lg n \text{ for } n \geq n_0$$

Include T_2 and T_3 as boundary conditions for the proof

$$T_2 = 4 \quad T_3 = 5$$

Choose c such that $T_2 \leq c2 \lg 2$ and $T_3 \leq c3 \lg 3$
True for any $c \geq 2$

– If a recurrence is similar to a known recurrence, it is reasonable to guess a similar solution

$$T_n = 2T_{\lfloor \frac{n}{2} \rfloor} + n$$

If n is large, difference between $T_{\lfloor \frac{n}{2} \rfloor}$ and $T_{\lfloor \frac{n}{2} \rfloor + 17}$ is relatively small

– Prove upper and lower bounds on a recurrence and reduce the range of uncertainty.
Start with a lower bound of $T_n = \Omega(n)$ and an initial upper bound of $T_n = O(n^2)$. Gradually lower the upp bound and raise the lower bound to get asymptotically tight solution of $T_n = \Theta(n \lg n)$

- Recursion trees
  - Recurrence

$$T_n = 2T_{\frac{n}{2}} + n^2$$

Assume n to be an exact power of 2.

$$\begin{aligned}
T_n &= n^2 + 2T_{\frac{n}{2}} \\
&= n^2 + 2\left(\left(\frac{n}{2}\right)^2 + 2T_{\frac{n}{4}}\right) \\
&= n^2 + \frac{n^2}{2} + 4\left(\left(\frac{n}{4}\right)^2 + 2T_{\frac{n}{8}}\right) \\
&= n^2 + \frac{n^2}{2} + \frac{n^2}{4} + 8\left(\left(\frac{n}{8}\right)^2 + 2T_{\frac{n}{16}}\right) \\
&= n^2 + \frac{n^2}{2} + \frac{n^2}{4} + \frac{n^2}{8} + \cdots \\
&= n^2\left(1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \cdots\right) \\
&= \Theta(n^2)
\end{aligned}$$

The values above decrease geometrically by a constant factor.

– Recurrence

$$T_n = T_{\frac{n}{3}} + T_{\frac{2n}{3}} + n$$

Longest path from root to a leaf

$$n \to \left(\frac{2}{3}\right) n \to \left(\frac{2}{3}\right)^2 n \to \cdots 1$$

- Using the master method
  - Recurrence
  $$T_n = 9T_{\frac{n}{3}} + n$$
  $a = 9,\ b = 3,\ f(n) = n$
  $n^{\log_b a} = n^{\log_3 9} = \Theta(n^2)$
  $f(n) = O(n^{\log_3 9-\epsilon})$, where $\epsilon = 1$
  Apply case 1 of master theorem and conclude $T_n = \Theta(n^2)$
  - Recurrence
  $$T_n = T_{\frac{2n}{3}} + 1$$
  $a = 1,\ b = \frac{3}{2},\ f(n) = 1$
  $n^{\log_b a} = n^{\log_{\frac{3}{2}} 1} = n^0 = 1$
  $f(n) = \Theta(n^{\log_b a}) = \Theta(1)$
  Apply case 2 of master theorem and conclude $T_n = \Theta(\lg n)$
  - Recurrence
  $$T_n = 3T_{\frac{n}{4}} + n\lg n$$
  $a = 3,\ b = 4,\ f(n) = n\lg n$
  $n^{\log_b a} = n^{\log_4 3} = O(n^{0.793})$
  $f(n) = \Omega(n^{\log_4 3+\epsilon})$, where $\epsilon \approx 0.2$
  Apply case 3, if regularity condition holds for $f(n)$
  For large n, $af(\frac{n}{b}) = 3\frac{n}{4}\lg(\frac{n}{4}) \le \frac{3}{4}n\lg n = cf(n)$ for $c = \frac{3}{4}$
  Therefore, $T_n = \Theta(n\lg n)$

13

Recurrence Relations

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}$$

Ex:

If the seq $a_n = 3.2^n$, $n \geq 1$ then find the corresponding recurrence relation.

Soln:

For $n \geq 1$

$a_n = 3.2^n$.

$a_{n-1} = 3.2^{n-1}$

$= 3 \cdot \frac{2^n}{2}$

$a_{n-1} = \frac{a_n}{2}$

$a_n = 2(a_{n-1})$, for $n \geq 1$ with $a_0 = 3$

Ex:

Solve the recurrence relation defined by $S_0 = 100$ and $S_k = (1.08) S_{k-1}$ for $k \geq 1$

Soln:

Given $S_0 = 100$

$S_k = (1.08) S_{k-1}$, $k \geq 1$

$$16(\alpha_1+2\alpha_2) = 16$$

$$\alpha_1+2\alpha_2 = 1 \longrightarrow (1)$$

$$S_3 = 80.$$

$$\Rightarrow (\alpha_1+3\alpha_2)4^3 = 80$$

$$\Rightarrow 64(\alpha_1+3\alpha_2) = 80.$$

$$\Rightarrow \alpha_1+3\alpha_2 = \frac{80}{64} = \frac{5}{4}$$

$$\alpha_1+3\alpha_2 = \frac{5}{4} \longrightarrow (2)$$

$$(2)-(1) \Rightarrow \boxed{\alpha_2 = \frac{1}{4}}$$

$$\Rightarrow \alpha_1 + 2\left(\frac{1}{4}\right) = 1$$

$$\alpha_1 + \frac{1}{2} = 1$$

$$\boxed{\alpha_1 = \frac{1}{2}}$$

$$S(k) = (\alpha_1+\alpha_2 k)4^k$$

$$= (2+k)4^{k-1}$$

which is the reqd soln

POSSIBLE QUESTIONS

TWO MARKS

1. Define characteristics equation.
2. Solve $a_n - 4a_{n-1} = 0$ for n ≥ 2 with $a_0 = 1, a_1 = 1$.
3. State Fibonacci sequence.
4. Write the methods for solving recurrence.
5. If the sequence $a_n = 3.2^n, n \geq 1$ then find the corresponding recurrence relation.

SIX MARKS

1. Solve the recurrence relation $a_n = a_{n-1}+2a_{n-2}$ with $a_0=2$ and $a_1=7$.
2. Solve the Fibonacci recurrence $a_n = a_{n-1} + a_{n-2}$ with the initial condition $a_0 = a_1 = 1$.
3. Solve the recurrence relation $a_n+2-6a_{n+1}+9a_n=0$ with $a_0=1$ & $a_1 = 4$.
4. Solve the Recurrence Relation $a_n =6a_{n-1} -9a_{n-2}$ with $a_0=1$ and $a_1=6$.
5. Find the recurrence relation which satisfies $y_n=A\ 3^n+B(-4)^n$
6. Identify the sequence having the expression $\frac{5+2x}{1-4x^2}$ as a generating function.
7. Solve the recurrence relation $a_n - 7a_{n-1} + 10a_{n-2} = 0$ for $n \geq 2$ given that $a_0 = 10$, $a_1 = 41$ using generating function.
8. Solve the recurrence relation $a_{n+2}-a_{n+1}-6a_n=0$ given $a_0=2$ and $a_1=1$ using generating functions.
9. Using the generating function, solve the recurrence relation $a_n =3a_{n-1}$ for $n\geq1$ with $a_0=2$.
10. Using generating function, solve the recurrence relation $a_n =3a_{n-1} +1$ for $n\geq 1$ with $a_0=1$.

UNIT – IV

Graph Theory : Basic terminology, models and types, multigraphs and weighted graphs, graph representation, graph isomorphism, connectivity, Euler and Hamiltonian Paths and circuits, Planar graphs, graph coloring, trees, basic terminology and properties of trees, introduction to Spanning trees

INTRODUCTION : GRAPH THEORY

Graph theory is used to analyses problems of combinatorial nature that arise in computer science, operations research , physical science and economics . The term graph is familiar to you because it has been used in the context of straight lines and linear in equalities .In this chapter , first we will combine the concepts of graph theory with digraph of a relation to define a more general type of graph that has more than one edge between a pair of vertices. Second , we will identify basic components of a graph ,its features any many applications of graphs.

Definitions and Examples

Definition: A **graph** $G = (V,E)$ is a mathematical structure consisting of two finite sets V and E. The elements of V are called **Vertices (or nodes)** and the elements of E are called Edges. Each edge
is associated with a set consisting of **either one** or **two vertices** called its **endpoints**.

The correspondence from edges to endpoints is called **edge-endpoint function**. This function is generally denoted by γ. Due to this function, some author denote graph by $G = (V, E, \gamma)$.

Definition: A graph consisting of one vertex and no edges is called a **trivial graph**.

Definition: A graph whose vertex and edge sets are empty is called a **null graph**.

Definition: An edge with just one end point is called a **loop** or a **self loop**.
Thus, a loop is an edge that joins a single endpoint to itself.

Definition: An edge that is not a self-loop is called a **proper edge**.

Definition: If two or more edges of a graph G have the same vertices, then these edges are said to be **parallel** or **multi-edges**.

Definition: Two vertices that are connected by an edge are called **adjacent**.

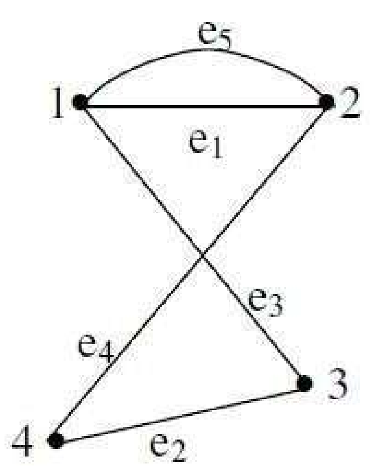
Definition: An endpoint of a loop is said to be **adjacent to itself**.

Definition: An edge is said to be **incident** on each of its endpoints.

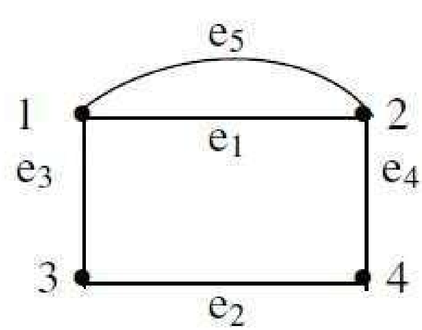
Definition: Two edges incident on the same endpoint are called **adjacent edges**.

Definition: The number of edges in a graph G which are incident on a vertex is called the degree of that **vertex**.

Definition: A vertex of degree zero is called an **isolated vertex**.
Thus, a vertex on which no edges are incident is called isolated.
Definition: A graph without multiple edges **(parallel edges)** and loops is called **Simple graph**.
Notation: In pictorial representations of a graph, the vertices will be denoted by dots and edges by line segments.

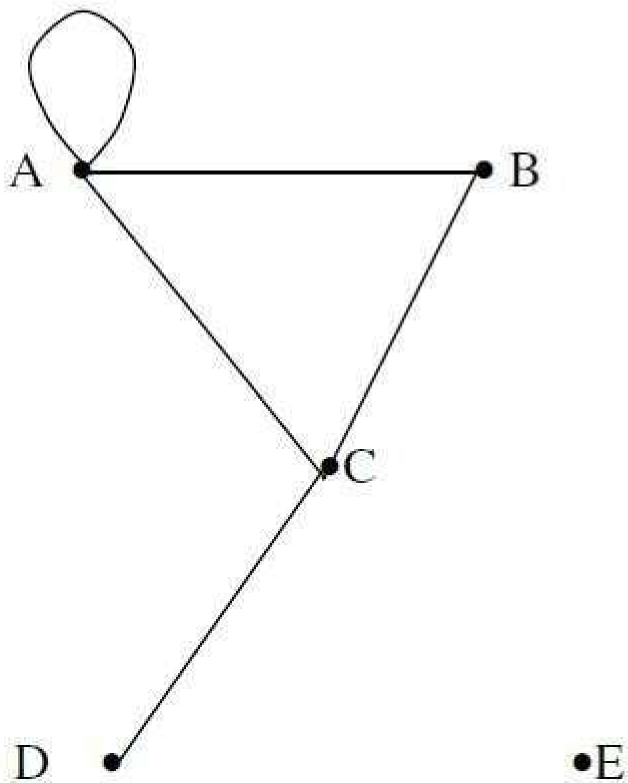


or



The edges e_2 and e_3 are adjacent edges because they are incident on the same vertex B.

2. Consider the graph with the vertices A, B , C, D and E pictured in the figure below.



In this graph, we note that

No. of edges = 5

Degree of vertex A = 4

Degree of vertex B = 2

Degree of vertex C = 3

Degree of vertex D = 1

Degree of vertex E = 0

Sum of the degree of vertices = 4 + 2 + 3 + 1 + 0 =10
Thus, we observe that

$$\sum_{i=1}^{5} \deg(v_i) = 2e \quad ,$$

where $\deg(v_i)$ denotes the degree of vertex v_i and e denotes the number of edges.

Euler's Theorem: (The First Theorem of Graph Theory): The sum of the degrees of the vertices of a graph G is equal to twice the number of edges in G.

(Thus, total degree of a graph is even)

Proof: Each edge in a graph contributes a count of 1 to the degree of two vertices (end points of
the edge), That is, each edge contributes 2 to the degree sum. Therefore the sum of degrees of the
vertices is equal to twice the number of edges.

Corollary: There must be an even number of vertices of odd degree in a given graph G.
Proof: We know, by the Fundamental Theorem, that

$$\sum_{i=1}^{n} \deg(v_i) = 2 \times \text{no. of edges}$$

Thus the right hand side is an even number. Hence to make the left-hand side an even number there
can be only even number of vertices of odd degree.

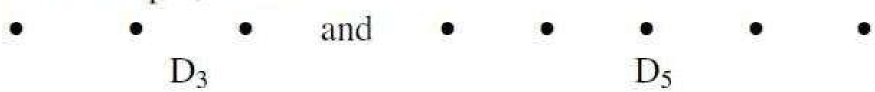
Theorem: A non-trivial simple graph G must have at least one pair of vertices whose degrees are
equal.

Proof: Let the graph G has n vertices. Then there appear to be n possible degree values, namely 0, 1, ….,n − 1. But there cannot be both a vertex of degree 0 and a vertex of degree n − 1 because if there is a vertex of degree 0 then each of the remaining n − 1 vertices is adjacent to atmost n−2 other vertices. Hence the n vertices of G can realize atmost $n-1$ possible values for their degrees. Hence the pigeonhole principle implies that at least two of the vertices have equal degree.

Definition: A graph G is said to **simple** if it has no parallel edges or loops. In a simple graph, an edge with endpoints v and w is denoted by $\{v, w\}$.

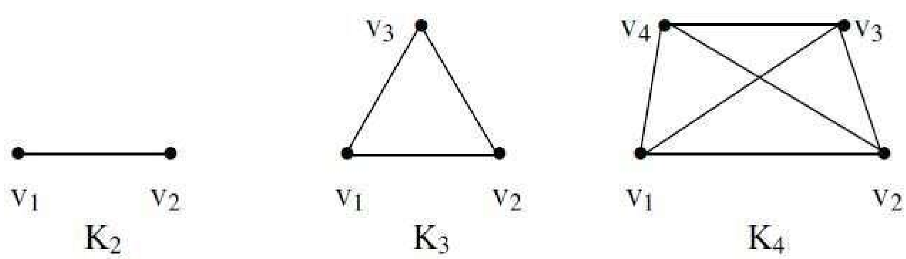
Definition: For each integer $n \geq 1$, let D_n denote the graph with **n vertices** and **no edges**. Then D_n is called the **discrete graph on n vertices**.

For example, we have



Definition: Let $n \geq 1$ be an integer. Then a simple graph with n vertices in which there is an edge between each pair of distinct vertices is called the **complete Graph** on n vertices. It is denoted by K_n.

For example, the complete graphs K_2, K_3 and K_4 are shown in the figures below:



Definition: If each vertex of a graph G has the same degree as every other vertex, then G is called a **regular graph**.

A **k-regular graph** is a regular graph whose common degree is k.

But this graph is not complete because v_2 and v_4 have not been connected through an edge. Similarly, v_1 and v_3 are not connected by any edge.

Thus

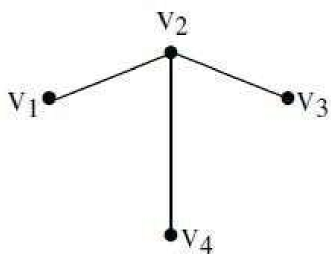
A Complete graph is always regular but **a regular graph need not be complete**.

Definition: If G is a simple graph, the **complement of G**, (**Edge complement**), denoted by G′ or G^c is a graph such that

(i) The vertex set of G′ is identical to the vertex set of G, that is $V_{G'} = V_G$

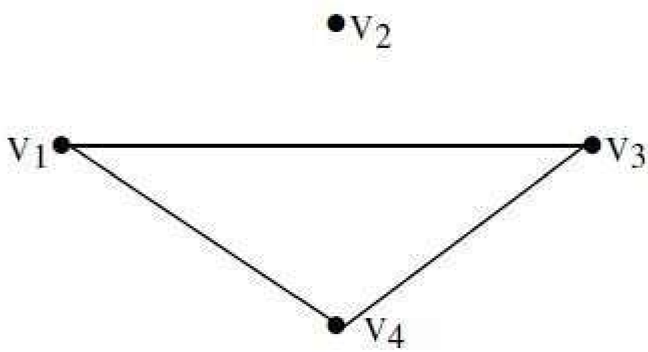
(ii) Two distinct vertices v and w of G′ **are connected by** an edge if and only if v and w **are not connected** by an edge in G.

For example, consider the graph G



G

Then complement G′ of G is the graph



G′

Definition: The property of mapping endpoints to endpoints is called **preserving incidence** or **the continuity rule** for graph mappings.
As a consequence of this property, a self-loop must map to a self-loop.
Thus, two isomorphic graphs are same except for the labeling of their vertices and edges.

Walks, Paths and Circuits

Definition: In a graph G, **a walk from vertex v_0 to vertex v_n** is a finite alternating sequence:

$$\{v_0, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v_n\}$$

of vertices and edges such that v_{i-1} and v_i are the endpoints of e_i.

The **trivial walk** from a vertex v to v consists of the single vertex v.

Definition: In a graph G, a **path** from the vertex v_0 to the vertex v_n is a walk from v_0 to v_n that does not contain a repeated edge.

Thus a **path** from v_0 to v_n is a walk of the form

$$\{v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n\},$$

where all the edges e_I are distinct.

Definition: In a graph, a simple path from v_0 to v_n is a path that does not contain a repeated vertex.

Thus a simple path is a walk of the form

$$\{v_0, e_1, v_1, e_2, v_2, \ldots, v_{i-1}, e_n, v_n\},$$

where all the e_i are distinct and all the v_i are distinct.

Definition: A walk in a graph G that starts and ends at the same vertex is called a **closed walk**.

Definition: A closed walk that does not contain a repeated edge is called a **circuit**.

Thus, closed a closed path is called a circuit (or a cycle) and so a circuit is a walk of the form

$$\{v_0, e_1, v_1, e_2, v_2, \ldots, v_{n-1}, e_n, v_n\},$$

where $v_0 = v_n$ and all the e_i are distinct.

Definition: In a graph the number of edges in the path $\{v_0, e_1, v_1, e_2, \ldots\ldots, e_n, v_n\}$ from v_0 to v_n is called the **length of the path**.

Theorem: If there is a path from vertex v_1 to v_2 in a graph with n vertices, then there does not exist a path of more than n-1 edges from vertex v_1 to v_2.

Proof: Suppose there is a path from v_1 to v_2. Let

$$v_1, \ldots\ldots, v_i, \ldots\ldots\ldots, v_2$$

be the sequence of vertices which the path meets between the vertices v_1 and v_2. Let there be m edges in the path. Then there will be m + 1 vertices in the sequence. Therefore if $m > n-1$, then there will be more than n vertices in the sequence. But the graph is with n vertices. Therefore some vertex, say v_k, appears more than once in the sequence. So the sequence of vertices shall be

$$v_1, \ldots\ldots, v_i, \ldots\ldots, v_k, \ldots\ldots, v_k, \ldots\ldots, v_2.$$

Deleting the edges in the path that lead v_k back to v_k we have a path from v_1 to v_2 that has less edges than the original one. This argument is repeated untill we get a path that has n-1 or less edges.

CONNECTED AND DISCONNECTED GRAPHS :

Definition: Two vertices v_1 and v_2 of a graph G are said to be **connected** if and only if there is a walk from v_1 to v_2.

Definition: A graph G is said to be **connected** if and only if given any two vertices v_1 and v_2 in G, there is a walk from v_1 to v_2.

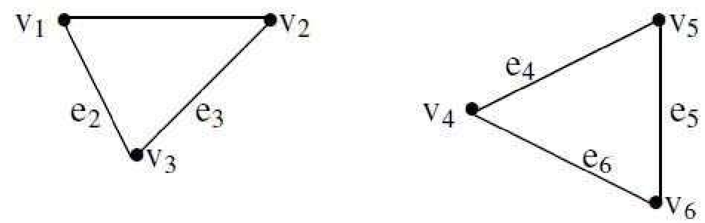
Thus, a graph G is connected if there exists a walk between every two vertices in the graph.

Definition: A graph which is not connected is called **Disconnected Graph.**

Example: Which of the graph below are connected?

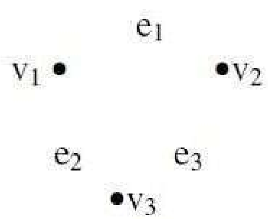
Definition: If a graph G is disconnected, then the various connected pieces of G are called the **connected components of the graph**.

Example: Consider the graph given below:



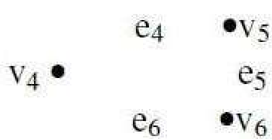
This graph is disconnected and have two connected components:

H_1 :



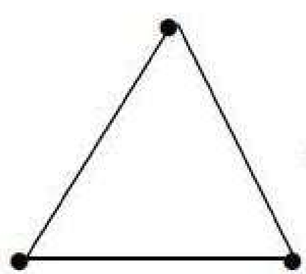
with vertex set $\{v_1, v_2, v_3\}$ and edge set $\{e_1, e_2, e_3\}$

H_2 :



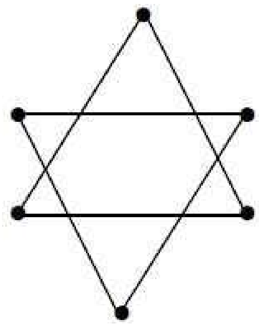
with vertex set $\{v_4, v_5, v_6\}$ and edge set $\{e_4, e_5, e_6\}$.

Solution: The connected components are :



and

Example: Find the number of connected components in the graph



Eulerian Paths And Circuits

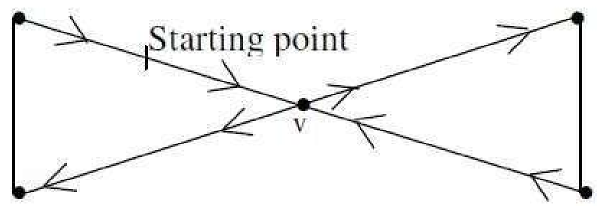
Definition: A path in a graph G is called an **Euler Path** if it includes **every edge exactly once**.

Definition: A graph is called **Eulerian graph** if there exists a Euler circuit for that graph.

Definition: A circuit in a graph G is called an **Euler Circuit** if it includes every edge exactly once. Thus, an Euler circuit (Eulerian trail) for a graph G is a sequence of adjacent vertices and edges in G that starts and ends at the same vertex, uses every vertex of G at least once, and uses **every edge of G exactly once**.

Theorem 1. If a graph has an Euler circuit, then every vertex of the graph has even degree.

Proof: Let G be a graph which has an Euler circuit. Let v be a vertex of G. We shall show that degree of v is even. By definition, Euler circuit contains every edge of graph G. Therefore the Euler circuit contains all edges incident on v. We start a journey beginning in the middle of one of the edges adjacent to the start of Euler circuit and continue around the Euler circuit to end in the middle of the starting edge. Since Euler circuit uses every edge exactly once, the edges incident on v occur



in entry / exist pair and hence the degree of v is a multiple of 2. Therefore the degree of v is even. This completes the proof of the theorem.

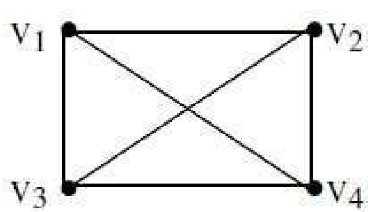
We know that contrapositive of a conditional statement is logically equivalent to statement. Thus the above theorem is equivalent to:

Theorem:2. If a vertex of a graph is not of even degree, then it does not have an Euler circuit.

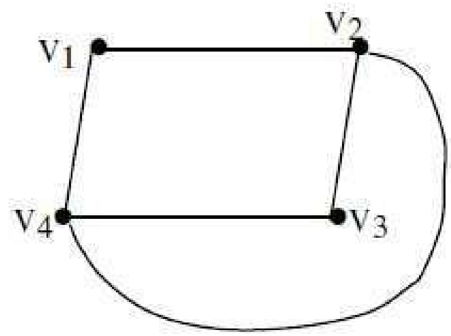
or

"If some vertex of a graph has odd degree, then that graph does not have an Euler circuit".

Example: Show that the graphs below do not have Euler circuits.
(a)



(b)



Solution: In graph (a), degree of each vertex is 3. Hence this **does not** have a Euler circuit.

In graph (b), we have

$$\deg(v_2) = 3$$

$$\deg(v_4) = 3$$

Since there are vertices of odd degree in the given graph, therefore it **does not** have an Euler circuit.

are graphs in which each vertex has degree 2 but these graphs do not have Euler circuits since there is no path which uses each vertex at least once.

Theorem 3. If G is a connected graph and every vertex of G has even degree, then G has an Euler circuit.

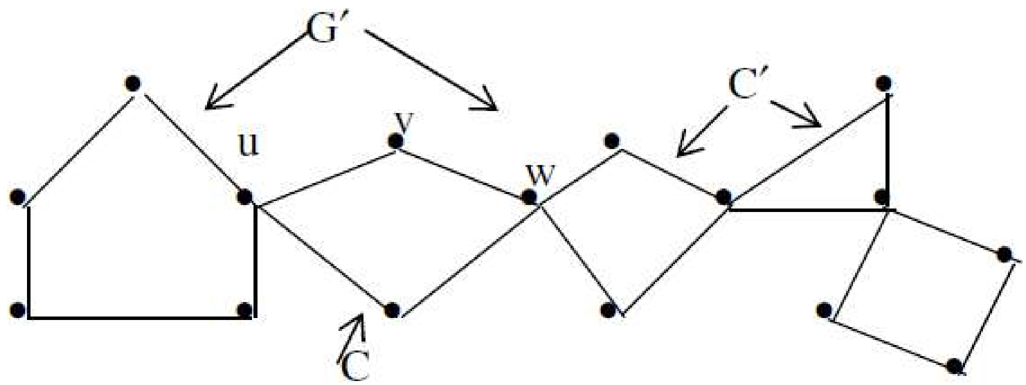
Proof: Let every vertex of a connected graph G has even degree. If G consists of a single vertex, the trivial walk from v to v is an Euler circuit. So suppose G consists of more than one vertices. We start from any verted v of G. Since the degree of each vertex of G is even, if we reach each vertex other than v by travelling on one edge, the same vertex can be reached by travelling on another previously unused edge. Thus a sequence of distinct adjacent edges can be produced indefinitely as long as v is not reached. Since number of edges of the graph is finite (by definition of graph), the sequence of distinct edges will terminate. Thus the sequence must return to the starting vertex. We thus obtain a sequence of adjacent vertices and edges starting and ending at v without repeating any edge. Thus we get a circuit C.

If C contains every edge and vertex of G, then C is an Eular circuit.

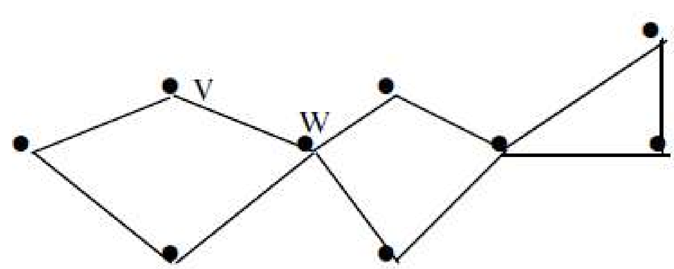
If C does not contain every edge and vertex of G, remove all edges of C from G and also any vertices that become isolated when the edges of C are removed. Let the resulting subgraph be G′. We note that when we removed edges of C, an even number of edges from each vertex have been removed. Thus degree of each remaining vertex remains even.

Further since G is connected, there must be at least one vertex common to both C and G′. Let it be w(in fact there are two such vertices). Pick any sequence of adjacent vertices and edges of G′ starting and ending at w without repeating an edge. Let the resulting circuit be C′.

Join C and C′ together to create a new circuit C″. Now, we observe that if we start from v and follow C all the way to reach w and then follow C′ all the way to reach back to w. Then continuing travelling along the untravelled edges of C, we reach v.



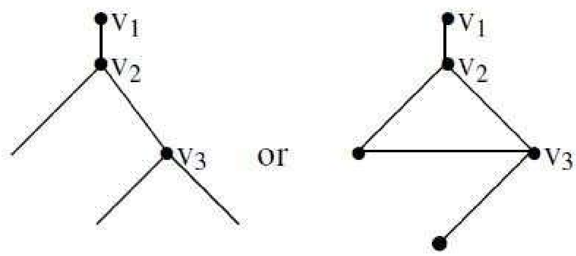
G



C″

Theorem 5. If a graph G has more than two vertices of odd degree, then there can be no Euler path in G.

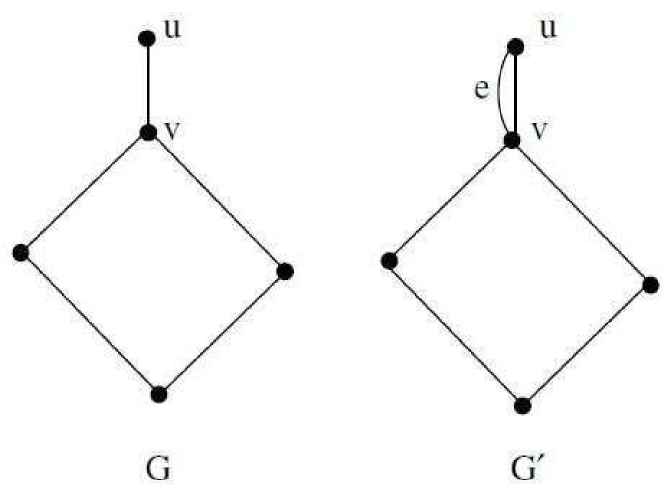
Proof : Let v_1, v_2 and v_3 be vertices of odd degree. Since each of these vertices had odd degree, any possible Euler path must leave (arrive at) each of v_1, v_2, v_3 with no way to return (or leave). One vertex of these three vertices may be the beginning of Euler path and another the end but this leaves the third vertex at one end of an untravelled edge. Thus there is no Euler path.



(Graphs having more than two vertices of odd degree).

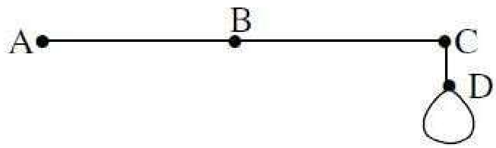
Theorem 6. If G is a connected graph and has exactly two vertices of odd degree, then there is an Euler path in G. Further, any Euler path in G must begin at one vertex of odd degree and end at the other.

Proof: Let u and v be two vertices of odd degree in the given connected graph G.



If we add the edge e to G, we get a connected graph G′ all of whose vertices have even degree. Hence there will be an Euler circuit in G′. If we omit e from Euler circuit, we get an Euler path beginning at u(or v) and edning at v(or u).

Examples. Has the graph given below an Eulerian path?



Solution: In the given graph,

$$\deg(A) = 1$$

$$\deg(B) = 2$$

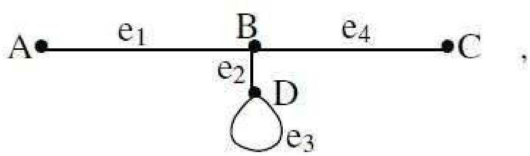
$$\deg(C) = 2$$

$$\deg(D) = 3$$

Thus the given connected graph has exactly two vertices of odd degree. Hence, it has an Eulerian path.

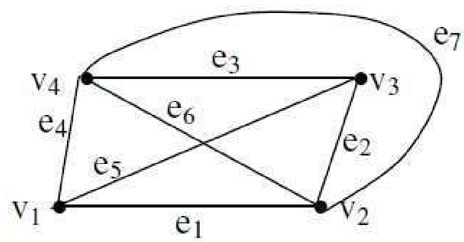
If it starts from A(vertex of odd degree), then it ends at D(vertex of odd degree). If it starts from D(vertex of odd degree), then it ends at A(vertex of odd degree).

But on the other hand if we have the graph as given below :



then deg(A) = 1, deg(B) = 3 deg(C) = 1, degree of D = 3 and so we have four vertices of odd degree. Hence it does not have Euler path.

Example: Does the graph given below possess an Euler circuit?



Solution: The given graph is connected. Further

$$\deg(v_1) = 3$$

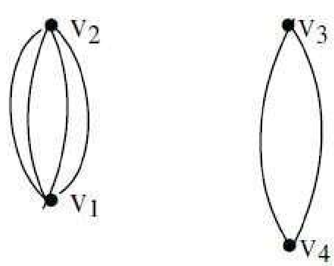
$$\deg(v_2) = 4$$

$$\deg(v_3) = 3$$

$$\deg(v_4) = 4$$

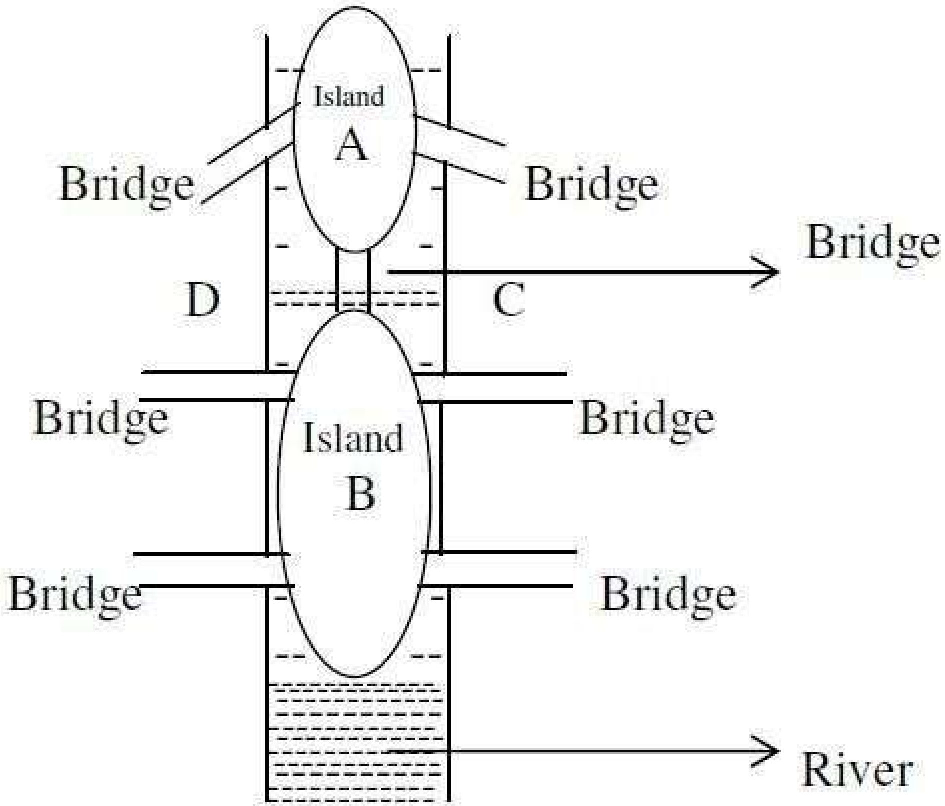
Since this connected graph has vertices with odd degree, it cannot have Euler circuit. But this graph has Euler path, since it has exactly two vertices of odd degree. For example, $v_3\, e_2\, v_2\, e_7\, v_4\, e_6\, v_2\, e_1\, v_1\, e_4\, v_4\, e_3\, v_3\, e_5\, v_1$

Example: Consider the graph

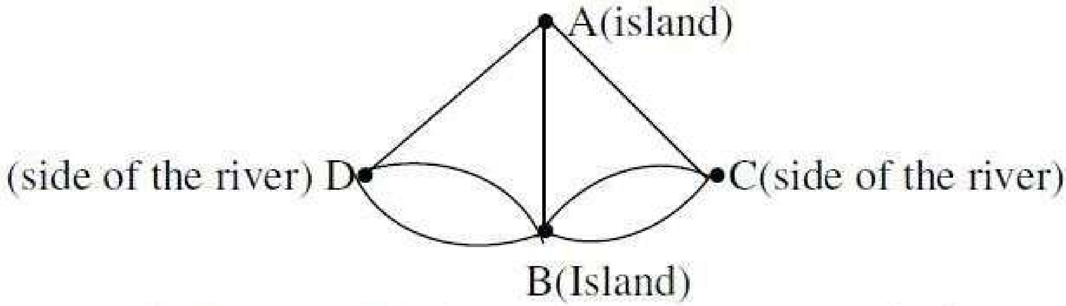


Here, $\deg(v_1) = 4$, $\deg(v_2) = 4$, $\deg(v_3) = 2$, $\deg(v_4) = 2$. Thus degree of each vertex is even. But the graph is not Eulerian since it is **not connected**.

Example 4:. The bridges of Konigsberg: The graph Theory began in 1736 when Leonhard Euler solved the problem of seven bridges on Pregel river in the town of Konigsberg in Prussia (now Kaliningrad in Russia). The two islands and seven bridges are shown below:



Thus the graph of the problem is



(Euler's graphical representation of seven bridge problem)

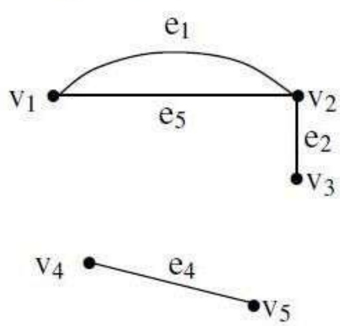
The problem then reduces to

"Is there any Euler's path in the above diagram?".

To find the answer, we note that there are more than two vertices having odd degree. Hence there exist no Euler path for this graph.

Definition: An edge in a connected graph is called a **Bridge** or a **Cut Edge** if deleting that edge creates a disconnected graph.

In this graph, if we remove the edge e_3, then the graph breaks into two Connected Component given below:

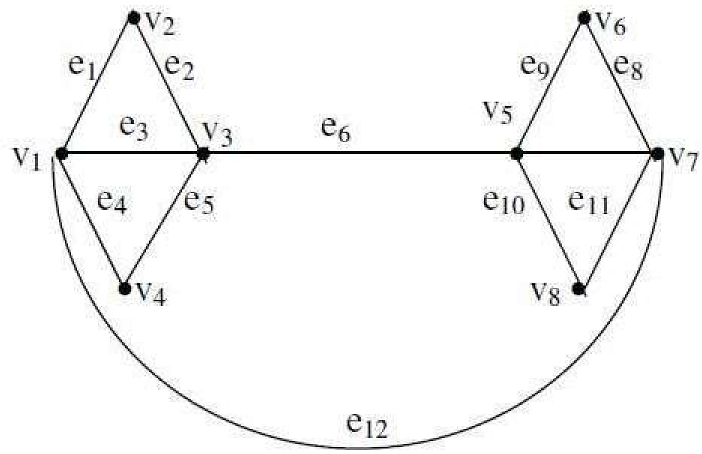


Hence the edge e_3 is a bridge in the given graph.

METHOD FOR FINDING EULER CIRCUIT

We know that if every vertex of a non empty connected graph has even degree, then the graph has an Euler circuit. We shall make use of this result to find an Euler path in a given graph.

Consider the graph



We note that

$$\deg(v_2) = \deg(v_4) = \deg(v_6) = \deg(v_8) = 2$$

$$\deg(v_1) = \deg(v_3) = \deg(v_5) = \deg(v_7) = 4$$

Hence all vertices have even degree. Also the given graph is connected. Hence the given has an Euler circuit. We start from the vertex v_1 and let C be

$$C : v_1\ v_2\ v_3\ v_1$$

Then C is not an Euler circuit for the given graph but C intersect the rest of the graph at v_1 and v_3. Let C' be

$$C' : v_1 v_4\ v_3\ v_5\ v_7\ v_6\ v_5\ v_8\ v_7\ v_1$$

(In case we start from v_3, then C' will be $v_3\ v_4\ v_1\ v_7\ v_6\ v_5\ v_7\ v_8\ v_5$)
Path C' into C and obtain

$$C'' : v_1 v_2\ v_3\ v_1\ v_4\ v_3\ v_5\ v_7\ v_6\ v_5\ v_8\ v_7\ v_1$$

Or we can write

$$C'' : e_1 e_2\ e_3\ e_4\ e_5\ e_6\ e_7\ e_8\ e_9\ e_{10} e_{11}\ e_{12}$$

(If we had started from v_2, then C'' : $v_1 v_2\ v_3\ v_4\ v_1\ v_7\ v_6\ v_5\ v_7\ v_8\ v_5\ v_3\ v_1$ **or**

$e_1 e_2\ e_5\ e_4\ e_{12}\ e_8\ e_9\ e_7\ e_{11}\ e_{10} e_6\ e_3$)

In C'' all edges are covered exactly once. Also every vertex has been covered at least once. Hence C'' is a Euler circuit.

Hamiltonian Circuits

Definition: A **Hamiltonian Path** for a graph G is a sequence of adjacent vertices and distinct edges in which every vertex of G appears exactly once.

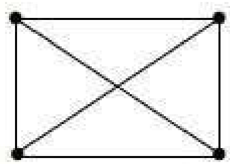
Definition: A **Hamiltonian Circuit** for a graph G is a sequence of adjacent vertices and distinct edges in which every vertex of G appears exactly once, except for the first and the last which are the same.

Definition: A graph is called **Hamiltonian** if it admits a Hamiltonian circuit.

Example 1 : A complete graph K_n has a Hamiltonian Circuit. In particular the graphs

K_3

and



K_4

are Hamiltonian.

Theorem: Let G be a connected graph with n vertices and let u and v be two vertices of G that are not adjacent. If

$$\deg(u) + \deg(v) \geq n,$$

then G has a Hamiltonian circuit.

Matrix Representation of Graphs

A graph can be represented inside a computer by using the adjacency matrix or the incidence matrix of the graph.

Definition: Let G be a graph with n ordered vertices $v_1, v_2, \ldots\ldots, v_n$. Then the **adjacency matrix of G** is the $n \times n$ matrix $A(G) = (a_{ij})$ over the set of non-negative integers such that

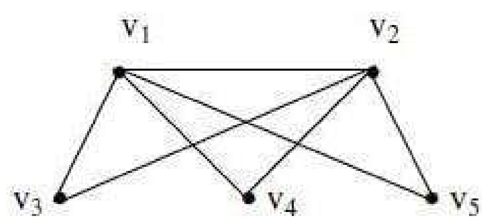
a_{ij} = the number of edges connecting v_i and v_j for all i, j = 1, 2,…,n.

We note that if G has no loop, then there is no edge joining v_i to v_i, i = 1, 2,…,n. Therefore, in this case, all the entries on the main diagonal will be 0.

Further, if G has no parallel edge, then the entries of A(G) are either 0 or 1.
It may be noted that adjacent matrix of a graph is symmetric.

Conversely, given a $n \times n$ symmetric matrix $A(G) = (a_{ij})$ over the set of non-negative integers, we can associate with it a graph G, whose adjacency matrix is A(G), by letting G have n vertices and joining v_i to vertex v_j by a_{ij} edges.

Example 1: Find the adjacency matrix of the graph shown below:



Solution: The adjacency matrix $A(G) = (a_{ij})$ is the matrix such that

a_{iJ} = No. of edges connecting v_i and v_j.

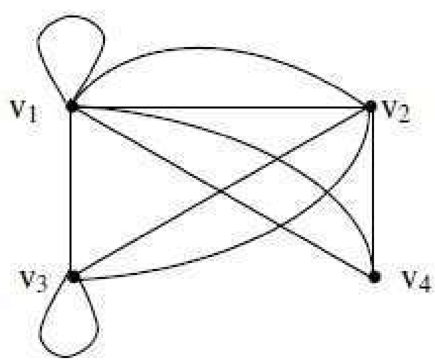
So we have for the given graph

$$A(G) = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Example 2 : Find the graph that have the following adjacency matrix

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

Solution: We note that there is a loop at v_1 and a loop at v_3. There are parallel edges between v_1, v_2; v_1, v_4; v_2, v_1; v_2, v_3, v_3, v_2; v_4, v_1. Thus the graph is



Trees

Definition: A graph is said to be a **Tree** if it is a connected acyclic graph.

THEOREM:

A graph G with $e = v - 1$, that has no circuit is a tree.

Proof: It is sufficient to show that G is connected. Suppose G is not connected and let G′, G″….. be connected component of G. Since each of G′, G″,…. is connected and has no cycle, they all are tree. Therefore, by Lemma 3,

$$e' = v' - 1$$

$$e'' = v'' - 1$$

-------------- ,

where e′, e″, … are the number of edges and v′, v″,… are the number of vertices in G′, G″, …respectively. We have, on adding

$$e' + e'' + \ldots\ldots = (v' - 1) + (v'' - 1) + \ldots\ldots$$

Since

$$e = e' + e'' + \ldots..$$
$$v = v' + v'' + \ldots. ,$$

we have

$$e < v - 1 ,$$

which contradicts our hypotheses. Hence G is connected. So G is connected and acyclic and is therefore a tree.

Definition: A directed tree is called a **rooted tree** if there is exactly one vertex whose incoming degree is 0 and the incoming degrees of all other vertices are 1.

Definition: In a rooted tree, a vertex, whose outgoing degree is 0 is called a **leaf** or **terminal node**, whereas a vertex whose outgoing degree is non - zero is called a **branch node** or an **internal node**.

Definition: Let u be a branch node in a rooted tree. Then a vertex v is said to be **child** (**son** or **offspring**) of u if there is an edge from u to v. In this case u is called **parent** (**father**) of v.

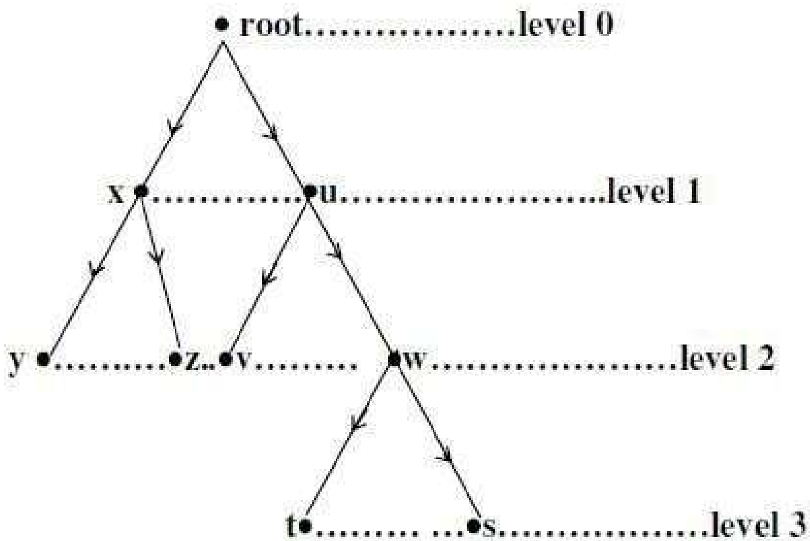
Definition: Two vertices in a rooted tree are said to be **siblings** (**brothers**) if they are both children of same parent.

Definition: A vertex v is said to be a **descendent** of a vertex u if there is a unique directed path from u to v.
In this case u is called the **ancestor** of v.

Definition: The **level** (or **path length**) of a vertex u in a rooted tree is the number of edges along the unique path between u and the root.

Definition: The **height** of a rooted tree is the maximum level to any vertex of the tree.

As an example of these terms consider the rooted tree shown below:



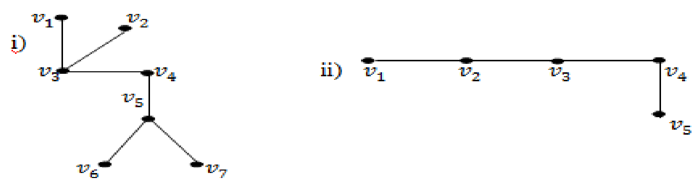
POSSIBLE QUESTIONS

PART-B (TWO MARKS)

1. Define directed graph.
2. How many vertices does a regular graph of degree 4 with 10 edges have
3. Define Hamiltonian path
4. Define isomorphic graph.
5. Define chromatic number

PART – C(SIX MARKS)

1. State and prove handshaking lemma
2. Define (i) Proper coloring graph (ii) Chromatic Number (iii) Independent set.
3. Give an example of a graph which is
   (i). Eulerian but not Hamiltonian
   (ii). Hamiltonian but not Eulerian
   (iii). Both eulerian and Hamiltonian
   (iv). Non Eulerian and non Hamiltonian
4. Show that if a fully binary tree has i internal vertices then it has (i+1) terminal vertices and (2i+1) total vertices.
5. Describe about konigsberg bridge problem.
6. Find the eccentricity of all vertices, center, radius and diameter of the following graph.



7. Prove that the number of vertices of odd degree in a graph is always even.
8. Prove that the number of pendent vertices of a tree is equal to $\frac{n+1}{2}$
9. Define graph. Explain the various types of graph with an example.
10. In a undirected graph, the number of odd degree vertices are even.

UNIT – V

Prepositional Logic: Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory.

Propositions. Compound Statements. Truth Tables

Statements (Propositions): Sentences that claim certain things, either true or false

Notation: A, B, …P, Q, R, …., p, q, r, etc.

Examples of statements: Today is Monday. This book is expensive
If a number is smaller than 0 then it is positive.

Examples of sentences that **are not statements**: Close the door! What is the time?

Propositional variables: A, B, C, …, P., Q, R, … Stand for statements. May have true or false value.

Propositional constants:

T – true
F - false

Basic logical connectives: NOT, AND, OR
Other logical connectives can be represented by means of the basic connectives

Logical connectives	pronounced	Symbol in Logic
Negation	NOT	¬, ~, ‘
Conjunction	AND	Λ
Disjunction	OR	V
Conditional	if then	$\rightarrow$
Biconditional	if and only if	$\leftrightarrow$
Exclusive or	Exclusive or	$\oplus$

Truth tables - Define formally the meaning of the logical operators.

The abbreviation **iff** means **if and only if**

a. Negation (NOT, ~, ¬, ‘)

P	~P	
T	F	~P is true if and only if P is false
F	T	

b. Conjunction (AND, Λ, &&)

P	Q	P Λ Q	
T	T	T	P Λ Q is true iff both P and Q are true. In all other cases P Λ Q is false
T	F	F	
F	T	F	
F	F	F	

c. Disjunction / Inclusive OR (OR, V, ||)

P	Q	P V Q	
T	T	T	P V Q is true iff P is true or Q is true or both are true.
T	F	T	P V Q is false iff both P and Q are false
F	T	T	
F	F	F	

d. Conditional , known also as implication (→)

P	Q	P→ Q	
T	T	T	The implication P→ Q is false iff P is true however Q is false.
T	F	F	In all other cases the implication is true
F	T	T	
F	F	T	

e. Biconditional (↔)

P	Q	P ↔ Q	
T	T	T	P↔ Q is true iff P and Q have same values - both are true or both are false.
T	F	F	If P and Q have different values, the biconditional is false.
F	T	F	
F	F	T	

f. Exclusive OR (⊕)

P	Q	P ⊕ Q	
T	T	F	P⊕ Q is true iff P and Q have different values
T	F	T	We say: "P or Q but not both"
F	T	T	
F	F	F	

Precedence of the logical connectives:

Connectives within parentheses, innermost parentheses first

¬	**negation**
Λ	**conjunction**
V	**disjunction**
→	**conditional**
↔, ⊕	**biconditional, exclusive OR**

Compound Statements: Logical expressions that consist of propositional variables and logical connectives. They may contain also propositional constants.

Evaluating compound statements : by building their truth tables

Example: ¬ P V Q

P	Q	¬ P	¬ P V Q
T	T	F	T
T	F	F	F
F	T	T	T
F	F	T	T

(P V Q) Λ ¬ (P Λ Q)

P	Q	P V Q A	P Λ Q B	¬ (P Λ Q) ¬ B	(P V Q) Λ ¬ (P Λ Q) A Λ ¬B (the letters A and B are used as shortcuts)
T	T	T	T	F	F
T	F	T	F	T	T
F	T	T	F	T	T
F	F	F	F	T	F

1. Tautologies and Contradictions

A propositional expression is a **tautology** if and only if for all possible assignments of truth values to its variables its truth value is **T**

Example: $P \vee \neg P$ is a tautology

P	$\neg P$	$P \vee \neg P$
T	F	T
F	T	T

A propositional expression is a **contradiction** if and only if for all possible assignments of truth values to its variables its truth value is **F**

Example: $P \wedge \neg P$ is a contradiction

P	$\neg P$	$P \wedge \neg P$
T	F	F
F	T	F

Usage of tautologies and contradictions - in proving the validity of arguments; for rewriting expressions using only the basic connectives.

Definition: Two propositional expressions P and Q are logically equivalent, if and only if $P \leftrightarrow Q$ is a tautology. We write $P \equiv Q$ or $P \Leftrightarrow Q$.

Note that the symbols $\equiv$ and $\Leftrightarrow$ are **not logical connectives**

Exercise:

a) Show that $P \to Q \leftrightarrow \neg P \vee Q$ is a tautology, i.e. $P \to Q \equiv \neg P \vee Q$

P	Q	$\neg P$	$\neg P \vee Q$	$P \to Q$	$P \to Q \leftrightarrow \neg P \vee Q$
T	T	F	T	T	T
T	F	F	F	F	T
F	T	T	T	T	T
F	F	T	T	T	T

2. Logical equivalences

Similarly to standard algebra, there are **laws** to manipulate logical expressions, given as logical equivalences.

1. Commutative laws	$P \vee Q \equiv Q \vee P$	
	$P \wedge Q \equiv Q \wedge P$	
2. Associative laws	$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$	
	$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$	
3. Distributive laws:	$(P \vee Q) \wedge (P \vee R) \equiv P \vee (Q \wedge R)$	
	$(P \wedge Q) \vee (P \wedge R) \equiv P \wedge (Q \vee R)$	
4. Identity	$P \vee F \equiv P$	
	$P \wedge T \equiv P$	
5. Complement properties	$P \vee \neg P \equiv T$	(excluded middle)
	$P \wedge \neg P \equiv F$	(contradiction)
6. Double negation	$\neg(\neg P) \equiv P$	
7. Idempotency (consumption)	$P \vee P \equiv P$	
	$P \wedge P \equiv P$	
8. De Morgan's Laws	$\neg(P \vee Q) \equiv \neg P \wedge \neg Q$	
	$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$	
9. Universal bound laws (Domination)	$P \vee T \equiv T$	
	$P \wedge F \equiv F$	
10. Absorption Laws	$P \vee (P \wedge Q) \equiv P$	
	$P \wedge (P \vee Q) \equiv P$	
11. Negation of T and F:	$\neg T \equiv F$	
	$\neg F \equiv T$	

1. Truth table of the conditional statement

P	Q	P →Q
T	T	T
T	**F**	**F**
F	T	T
F	F	T

P is called **antecedent**
Q is called **consequent**
Meaning of the conditional statement: The truth of P implies (leads to) the truth of Q

Note that when P is false the conditional statement is true no matter what the value of Q is. We say that in this case the conditional statement is **true by default or vacuously true.**

2. Representing the implication by means of disjunction

$$\mathbf{P \to Q \equiv \neg P \vee Q}$$

P	Q	¬ P	**P→ Q**	**¬ P V Q**
T	T	F	T	T
T	F	F	**F**	**F**
F	T	T	T	T
F	F	T	T	T

Same truth tables

Usage:
1. To rewrite "OR" statements as conditional statements and vice versa (for better understanding)
2. To find the negation of a conditional statement using De Morgan's Laws

3. Rephrasing "or" sentences as "if-then" sentences and vice versa

Consider the sentence:
(1) "The book can be found in the library or in the bookstore".

Let
A = The book can be found in the library
B = The book can be found in the bookstore

Logical form of (1): **A V B**

Rewrite A V B **as a conditional statement**

In order to do this we need to use the commutative laws, the equivalence $\neg(\neg P) \equiv P$, and the equivalence $P \rightarrow Q \equiv \neg P \vee Q$

Thus we have:

$A \vee B \equiv \neg(\neg A) \vee B \equiv \neg A \rightarrow B$

The last expression $\neg A \rightarrow B$ is translated into English as
"If the book cannot be found in the library, it can be found in the bookstore".

Here the statement "The book cannot be found in the library" is represented by $\neg A$

There is still one more conditional statement to consider.
$A \vee B \equiv B \vee A$ (commutative laws)

Then, following the same pattern we have:

$B \vee A \equiv \neg(\neg B) \vee A \equiv \neg B \rightarrow A$

The English sentence is: **"If the book cannot be found in the bookstore, it can be found in the library.**

We have shown that:

$A \vee B \equiv \neg(\neg A) \vee B \equiv \neg A \rightarrow B$
$A \vee B \equiv B \vee A \equiv \neg(\neg B) \vee A \equiv \neg B \rightarrow A$

Thus the sentence **"The book can be found in the library or in the bookstore"** can be rephrased as:
"If the book cannot be found in the library, it can be found in the bookstore".
"If the book cannot be found in the bookstore, it can be found in the library.

4. Negation of conditional statements

Positive: The sun shines
Negative: The sun does not shine

Positive: " If the temperature is 250°F then the compound is boiling "
Negative: ?
In order to find the negation, we use De Morgan's Laws.

Let
P = the temperature is 250°F
Q = the compound is boiling

Positive: $P \rightarrow Q \equiv \neg P \vee Q$
Negative: $\neg (P \rightarrow Q) \equiv \neg(\neg P \vee Q) \equiv \neg(\neg P) \wedge \neg Q \equiv P \wedge \neg Q$

Negative: The temperature is 250°F however the compound is not boiling

IMPORTANT TO KNOW:

The negation of a disjunction is a conjunction.
The negation of a conjunction is a disjunction

The **negation of a conditional statement is a conjunction**, not another if-then statement

Question: Which logical connective when negated will result in a conditional statement?

5. Necessary and sufficient conditions

Definition:

"P is a **sufficient condition** for Q" means : **if P then Q, $P \rightarrow Q$**

"P is a **necessary condition** for Q" means: **if not P then not Q, $\sim P \rightarrow \sim Q$**

The statement **$\sim P \rightarrow \sim Q$ is equivalent to $Q \rightarrow P$**

Hence given the statement **$P \rightarrow Q$,**

P is a sufficient condition for Q, and Q is a necessary condition for P.

Examples:

If n is divisible by 6 then n is divisible by 2.
The sufficient condition to be divisible by 2 is to be divisible by 6.
The necessary condition to be divisible by 6 is to be divisible by 2

If n is odd then n is an integer.
The sufficient condition to be an integer to be odd.
The necessary condition to be odd is to be an integer.

If and only if - the biconditional

P	Q	$P \leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T

$P \leftrightarrow Q$ is true whenever P and Q have same values. Otherwise it is false.

This means that **both $P \to Q$ and $Q \to P$ have to be true**

P	Q	$P \to Q$	$Q \to P$	$P \leftrightarrow Q$
T	**T**	**T**	**T**	**T**
T	F	F	T	F
F	T	T	F	F
F	**F**	**T**	**T**	**T**

Contrapositive

Definition: The expression $\sim Q \to \sim P$ is called **contrapositive** of $P \to Q$

The conditional statement $P \to Q$ and its contrapositive $\sim Q \to \sim P$ **are equivalent**. The proof is done by comparing the truth tables

The truth table for $P \to Q$ and $\neg Q \to \neg P$ is:

P	Q	$\neg P$	$\neg Q$	$P \to Q$	$\neg Q \to \neg P$
T	T	F	F	T	T
T	F	F	T	F	F
F	T	T	F	T	T
F	F	T	T	T	T

We can also prove the equivalence by using the disjunctive representation:

$P \to Q \equiv \neg P \vee Q \equiv Q \vee \neg P \equiv \neg(\neg Q) \vee \neg P \equiv \neg Q \to \neg P$

Converse and inverse

Definition: The converse of $P \to Q$ is the expression $Q \to P$

Definition: The inverse of $P \to Q$ is the expression $\sim P \to \sim Q$

Neither the converse nor the inverse are equivalent to the original implication.
Compare the truth tables and you will see the difference.

P	Q	$\neg P$	$\neg Q$	$P \to Q$	$Q \to P$	$\neg P \to \neg Q$
T	T	F	F	T	T	T
T	F	F	T	**F**	**T**	**T**
F	T	T	F	**T**	**F**	**F**
F	F	T	T	T	T	T

Valid and Invalid Arguments.

Definition: An argument is a sequence of statements, ending in a conclusion. All the statements but the final one (the conclusion) are called premises(or assumptions, hypotheses)

Verbal form of an argument:

(1) If Socrates is a human being then Socrates is mortal.
(2) Socrates is a human being

Therefore (3) Socrates is mortal

Another way to write the above argument:

$$P \to Q$$
$$P$$
$$\therefore\ Q$$

2. Testing an argument for its validity

Three ways to test an argument for validity:

A. Critical rows

1. Identify the assumptions and the conclusion and assign variables to them.
2. Construct a truth table showing all possible truth values of the assumptions and the conclusion.
3. Find the **critical rows - rows in which all assumptions are true**
4. For each critical row determine whether the conclusion is also true.
   a. If the conclusion is **true in all critical rows**, then **the argument is valid**
   b. If there is at least one row **where the assumptions are true, but the conclusion is false**, then the argument is **invalid**

B. Using tautologies

The argument is true if the conclusion is true whenever the assumptions are true.
This means: If all assumptions are true, then the conclusion is true.
"All assumptions" means the conjunction of all the assumptions.

Thus, let A1, A2, … An be the assumptions, and B **-** the conclusion.

For the argument to be valid, the statement
If (A1 Λ A2 Λ… Λ An) then B must be **a tautology -** true for all assignments of values to its variables, i.e. its column in the truth table must contain only **T**

i.e.

$$(A1 \wedge A2 \wedge \ldots \wedge An) \rightarrow B \equiv \mathbf{T}$$

C. Using contradictions

If the argument is valid, then we have $(A1 \wedge A2 \wedge \ldots \wedge An) \rightarrow B \equiv \mathbf{T}$
This means that the negation of $(A1 \wedge A2 \wedge \ldots \wedge An) \rightarrow B$ should be a contradiction - containing only **F** in its truth table

In order to find the negation we have first to represent the conditional statement as a disjunction and then to apply the laws of De Morgan

$(A1 \wedge A2 \wedge \ldots \wedge An) \rightarrow B \equiv \sim(A1 \wedge A2 \wedge \ldots \wedge An) \vee B \equiv$

$\sim A1 \vee \sim A2 \vee \ldots. \vee \sim An \vee B$.

The negation is:

$\sim((A1 \wedge A2 \wedge \ldots \wedge An) \rightarrow B) \equiv \sim(\sim A1 \vee \sim A2 \vee \ldots. \vee \sim An \vee B)$

$\equiv A1 \wedge A2 \wedge \ldots. \wedge An \wedge \sim B$

The argument is valid if $A1 \wedge A2 \wedge \ldots. \wedge An \wedge \sim B \equiv F$

There are two ways to show that a logical form is a tautology or a contradiction:

a. by constructing the truth table
b. by logical transformations applying the logical equivalences (logical identities)

Examples:

1. Consider the argument:

$$\begin{array}{ll} & P \rightarrow Q \\ & P \\ \therefore & Q \end{array}$$

Testing its validity:

a. by examining the truth table:

P	Q	P→Q
T	**T**	**T**
T	F	F
F	T	T
F	F	T

b. By showing that the statement 'If all premises then the conclusion" is a tautology: The premises are P and P→ Q. The statement to be considered is:

(P Λ (P→ Q)) → Q

We shall show that it is a tautology by using the following identity laws:

(1) P→ Q ≡ ~P V Q

(2) (P V Q) V R ≡ P V (Q V R) commutative laws

(P Λ Q) Λ R ≡ P Λ (Q Λ R)

(3) (P Λ Q) V R ≡ (P V R) Λ (Q V R) distributive law

(4) P Λ ~P ≡ F

(5) P V ~P ≡ T

(6) P V F ≡ P

(7) P V T ≡ T

(8) P Λ T ≡ P

(9) P Λ F ≡ F

(10) ~(P Λ Q) ≡ ~P V ~Q De Morgan's Laws

		(P Λ (P→ Q)) → Q
by (1)	≡	~(P Λ (P→ Q)) V Q
by (10)	≡	(~P V ~(P→ Q)) V Q
by (1)	≡	(~P V ~(~P V Q)) V Q
by (10)	≡	(~P V (P Λ ~Q)) V Q
by (3)	≡	((~P V P) Λ (~P V ~Q)) V Q
by (5)	≡	(**T** Λ (~P V ~Q)) V Q
by (8)	≡	(~P V ~Q) V Q
by (2)	≡	~P V (~Q V Q)
by (5)	≡	~P V **T**
by(7)	≡	**T**

2. Consider the argument

$$\begin{array}{ll} & P \rightarrow Q \\ & Q \\ \therefore & P \end{array}$$

We shall show that this argument is invalid by examining the truth tables of the assumptions and the conclusion. The critical rows are in boldface.

P	Q	P→ Q	
T	**T**	**T**	
T	F	F	
F	**T**	**T**	**here the assumptions are true, however the conclusion is false**
F	F	T	

Exercise:

Show the validity of the argument:

1. P V Q (premise)
2. ~Q (premise)

Therefore P (conclusion)

a. by using critical rows
b. by contradiction using logical identities

Solution:

a. by critical rows

conclusion		Premises		
P	**Q**	**P V Q**	**~Q**	
T	T	T	F	
T	F	**T**	**T**	**Critical row**
F	T	T	F	
F	F	F	T	

b. By contradiction using identities

$((P \vee Q) \wedge \sim Q) \wedge \sim P \equiv$

$((P \wedge \sim Q) \vee (Q \wedge \sim Q)) \wedge \sim P \equiv$

$((P \wedge \sim Q) \vee \mathbf{F}) \wedge \sim P \equiv$

$(P \wedge \sim Q) \wedge \sim P \equiv$

$P \wedge \sim P \wedge \sim Q \equiv \mathbf{F} \wedge \sim Q \equiv \mathbf{F}$

POSSIBLE QUESTIONS

2 MARKS

1. Construct the truth table for ˥(P^Q).

2. Define tautology .

3. Prove that without using truth table (˥Q ^ (P →Q))→ ˥P is a tautology.

4. Prove that P→(Q˅R) ↔ (P→Q) ˅ (P→R).

5. Construct the truth table for ˥(P) ˅ ˥(Q).

6 MARKS

1.construct the truth table ˥(P ∨(Q∧ R)

2.show that (x)(H(x) →M(x)) ∧H(S) →M(S)

3. Define disjunctive normal form and conjunctive normal form. Also obtain disjunctive normal form of (P∧Q) ∨(˥P∧R) ∨(Q∧R)

4. Prove that (P∨Q) ∧ ˥(˥P ∧ (˥Q ∨ ˥R) ∨ (˥P ∧ ˥Q) ∨ (˥P ∧ ˥R)) is a tautology.

5.Verify that a proposition P ∨ ˥(P∧Q) is a tautology.

6. Obtain the PDNF of (P∧Q) ∨(˥P∧R) ∨(Q∧R).

7. Lions are dangerous animals. There are lions. There are dangerous animals.

8. Construct the truth table for (P↔R) ∧ (˥Q→S)

9.Obtain PDNF of (˥((P ∨Q) ∧R)) ∧ (P ∨R))

10.Demonstrate that R is a valid inference from the premises P→ $Q, Q \rightarrow R, and\ P$.