Course Code : MCS-212
Course Title : Discrete Mathematics
Assignment Number : MCA(1)/212/Assign/2024
Maximum Marks : 100
Weightage : 30% www.ignousite.com
Last Dates for Submission : 30th April, 2024 (for January session)
: 31st October, 2024 (for July session)

Q1: Use Mathematical Induction to prove that : $1^2 + 2^2 + \ldots\ldots + n^2 = n(n+1)(2n+1)/6$

Ans.

Suppose the given statement is $P(n)$

$$P(n) = 1^2 + 2^2 + 3^2 + 4^2 + \ldots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

$n = 1$

$$P(1) = 1 = \frac{1(1+1)(2\times1+1)}{6} = \frac{2\times3}{6} = \frac{6}{6} = 1$$

which are true

$n = k$ For

$$1^2 + 2^2 + 3^2 + \ldots + k^2 = \frac{k(k+1)(2k+1)}{6} \quad \text{---} (1)$$

Now we will prove $P(k+1)$ is also true

$$1^2 + 2^2 + 3^2 + \ldots + k^2 + (k+1)^2 = \frac{(k+1)(k+2)(2k+3)}{6}$$

L.H.S $\Rightarrow$

$$\frac{k(k+1)(2k+1)}{6} + \frac{(k+1)^2}{1} \quad \text{From (i)}$$

$$= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6}$$

$$= \frac{(k+1)\left[k(2k+1) + 6(k+1)\right]}{6}$$

$$= \frac{(k+1)\left[2k^2 + k + 6k + 6\right]}{6} = \frac{(k+1)(2k^2+7k+6)}{6}$$

$$= \frac{(k+1)(2k^2+4k+3k+6)}{6} = \frac{(k+1)(k+2)(2k+3)}{6}$$

Q2: How many different permutations are possible of the letters, taken all at a time, of the word: ASSESSES?

Ans. The word "ASSESSES" has 8 letters, and there are repeated letters. To find the number of different permutations, we can use the formula for permutations of a multiset.

The formula for permutations of a multiset with n elements, where n_1 elements are of one kind, n_2 elements are of another kind, and so on, is given by:

$$\frac{n!}{n_1! \cdot n_2! \cdot \ldots \cdot n_k!}$$

In the case of the word "ASSESSES," there are three distinct types of letters:

'A' appears once, 'S' appears four times, and 'E' appears three times.

So, the number of different permutations is:

$$\frac{8!}{1!\cdot 4!\cdot 3!}$$

Now, calculate the value:

$$\frac{40320}{1\cdot 24\cdot 6} = \frac{40320}{144} = 280$$

Therefore, there are 280 different permutations of the letters in the word "ASSESSES" when taken all at a time.

Q3: A die is rolled once. What are the probabilities of the following events:
a. Getting an odd number
b. Getting at least a value 2
c. Getting at most a value 2
d. Getting at least 7

Ans. When a die is rolled, it has six faces numbered from 1 to 6. Each face is equally likely to appear. Let's calculate the probabilities for the given events:

a. Getting an odd number:

The odd numbers on a die are 1, 3, and 5.

Probability = Number of favorable outcomes / Total number of outcomes

Probability = 3 (odd numbers) / 6 (total numbers on the die)

Probability = 1/2

b. Getting at least a value 2:

The values on the die that are 2 or greater are 2, 3, 4, 5, and 6.

Probability = Number of favorable outcomes / Total number of outcomes

Probability = 5 (favorable numbers) / 6 (total numbers on the die)

Probability = 5/6

c. Getting at most a value 2:

The values on the die that are at most 2 are 1 and 2.

Probability = Number of favorable outcomes / Total number of outcomes

Probability = 2 (favorable numbers) / 6 (total numbers on the die)

Probability = 1/3

d. Getting at least 7:

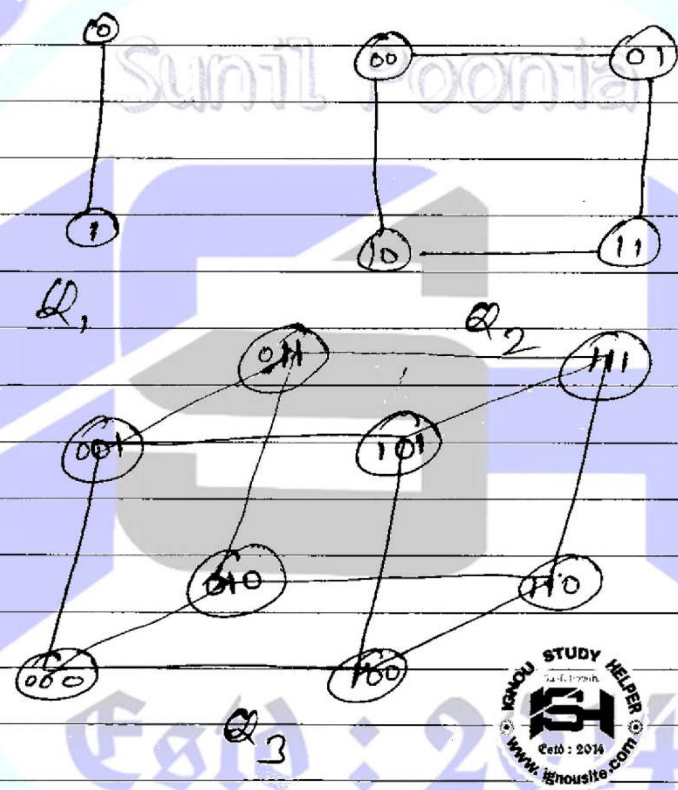
It is not possible to get a value of 7 or greater on a standard six-sided die.

Probability = 0 (no favorable outcomes) / 6 (total numbers on the die)

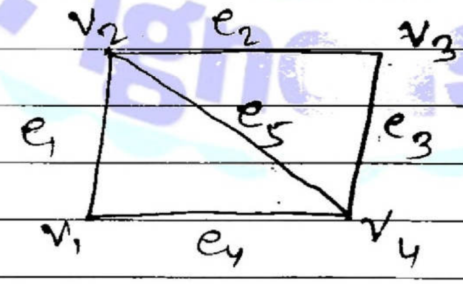
Probability = 0

Q4: Draw a hypercube graph Q3 (also called the cubical hypercube). Check whether the hypercube Q3 is Hamiltonian.

Ans → Hypercube → The n-cube Q_n is the graph whose vertices are binary n-tuples two vertices of Q_n are adjicent if and only if they differ in exactly one place.

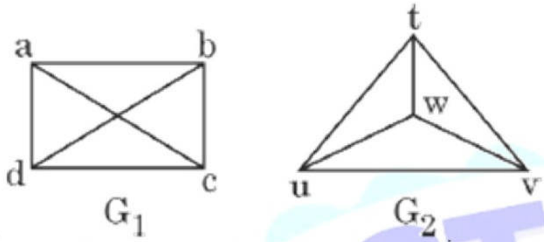


A graph G is called a Hamiltonian graph if it contains a Hamiltonian circuit.



Hamiltonian Graph.

Q5: What is isomorphism? Find, if the following graphs G1 and G2 are isomorphic or not. Explain how you arrived at your answer.

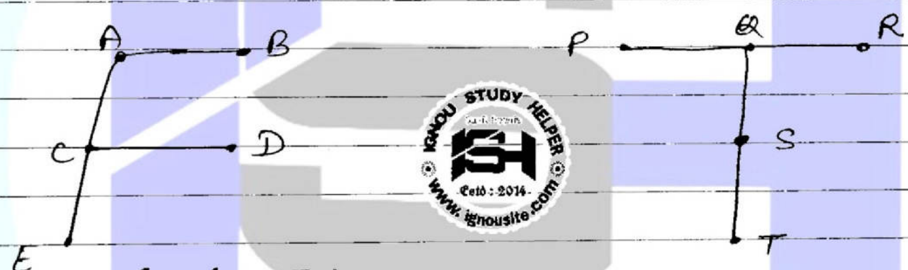


Ans → Isomorphism ⇒

Existance of different substance with some crystalline form is called isomorphism.

NaCl NaBr MgO

cubic crystal



$G_1(V_1; E_1)$ $G_2(V_2, E_2)$

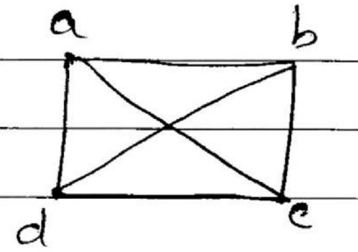
(1) No. of vertices = 5	(i) No. of vertices = 5
(2) No. of Edges = 4	(2) No. of edges = 4
(3) Degree squence (A, B, C, D, E) = (2, 1, 3, 1, 1)	(3) Degree squence (P, Q, R, S, T) = (1, 3, 1, 2, 1)
(4) One to One mapping	(4) One to One mapping

(3) C → A (2), D (1), E (1)

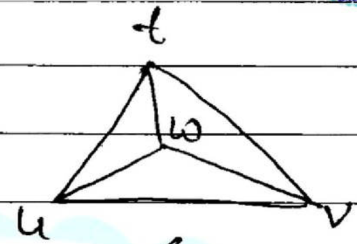
(3) Q → P (1), R (1), S (2)

(2) A → B (1), C (3)

(2) S → Q (3), T (1)



G_1,



G_2

No. of vertex = 4
No. of edges = 6
Degree of sequence = (a, b, c, d)
= (3, 3, 3, 3)

No. of vertex = 4
No. of edges = 6
Degree of sequence =
(u, v, w, t) = (3, 3, 3, 3)

Q6: What is a finite automata? Why is it needed? How is a finite automata represented? Also explain the term regular expression with the help of an example.

Ans. Finite automaton: A finite automaton, also known as a finite state machine, is a computational model used to recognize patterns in strings or sequences of symbols. It consists of a set of states, transitions between these states based on input symbols, and a set of accepting states.

Need for Finite Automata: Finite automata are essential in various areas of computer science and computational theory because they are used to recognize regular languages, which are fundamental in lexical analysis, parsing, pattern matching, and designing algorithms. Finite automata help in understanding and implementing efficient algorithms for text processing, compiler construction, and natural language processing tasks.

Representation of Finite Automata: Finite automata can be represented using diagrams, tables, or transition diagrams, where states are represented as nodes and transitions as directed edges between nodes labeled with input symbols.

Regular Expression: A regular expression is a concise and powerful notation used to describe patterns in strings or sequences of symbols. It's a sequence of characters that defines a search pattern, typically used for pattern matching in strings.

Example: consider the regular expression: ab+c

a matches the character 'a' exactly once.

b+ matches one or more occurrences of the character 'b'.

c matches the character 'c' exactly once.

So, the regular expression ab+c matches strings like "abc", "abbc", "abbbc", etc., where 'a' appears exactly once, followed by one or more 'b's, and ending with a 'c'.

Q7: Differentiate between
a. Deterministic and Non-deterministic finite automata
b. Deterministic and Non-deterministic Turing Machine
c. Moore and Mealy Machine

Ans.

a. Deterministic and Non-deterministic Finite Automata

Deterministic Finite Automata (DFA): In a DFA, at most one transition can be made from any state for each possible input symbol. It always knows what state to transition to based on the current state and the input symbol. Deterministic transitions make DFAs easier to understand and analyze but may require more states to represent certain languages.

Non-deterministic Finite Automata (NFA): In an NFA, multiple transitions can be made from a state for the same input symbol. The transition function is not deterministic, meaning there can be multiple possible next states for a given input. NFAs can be more concise than DFAs, as they can represent certain languages with fewer states and transitions.

b. Deterministic and Non-deterministic Turing Machine

Deterministic Turing Machine (DTM): In a DTM, the transition from one configuration to another is uniquely determined by the current state and the symbol being read from the tape. It always knows what state to transition to and what symbol to write on the tape based on the current state and the symbol read. Deterministic transitions make DTMs easier to simulate and analyze but may require more states and rules to represent certain computations.

Non-deterministic Turing Machine (NTM): In an NTM, there can be multiple possible transitions from a given configuration. The transition function is not deterministic, meaning there can be multiple possible next configurations for a given input and current configuration. NTMs are more expressive than DTMs and are used to model certain types of computations more naturally, although they are more complex to analyze.

c. Moore Machine and Mealy Machine:

Moore Machine: In a Moore machine, the outputs depend only on the current state of the machine. Outputs are associated with states rather than transitions. The output is produced when the machine enters a particular state.

Mealy Machine: In a Mealy machine, the outputs depend on both the current state and the input symbol. Outputs are associated with transitions. The output is produced when the machine makes a transition from one state to another based on the input symbol.

Q8: Describe the divide-and-conquer approach to solve recurrences ? Explain how this approach can be used to apply binary search in a sorted list.

Ans. The divide-and-conquer approach is a problem-solving paradigm that involves breaking down a problem into smaller sub-problems, solving them independently, and then combining their solutions to solve the original problem. This approach is often used in algorithm design and analysis. To solve recurrences using the divide-and-conquer approach, we typically follow these steps:

Divide: Break the problem into smaller sub-problems that are similar to the original problem but of reduced size.

Conquer: Solve each sub-problem recursively. If the sub-problems are small enough, solve them directly.

Combine: Combine the solutions of the sub-problems to obtain the solution to the original problem.

Applying Binary Search using Divide and Conquer: Binary search is a classic example of a divide-and-conquer algorithm. It is used to efficiently search for a specific element in a sorted list or array. Here's how the divide-and-conquer approach is applied to binary search:

Divide: Identify the middle element of the sorted list. Compare the target element with the middle element to determine whether it lies in the left or right half of the list.

Conquer: If the target element is equal to the middle element, the search is successful. If the target element is less than the middle element, recursively apply binary search to the left half of the list. If the target element is greater than the middle element, recursively apply binary search to the right half of the list.

Combine: If the target element is found during the recursive calls, return the index of the element. If the target element is not found, the recursion will eventually reach a base case (e.g., an empty list), indicating that the element is not present.

Pseudocode for Binary Search using Divide and Conquer:

```
def binary_search(arr, target, low, high):
    if low <= high:
        mid = (low + high) // 2

        if arr[mid] == target:
            return mid  # Element found
        elif arr[mid] < target:
            return binary_search(arr, target, mid + 1, high)  # Search in the right half
        else:
            return binary_search(arr, target, low, mid - 1)  # Search in the left half
    else:
        return -1 # Element not found
```

This binary search algorithm efficiently reduces the search space by half in each recursive call, making it a logarithmic time (O(log n)) algorithm for searching in a sorted list or array.

Q9: What is proposition? Explain with the help of an example. Explain Disjunction and Conjunction with the help of truth table for each.

Ans. Proposition: In logic, a proposition is a statement that is either true or false but not both. Propositions are often denoted by letters or symbols and are the building blocks of logical reasoning.

Example:

1. P: "The sky is blue."

This is a proposition because it makes a definite claim that can be either true or false.

If the sky is currently blue, the proposition is true; otherwise, it is false.

2. Q: "2 + 2 = 5."

This is also a proposition. In this case, it is false because the statement is mathematically incorrect.

Disjunction (Logical OR): The disjunction is a logical connective that represents the "OR" operation. The proposition P∨Q is true if at least one of P or Q is true.

Truth Table for Disjunction (P∨Q):

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

Conjunction (Logical AND): The conjunction is a logical connective that represents the "AND" operation. The proposition P∧Q is true only if both P and Q are true.

Truth Table for Conjunction (P∧Q):

P	Q	P ∧ Q
T	T	T
T	F	F
F	T	F
F	F	F

Q10: Prove the following theorem by direct proof method: "The square of an even integer is an even integer."

Ans. Certainly! Let's prove the theorem "The square of an even integer is an even integer" using the direct proof method.
Theorem: For any even integer n, n^2 is also an even integer.

Proof:

Let n be an even integer. By definition, an even integer can be expressed as n=2k, where k is an integer.

Now, let's consider the square of n:

$n^2=(2k)^2$

Expanding the expression:

n2=4k2

Since k^2 is an integer (since k is an integer), we can rewrite n^2 as:

$n^2=2(2k^2)$

Let $m=2k^2$. Since $2k^2$ is an integer, m is also an integer.

Therefore, we can express n^2 as:

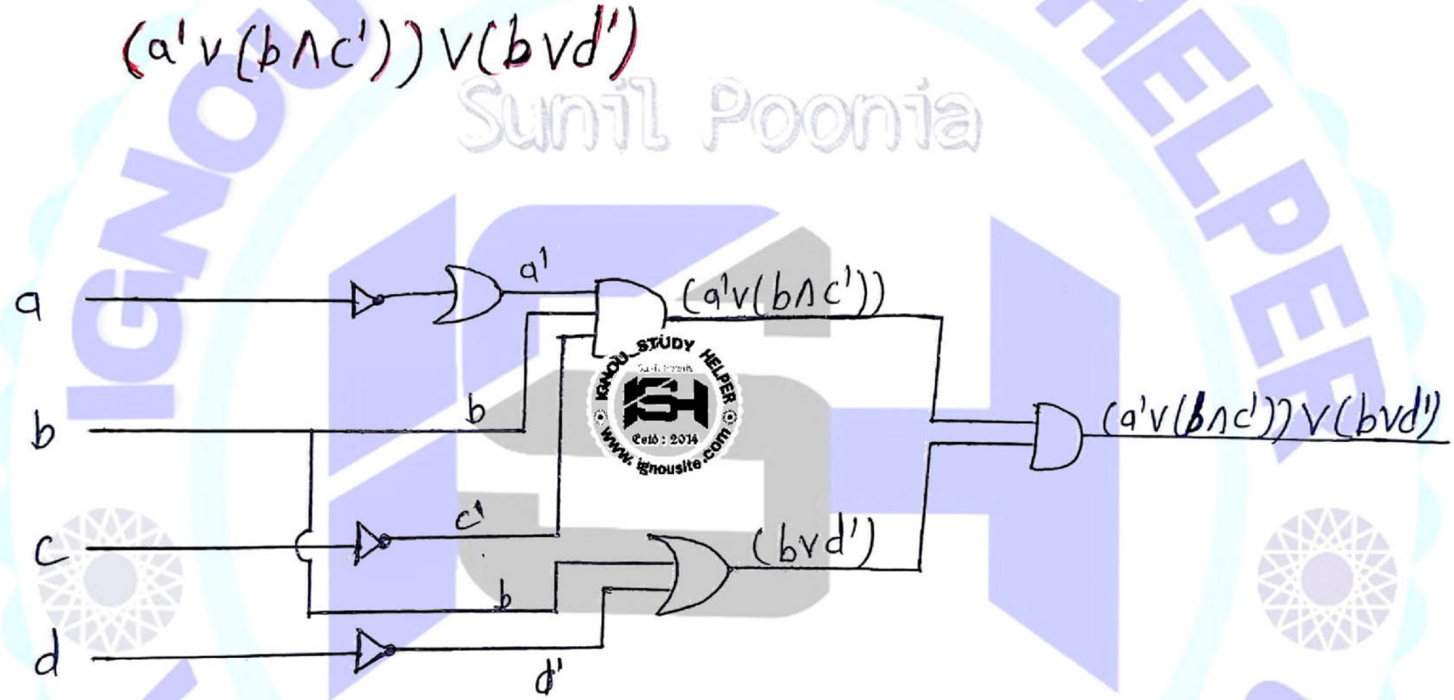
$n^2=2m$

This shows that n^2 is divisible by 2 and, by definition, an even integer.

Hence, we have proved that if n is an even integer, then n^2 is also an even integer.

Q11: Given the Boolean expression (a’ V (b ∧ c’)) V (b V d’), draw the corresponding circuit, where a, b, c and d are the inputs to the circuitry.

Ans.



Q12: Define the terms Domain, Co-domain and Range in the context of a function. Also find the domain, co-domain and range for a function A to B, where A={1,2,3,4} and B={1,4,9,16,25}.

Ans. The context of a function, the terms "domain," "co-domain," and "range" have specific meanings:

Domain: The domain of a function is the set of all possible input values (or independent variables) for which the function is defined. In simpler terms, it's the set of all x-values that you can plug into the function.

Co-domain: The co-domain of a function is the set of all possible output values (or dependent variables) that the function can produce. It represents the range of possible values that the function can output.

Range: The range of a function is the set of all actual output values (or dependent variables) that the function produces when you plug in values from the domain. It's the set of all y-values that the function can generate.

Now, let's find the domain, co-domain, and range for a specific function from set A to set B, where A={1,2,3,4} and B={1,4,9,16,25}.

Suppose the function f : A→B is defined such that $f(x)=x^2$ for all x in set A.

Domain: Since the function $f(x)=x^2$ is defined for all real numbers, and all elements of set A are real numbers, the domain of this function is A={1,2,3,4}.

Co-domain: The co-domain of the function is set B={1,4,9,16,25}, which is the set of all possible outputs of the function.

Range: To find the range, we need to determine what outputs the function can produce when we plug in values from the domain. For each element x in the domain, we compute $f(x)=x^2$. So, the range of the function is the set of all possible squares of the elements in the domain, which is {1,4,9,16} {1,4,9,16}. Therefore, the range of the function is {1,4,9,16} {1,4,9,16}.

Q13: A committee consisting of 2 male and 2 female workers is to be constituted from 8 male and 9 female workers. In how many distinct ways can this be done?

Ans. To find the number of distinct ways to constitute the committee, we'll use combinations, which is a counting technique that disregards the order of selection.

First, we need to choose 2 male workers from the 8 available male workers. This can be done in$\left(\frac{8}{2}\right)$ways.

Next, we need to choose 2 female workers from the 9 available female workers. This can be done in $\left(\frac{9}{2}\right)$ ways.

Finally, to find the total number of distinct ways to constitute the committee, we multiply the number of ways to choose male workers by the number of ways to choose female workers:

$\left(\frac{8}{2}\right) \text{X} \left(\frac{9}{2}\right)$

Using the formula for combinations:

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

we calculate:

$$\binom{8}{2} = \frac{8!}{2!(8-2)!} = \frac{8!}{2!6!} = \frac{8\times 7}{2\times 1} = 28$$

$$\binom{9}{2} = \frac{9!}{2!(9-2)!} = \frac{9!}{2!7!} = \frac{9\times 8}{2\times 1} = 36$$

So, the total number of distinct ways to constitute the committee is:

28×36=1008

Therefore, there are 1008 distinct ways to form the committee consisting of 2 male and 2 female workers.

Q14: In a tennis tournament, each entrant plays a match in the first round. Next, all winners from the first round play a second-round match. Winners continue to move on to the next round, until finally only one player is left as the tournament winner. Assuming that tournaments always involve $n = 2^k$ players, for some k, find the recurrence relation for the number rounds in a tournaments of n players.

Ans. Analyze the progression of rounds in the tournament:
In the first round, all n players participate, resulting in n/2 matches.

In the second round, n/2 winners from the first-round advance, resulting in (n/2)/2=n/4 matches.

In the third round, n/4 winners from the second-round advance, resulting in (n/4)/2=n/8 matches.

This pattern continues until there is only one player left, who is the tournament winner.

We can observe that the number of rounds needed for the tournament to conclude is determined by how many times n can be halved until it reaches 1.

Let R(n) represent the number of rounds needed for a tournament with n players.

When n=2, there's only one match, and the tournament concludes, so R(2)=1.

When n>2, each round halves the number of players, so the recurrence relation becomes:

R(n)=1+R(n/2)

This is the recurrence relation for the number of rounds in a tournament with n players. It expresses the number of rounds needed as one plus the number of rounds needed for half the number of players in the previous round.

Q15: Show, using the pigeonhole principle, that in any group of 30 people, 5 people can always be found who were born on the same day of the week.

Ans. To show that in any group of 30 people, 5 people can always be found who were born on the same day of the week, we can use the Pigeonhole Principle.

The Pigeonhole Principle states that if n items are placed into m containers and n>m, then at least one container must contain more than one item.

In this problem, we have 30 people and 7 days of the week. Each person's birthday can be associated with one of the seven days of the week. Since there are only 7 days of the week, at least two people must have been born on the same day of the week. We can see this as the "pigeons" and the "pigeonholes", where the "pigeonholes" represent the days of the week and the "pigeons" represent the people.

Now, if we consider the remaining 28 people, they could all have been born on different days of the week, but when we add the 2 people who share the same birthday, we have 3 people born on that specific day of the week.

For the remaining 28 people, again, each person's birthday can be associated with one of the seven days of the week. By the Pigeonhole Principle, at least two more people must have been born on the same day of the week. Now, we have 5 people born on the same day of the week.

Therefore, using the Pigeonhole Principle, we have shown that in any group of 30 people, 5 people can always be found who were born on the same day of the week.

Q16: Define the following in the context of graph, with the help of an example:
a. Complete graph
b. Degree of a vertex
c. Cycle
d. Path
e. Circuit

Ans.

Ans →

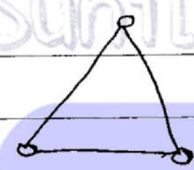
(a) Complete Graph ⇒

A Simple Graph is complete graph (k_n) in which every pair of vertex are adjesent.

k_1 = o

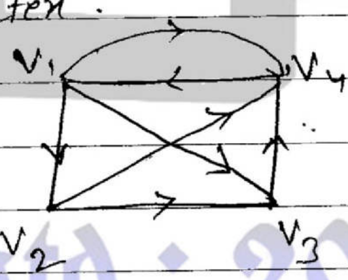
k_2 =

k_3 =

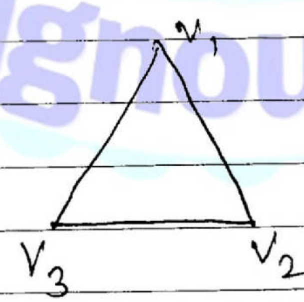


(b) Degree of a vertex ⇒

The Number of edges concenerted with a vertex is called Degree of a vertex.

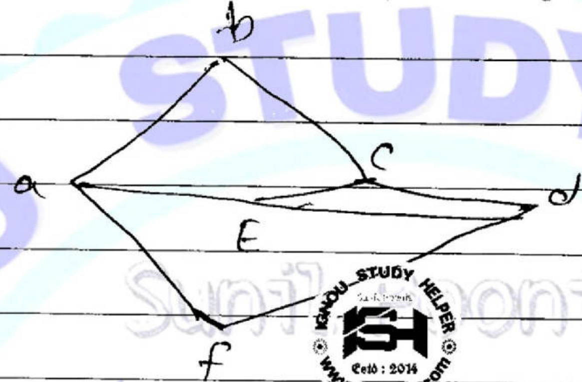


(c) Cycle ⇒ A graph containing atleast one cycle in it as known cycle.



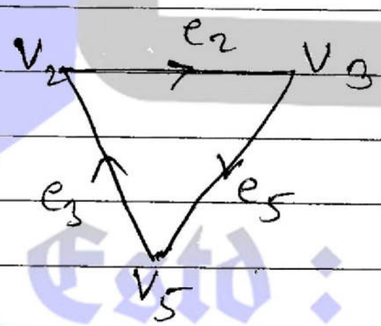
(d) Path =>

A open walk in which no vertice is appear more than ones is called path.



(e) Circuit =>

Circuit is a closed walk in which no edge appear more than once.



Q17: What is a bipartite graph? Explain with the help of an example. Give one or two applications of bipartite graphs.

Ans. Bipartite graph: A bipartite graph is a type of graph in which the vertices can be divided into two disjoint sets such that no two vertices within the same set are adjacent (connected by an edge). In other words, a bipartite graph is a graph whose vertices can be partitioned into two sets such that all edges connect vertices from different sets.

Example to illustrate a bipartite graph:

Consider a graph representing the relationships between students and courses in a university. Let the vertices be the students and the courses, and an edge exists between a student and a course if that student is enrolled in that course. In this graph:

One set of vertices represents the students.

The other set represents the courses.

An edge exists between a student and a course if the student is enrolled in that course.

This graph is bipartite because we can partition the vertices into two sets (students and courses) such that all edges connect a student with a course, and no two students or no two courses are connected directly.

Applications of bipartite graphs:

1. **Matching Problems**: Bipartite graphs are often used to model matching problems where there are two sets of entities, and the goal is to find pairs of entities such that each entity is paired with exactly one entity from the other set, with no two entities from the same set being paired together. For example, matching students to schools or employees to tasks.
2. **Recommendation Systems:** Bipartite graphs can be used in recommendation systems where there are two sets of entities, such as users and items. The edges in the graph represent interactions or preferences between users and items. By analyzing the bipartite graph, recommendation algorithms can suggest items to users based on the preferences of similar users or the items they have interacted with previously.

Q18: How Hamiltonian graphs differ from the Eulerian graphs? Give Dirac's and Ore's criterion for the Hamiltonian graphs.

Ans. Hamiltonian graphs and Eulerian graphs are two types of graphs characterized by different types of paths.

Hamiltonian Graphs: A Hamiltonian graph is a graph that contains a Hamiltonian cycle, which is a cycle that visits every vertex exactly once and returns to the starting vertex. In simpler terms, a Hamiltonian graph has a cycle that passes through every vertex exactly once. Finding a Hamiltonian cycle is generally a harder problem than finding an Eulerian cycle.

Eulerian Graphs: An Eulerian graph is a graph that contains an Eulerian cycle, which is a cycle that traverses every edge exactly once and returns to the starting vertex. In other words, an Eulerian graph has a cycle that contains all edges and returns to the starting vertex. Finding an Eulerian cycle can be relatively easier compared to finding a Hamiltonian cycle.

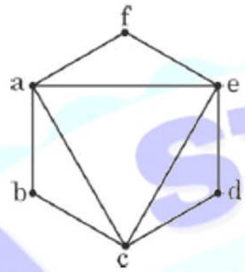
Dirac's Criterion for Hamiltonian Graphs: Dirac's theorem provides a sufficient condition for a graph to be Hamiltonian. It states that if a graph G has n vertices (where n≥3) and the degree of each vertex is at least 2n, then G is Hamiltonian.

In simpler terms, if every vertex in a graph has a degree that is at least half the total number of vertices, then the graph is likely to be Hamiltonian.

Ore's Criterion for Hamiltonian Graphs: Ore's theorem is another sufficient condition for a graph to be Hamiltonian. It states that if a graph G has n vertices (where n≥3) and for every pair of non-adjacent vertices u and v, the sum of their degrees is at least n/2, then G is Hamiltonian.

Mathematically, if for every pair of non-adjacent vertices u and v in G, deg(u)+deg(v)≥n, then the graph is likely to be Hamiltonian.

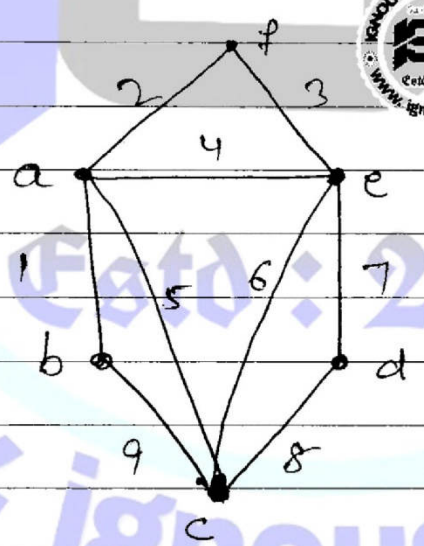
Q19: Differentiate between Eulerian graph and Eulerian circuit. Find the Eulerian circuit in the graph given below (if it exists).



Q.17

Ans → Eulerian ~~graph~~ Graph ⇒ An Euler graph is connected graph whose all vertices are of even degree.

Eulerian circuit ⇒ Euler circuit has no vertex and Number of edges.



every edge once no represents return the starting point.

Q20: Write Short notes on following
a. Travelling Salesman Problem
b. Vertex Coloring
c. Edge Coloring
d. Planar graphs
e. Pascal's Formula

Ans.

a. Travelling Salesman Problem (TSP): The Traveling Salesman Problem is a classic problem in the field of optimization and graph theory. It involves finding the shortest possible route that visits each city (or vertex) exactly once and returns to the starting city. Mathematically, it can be defined as finding the minimum Hamiltonian cycle in a complete weighted graph. TSP is NP-hard, meaning that there's no known polynomial-time algorithm that guarantees to solve it optimally for all instances. Various heuristics and approximation algorithms are employed to tackle TSP, such as nearest neighbor, genetic algorithms, and simulated annealing.

b. Vertex Coloring: Vertex coloring is a fundamental concept in graph theory. It involves assigning colors to the vertices of a graph in such a way that no two adjacent vertices share the same color. The minimum number of colors required to color the vertices of a graph without violating this rule is called the chromatic number of the graph. Vertex coloring has various applications, including scheduling problems, register allocation in compilers, and solving graph optimization problems.

c. Edge Coloring: Edge coloring is another important concept in graph theory. It involves assigning colors to the edges of a graph in such a way that no two adjacent edges share the same color. The minimum number of colors required to color the edges of a graph without violating this rule is called the chromatic index of the graph. Edge coloring has applications in scheduling problems, channel assignment in wireless communication, and designing network topologies.

d. Planar Graphs: A planar graph is a graph that can be embedded in the plane without any edges crossing each other. In other words, it can be drawn on a plane without any edges intersecting. Planar graphs are characterized by their ability to be represented visually without any edge crossings. They have several interesting properties, such as Euler's formula (which relates the number of vertices, edges, and faces in a planar graph) and Krakowski's theorem (which provides a characterization of non-planar graphs). Planar graphs have applications in circuit design, network layout, and geographical mapping.

e. Pascal's Formula: Pascal's formula, also known as Pascal's identity or the binomial theorem, is a fundamental result in combinatorics. It gives a formula for calculating the binomial coefficients, which represent the number of ways to choose k elements from a set of n elements without regard to the order. Pascal's formula states that the binomial coefficient (n/k) can be calculated as (n−1)/(k-1)+(n−1)/(k). This formula is useful in various combinatorial problems, such as counting arrangements, calculating probabilities, and solving recurrence relations. Pascal's formula is named after the French mathematician Blaise Pascal, who made significant contributions to combinatorics and probability theory in the 17th century.